Парадигма развития науки

Методологическое обеспечение

А.Е. Кононюк

КОНСАЛТОЛОГИЯ

ОБЩАЯ ТЕОРИЯ КОНСАЛТИНГА

Книга 2

Киев
„Корнійчук”
2009

УДК 51 (075.8)
ББК В161.я7
К65.

Рецензент: *Н.К. Печурин* - д-р техн. наук, проф. (Нацинальный авиационный университет).

Кононюк А.Е.
К65 Консалтология. Общая теория консалтинга
К.:"Корнійчук", 2009. К.2- 412 с.
ISBN 978-966-7699-50-8

Настоящая работа посвящена вопросам развития общей теории консалтинга с использованием научных методов формирования рекомендаций для решения задач консультируемых проблем. Сформулированы основные положения построения автоматизированных консультационных процессов. Рассмотрены принципы построения систем автоматизарованного консультирования (САК). С позиций пользователя (лица, формирующего рекомендации) изложены основные положения, связанные с разработкой, исследованием и реализацией сформированных рекомендаций по решению задач консультируемых проблем различных проблемных областей.

Книга предназначена для научных работников, магистров, аспирантов, докторантов соответствующих специальностей.

ББК В161.я7

ISBN 978-966-7699-50-8

Оглавление

3. Консультационный процесс, его маматические модели и его моделирование. Консультационные алгоритмы

3.1. Основные понятия и определения

При формировании рекомендаций выполняются определенные процессы. Существует ряд определений понятия «процесс». Мы под *процессом* будем понимать выполняемую совокупность действий, обеспечивающую направленное решение определенной задачи и могущая на основе определенных правил сформировать требуемые рекомендации по решению задач консультируемой проблемы.

При этом можно выделить два основных типа процессов: *детерминированный и стохастический*. Если при применении для одного и того же исходного материала одного и того же детерминированного процесса будет получен один и тот же продукт, то для одного и того же исходного материала, применяя несколько раз один и тот же стохастический процесс, вообще говоря, можно получить различные продукты. Однако при многократном использовании для одного и того же исходного материала одного и того же стохастического процесса можно получить с определенной достоверностью в среднем один и тот же продукт. Этим свойством стохастического процесса широко пользуются, например, при проведении статистического моделирования

Частным случаем процесса может служить выполняемая на ЭВМ программа с обрабатываемыми данными. При этом исходным материалом в данном случае являются исходные данные, а продуктом — результирующие. Другим частным примером процесса может служить процесс изготовления какой-либо детали. В этом случае исходным материалом служит заготовка, а продуктом — сама деталь.

Следует заметить, что понятие процесса является понятием динамическим, т. е. процесс существует только в том случае, если происходит его выполнение. Поэтому говорят, что *процесс зарождается, когда он начинает выполняться, и процесс умирает, когда его выполнение завершается.*

Процессы могут выполняться как в консультируемых проблемах, так и консультантами.

Рассмотрим вначале процессы, выполняемые в консультируемых проблемах.

Для обеспечения взаимосвязи различных проблемных блоков из которых состоит сложная проблема, между проблемными блоками создается система функциональных связей.

Последовательность выполнения функциональных операций на каждом проблемном блоке будем называть *функциональным процессом.*

Выполняемый консультантом процесс, обеспечивающий формирование рекомендаций по реализации функционального процесса консультируемой проблемы, будем называть *консультационным процессом или процессом консультирования*.

Функциональный процесс и соответствующий ему консультационный процесс можно выделить в любой системе «консультант – консультируемая проблема».

В консультируемой проблеме одновременно может выполняться несколько процессов или частей одного и того же функционального процесса. В этом случае будем говорить, что в консультируемой проблеме выполняется несколько параллельных функциональных процессов или один и тот же функциональный процесс имеет несколько параллельных участков (функциональных подпроцессов , или частных функциональных процессов).

Функциональный процесс, в котором допускается одновременное (параллельное) выполнение частных функциональных процессов, будем называть *параллельным функциональным процессом.* Функциональный процесс, в котором отсутствуют частные функциональные процессы, выполнение которых может быть осуществлено параллельно, будем называть *последовательным функциональным процессом.*

Легко понять, что если в консультикуемой проблеме параллельно выполняется r последовательных функциональных процессов, то без ограничения общности для простоты будем считать, что в консультикуемой проблеме выполняется один параллельный функциональный процесс с r параллельно выполняемыми частными функциональными процессами.

Аналогично *консультационный процесс* будем называть *параллельным,* если в нем допускается параллельное выполнение частных консультационных процессов. При этом так же, как и для функционального процесса, примем, что консультант выполняет один параллельный консультационный процесс, если консультант одновременно выполняет несколько консультационных процессов, которые будем называть *частными консультационными процессами.* Если в консультационном процессе нет параллельно выполняемых частных консультационных процессов (консультационных подпроцессов), то такой *консультационный процесс* будем называть *последовательным.*

С каждой функциональной операцией, из которых состоит функциональный процесс, в консультационном процессе может быть сопоставлена *процедура,* начало выполнения которой определяет начало выполнения соответствующей функциональной операции, а ее окончание — завершение этой функциональной операции.

Условно процедуру можно представить в виде трех частей: *начальная, основная и заключительная*. Начальная (пусковая) часть процедуры обеспечивает активизацию выполнения соответствующей функциональной операции консультируемой проблемой. Основная часть (тело процедуры) консультирует управление ходом выполнения функциональной операции, а заключительная часть обеспечивает останов выполнения функциональной операции. По окончании и в процессе выполнения функциональной операции процедура при необходимости может обеспечить снятие показаний датчиков консультируемой проблемы и в зависимости от их значений может обеспечить скорректированное продолжение выполнения этой функциональной операции. В частности, в процедуре может быть предусмотрен контроль за правильностью выполнения в консультируемой проблеме функциональной операции после завершения каждой из трех указанных выше частей процедуры.

В качестве исполнительных средств консультируемой проблемы могут использоваться разнообразные устройства и приборы, имеющие различные свойства.

По способу восприятия консультационного сигнала от консультанта исполнительные средства можно разделить на два класса: *с фиксацией и без фиксации воздействия.*

При применении в консультируемой проблеме исполнительных средств с фиксацией воздействия консультантом может реализоввываться такой консультационный процесс, процедуры которого имеют только две части: начальную, обеспечивающую включение некоторого исполнительного средства, и заключительную, обеспечивающую выключение этого исполнительного средства. Такое исполнительное средство будем называть *пусковым.*

В данном случае предполагается, что после активизаци функциональной операции блок консультируемой проблемы при включенном пусковом исполнительном средстве сам обеспечивает контроль за ходом ее выполнения и консультанту необходимо после ее завершения только выключить пусковое исполнительное средство, приостановив выполнение функциональной операции, или просто установить факт завершения выполнения функциональной операции, что обеспечивается заключительной частью процедуры. Такие блоки консультируемой проблемы будем называть *активными.*

Те блоки консультируемой проблемы, которые требуют со стороны консультанта постоянного консультирования и контроля за ходом выполнения функциональной операции в нем, будем называть *пассивными.*

Пусть в качестве блока консультируемой проблемы рассматривается электродвигатель с компрессором, нагнетающим воздух в резервуар. Если электродвигатель включается электромеханическим реле, т. е. пусковым исполнительным средством, которое после срабатывания блокируется, то после подачи консультантом сигнала включения пускового реле последнее запускает электродвигатель и, оставаясь заблокированным, обеспечивает непрерывную работу электродвигателя. Если при этом необходимо контролировать какие-либо параметры, например давление в резервуаре, создаваемое компрессором, приводимым в действие электродвигателем, то после создания необходимого давления в резервуаре датчик, сработав, отключит от электропитания электродвигатель, который после этого остановится. Если в резервуаре давление упадет ниже нормы, то датчик вновь создаст цепь электропитания электродвигателя. Таким образом, необходимое давление в резервуаре будет поддерживаться до тех пор, пока от консультанта не поступит рекомендация на выключение пускового реле — это есть пример активного блока консультируемой проблемы.

Консультационные процессы, процедуры которых не имеют тела, а включают только начальную и заключительную части, будем называть *стартстопными.*

Частным случаем стартстопных консультационных процессов является процесс, процедуры которого имеют только начальные части. В этом случае консультант подает только сигнал активизации выполнения функуиональной операции в блок консультируемой проблемы, после чего блок консультируемой проблемы функционирует автономно и выключается после завершения выполнения функуиональной операции. При этом предполагается, что время выполнения i-й функуиональной операции ограничено некоторым значением t_i. Поэтому консультанту нет нужды посылать в блок консультируемой проблемы рекомендацию по останове выполнения функуиональной операции. При этом консультант реализует такой процесс, в котором после начала выполнения i-й процедуры переход к следующей, за ней (i +1)-й (соседней) процедуры произойдет через время t_i.

Процесс, процедуры которого имеют только начальные части, а интервалы времени между двумя соседними процедурами π_i и π_{i+1} составляют t_i, будем называть *стартовым консультационным процессом.*

Стартовый процесс, у которого время t_i постоянно и одинаково для всех соседних пар имеющихся в процессе процедур, будем называть *синхронным стартовым консультационным процессом.*

При реализации синхронного стартового консультационного процесса консультанту, очевидно, необходимо иметь тактовый генератор.

Если время t_i не является постоянным и тем более неопределенным, то стартовый консультационный процесс реализовать невозможно, поэтому вместо стартового процесса можно использовать стартстопный или обычный консультационный процесс с тремя частями в каждой процедуре. В этом случае будем говорить, что консультант реализует *простой консультационный процесс.* Если в дальнейшем не будет специально оговорен вид процесса, то это будет означать, что подразумевается простой консультационный процесс.

Если консультант реализует стартстопный консультационный процесс и длительности выполнения каждой из трех частей его процедур постоянны и одинаковы, то может быть реализован такой консультационный процесс, при котором моменты начала выполнения начальной и заключительной частей процедуры будут задаваться тактовым генератором. В этом случае будем говорить, что консультант реализует *синхронный стартстопный консультационный процесс.*

Синхронный стартстопный консультационный процесс можно использовать и в том случае, когда времена выполнения тела различных процедур различны, но возможно установить максимальное время выполнения тела процедуры. Тогда интервалы времени между сигналами от тактового генератора будут определяться этим максимальным временем. Если же времена выполнения тел процедур не только различны, но и непостоянны и неопределенны, то вместо синхронного стартстопного консультационный процесса можно использовать *квазисинхронный стартстопный консультационный процесс,* при котором выполнение заключительной части процедуры повторяется через равные интервалы времени, определяемые тактовым генератором, до тех пор, пока не будет установлено, что тело процедуры выполнено, т. е. в консультируемой проблеме закончено выполнение соответствующей функциональной операции.

Вместо квазисинхронного стартстопного консультационный процесса в этом случае может быть использован и *асинхронный стартстопный консультационный процесс.* Однако в последнем случае из консультируемой проблемы необходимо посылать сигнал об окончании выполнения функциональной операции. При этом после завершения выполнения начальной части процедуры активизируется заключительная ее часть, которая ожидает появление на входе сигнала для консультанта от консультируемой проблемы об окончании соответствующей функциональной операции.

Таким образом, когда консультант осуществляет выполнении асинхронного стартстопного консультационного процесса из консультируемой проблемы должны посылаться консультанту сигналы об окончании функциональных операций. Если же реализуется синхронный стартстопный консультационный процесс, то эти сигналы имитируются таковым генератором.

Простой консультационный процесс также может быть *синхронным и асинхронным*.

Следует заметить, что в большинстве случаев консультанту целесообразно реализовывать *смешанный консультационный процесс,* в котором сочетаются все или часть указанных выше разновидностей консультационных процессов.

В процессе выполнения консультационный процесс использует программно-аппаратурные ресурсы САК. Так, для того чтобы в САК выполнялась некоторая программа, должен использоваться процессор, заниматься один или несколько блоков оперативного запоминающего устройства. Возможно использование также и других ресурсов САК.
В том случае, когда в САК используется многопрограммный режим работы, то один и тот же процессор, блок памяти или канал ввода-вывода может затребоваться различными программами. Поэтому возникает их конкуренция при занятии того или иного ресурса, в частности процессора.

При наличии нескольких процессоров конкуренция параллельно выполняемых в них программ (процессов) может возникнуть при занятии одного и того же блока ОЗУ. Конкуренция процессов может возникнуть и при управлении формированием рекомендаций. Таким образом, конкуренция консультационных процессов может возникнуть не только от использования общих ресурсов САК, но и из-за общих ресурсов консультируемой проблемы.

Для того чтобы устранить конкуренцию параллельно выполняемых процессов при занятии некоторого общего ресурса, необходимо предусмотреть определенные меры.

Процесс (консультационный, управляющий) будем называть *правильно построенным,* или *правильным процессом,* если в нем отсутствует конкуренция параллельных частных процессов из-за занятия общих ресурсов.

Легко понять, что если в консультационном и в управляющем процессах нет параллельно выполняемых участков (частных процессов, подпроцессов), то конкуренции частных процессов из-за общих ресурсов не может быть —общие ресурсы используются частными процессами последовательно. Таким образом, если в САК выполняется последовательный консультационный процесс, т. е. процесс без параллельных участков, или такой параллельный консультационный процесс, в котором параллельные участки (подпроцессы) не требуют общих ресурсов САК, то, очевидно, любой консультационный процесс для такого консультационного процесса свободен от конкуренции из-за общих ресурсов САК. Если же в САК допускается выполнение параллельного консультационного процесса, в котором при выполнении подпроцессов может возникнуть необходимость использования общего ресурса, то даже в последовательном консультационном процессе возможна конкуренция подпроцессов из-за общих ресурсов САК. В этом случае требуется перевод консультационного процесса в правильный для обеспечения устранения конкуренции подпроцессов в управляющем процессе из-за общих ресурсов САК.

Однако не всегда целесообразно при формировании консультационного процесса добиваться того, чтобы он был правильным. Может оказаться целесообразным перевод его в правильный осуществлять в процессе разработки управляющего процесса. При этом конкуренция подпроцессов консультационного процесса и подпроцессов управляющего процесса из-за общих ресурсов САК может быть устранена выполнением консультационных операций, которые требуют одних и тех же ресурсов САК.

Если используется параллельный консультационный процесс, то в нем может быть конкуренция подпроцессов за счет общих ресурсов САК.

Таким образом, в качестве исходных данных при построении правильного (корректного) консультационного процесса, а затем и консультационного алгоритма (алгоритма формирования рекомендаций) служит консультационный процесс. Процесс построения консультационного алгоритма на основе консультационного процесса из-за сложности может быть представлен в виде *двух основных этапов*.

На первом этапе формируется консультационный процесс, в котором устраняются все конкуренции из-за общих ресурсов САК. При этом на основе параллельного консультационного процесса может быть построен как параллельный, так и последовательный консультационные процессы. На втором этапе на основе правильного (корректного) консультационного процесса формируется консультационный алгоритм (алгоритм формирования рекомендаций), который рассматривается в виде условий работы консультационного процесса и является исходным данным для его синтеза.

3.2. Формальное писание и преобразование консультационных процессов

3.2.1. Исчисление высказываний как язык описания процессов

Сравнительно недавно консультирование проблем осуществлялось без использования средств вычислительной техники. Бурное развитие методов построения логических схем позволило создать методы построения систем автоматизированного консультирования (САК). Рассмотрим аппарат математической логики, открывший широкие возможности для построения схем автоматизированных консултационных процессов.

Логика является наукой о формах и законах мышления. Одна из отраслей логики, развивающаяся применительно к потребностям математики, называется математической логикой, а одним из ее разделов является алгебра логики или Булева алгебра.

Начало разработке логического исчисления, позволяющего оперировать логическими суждениями так же, как алгебраическими символами в элементарной математике, положил Буль.

Исчисление высказываний — это первое и наиболее широко применяемое в математической логике понятие.

Высказывание - это всякое предложение, которое может быть либо истинным, либо ложным и не может иметь никакого третьего значения. Отдельные простые высказывания могут быть связаны при помощи различных логических связей в сложные, которые также могут быть либо истинными, либо ложными. Используем двоичный код: истинное высказывание обозначим единицей (1), ложное — нулем (0). Благодаря этому на базе нескольких простейших электронных или иных элементов представляется возможным синтезировать любые сложные логические высказывания.

Рассмотрим основные связи между простыми высказываниями.

Предложение «Коля пойдет гулять, если будет тепло и сухо» может быть разбито на два простых высказывания или предложения, связанные союзом *И:* «Коля пойдет гулять, если будет тепло» и «Коля пойдет гулять, если будет сухо».

Здесь возможны четыре варианта: не будет тепло (0) и не будет сухо (0), т. е. оба составляющие высказывания оказываются ложными. Очевидно, что в этом случае Коля гулять не пойдет, и все предложение в целом оказывается ложным. Во-вторых, возможно, что будет тепло (1), но не будет сухо (0), т. е. первое высказывание оказывается истинным, а второе, ложным. И в этом случае Коля не пойдет гулять и сложное предложение окажется ложным. В третьем варианте будет холодно (не тепло), но будет сухо (1); здесь первое высказывание ложно, а второе истинно и сложное высказывание опять-таки ложно.

Сложное высказывание, объединяющее союзом *И* два и более простых высказываний, будет истинным только тогда, когда истинны все составляющие его высказывания, и будет ложным во всех остальных случаях. В нашем примере все предложение будет истинным, когда будет тепло и сухо. Это и является четвертым возможным вариантом, когда Коля действительно пойдет гулять.
Обозначим сложное высказывание — возможность прогулки Коли — через P, а составляющие его простые высказывания «Коля пойдет гулять, если будет тепло» — a и «Коля пойдет гулять, если будет сухо» — b. Тогда сложное высказывание можно записать в виде формулы

$$P = ab,$$

которая обычно называется логическим умножением, или логической функцией *И*.

Кроме союза *И,* для объединения простых высказываний употребляется также союз *ИЛИ.* Например, предложение «Вечером Петя пьет чай или ужинает» будет истинным в трех случаях: когда Петя пьет чай, но не ужинает; когда Петя ужинает, но не пьет чай, и, наконец, когда он и ужинает, и пьет чай. Следовательно, сложное предложение, объединяющее союзом *ИЛИ* два или более простых высказывания, будет ложным только тогда, когда ложны все составляющие его высказывания, и истинным во всех остальных случаях.

Эта логическая связь — логическое сложение— может быть записана в виде формулы

$$P = a + b.$$

Третья логическая функция носит название *НЕ* — отрицание. Например, мы имеем простое высказывание «Дождь идет». Образуем из него сложное при помощи отрицания — «Дождь *НЕ* идет». Если простое высказывание истинно, т. е. действительно идет дождь, то сложное высказывание будет ложным. Если же простое высказывание ложно, т. е. дождь не идет, то выражение «Дождь *НЕ* идет» будет истиной. Эта функция имоет вид

$$P = \overline{A}$$

и читается как «*P* равно не *A*».

Основные законы алгебры логики

Алгебра логики подчиняется законам, иногда совпадающим с законами обычной алгебры, а иногда — своим своеобразным законам. Рассмотрим основные законы алгебры логики.

Законы множеств:

$$0 \bullet a = 0;$$
$$0 + a = a;$$
$$0 \bullet abc \ldots w = 0,$$

т. е. произведение любого числа переменных обращается в нуль, если какая-либо одна переменная имеет значение 0, независимо от значений других переменных;

$$1 \bullet a = a;$$
$$1 + a = 1;$$
$$1 + a + b + c + d + \ldots + w = 1,$$

т. е. сумма любого числа переменных обращается в единицу, если одна из ее переменных имеет значение 1, независимо от значений других переменных.

Законы перемещения:

$$ab = ba;$$
$$a + b = b + a,$$

т. е. результаты выполнения операций умножения и сложения не зависят от того, в каком порядке следуют переменные.

Законы тавтологии (повторения):

$$aa = a,$$
$$aaa \ldots a = a^n = a,$$
$$a + a = a,$$
$$a + a + a + \ldots + a = na = a.$$

Здесь можно сказать, что истина или ложь всегда остается истиной (или ложью), сколько ее не повторяй.

Законы дополнительности:

а) логическое противоречие:

$$a\overline{a} = 0$$

т. е. произведение любой переменной и ее инверсии есть 0. Как пример можно привести строку из известной песни «Речка движется и не движется» — заведомая ложь, б) закон исключенного третьего:

$$a + \overline{a} = 1,$$

т. е. сумма любой переменной и ее инверсии есть 1. Так, например, утверждая, что «студент сдаст экзамен или не сдаст», мы всегда будем правы.

Законы инверсии (Де Моргана):

$$\overline{ab} = \overline{a} + \overline{b},$$

т. е. инверсия произведения равна сумме инверсий;

$$\overline{a+b} = \overline{a}\overline{b},$$

а инверсия суммы есть произведение инверсий.

Здесь записаны законы для двух переменных, но они справедливы для любого числа переменных.

Законы распределительные (дистрибутивные)}

а) произведения относительно суммы:

$$a(b + c) = ab + ac.$$

Справедливость этого закона можно подтвердить высказываниями. Например: «Я зайду за Вами *И* мы пойдем в театр *ИЛИ* в кино». Так можно формулировать левую часть приведенного выше выражения, а правая часть тогда может быть прочитана так: «Я зайду за Вами *И* мы пойдем в театр *ИЛИ* я зайду за Вами *И* мы пойдем в кино». Смысл один и тот же, но правая часть несколько длиннее:

б) суммы относительно произведения:

$$a + bc = (a + b)(a + c).$$

Справедливость этого закона можно доказать, опираясь на предыдущие. Раскрыв скобки, получим

$$a + bc = aa + ac + ba + be = a + ac + ba + bc.$$

Из первых двух членов вынесем за скобки переменную *а:*

$$a(1+c),$$

но

$$1 + c = 1, \text{ а } a \cdot 1 = a.$$

Рассматривая следующее выражение $ba + bc$, мы устанавливаем, что и оно равно a. Тогда вся правая часть превращается в $a + bc$, т. е. в такое же выражение, как и в левой части.

Законы склеивания:

$$ab + a\overline{b} = a;$$

$$(a + b)(a + \overline{b}) = a.$$

Эти законы легко подтверждаются на основании рассмотренных ранее законов, например:

$$ab + a\overline{b} = a(b + \overline{b}) — a \bullet 1 = a.$$

Законы поглощения:

$$a(a + b) = a;$$
$$a(a + b)(a + c) \dots (a + w) = a;$$
$$a + ab = a;$$
$$a + ab + ac + .., + aw = a;$$
$$a(\bar{a} + b) = ab;$$
$$a + \bar{a}b = a + b.$$

Эти законы можно легко доказать с помощью других законов алгебры логики, например, умножая в последнем выражении первый член a на $(1 + b)$, получаем

$$a(1 + b) + \bar{a}b = a + ab + \bar{a}b = a + b(a + \bar{a}) = a + b \bullet 1 = a + b.$$

Так как для логического сложения и умножения характерны все свойства сложения и умножения алгебры чисел, то над многочленами алгебры высказываний можно производить те же действия, что и над многочленами алгебры чисел. Но логическое сложение и умножение обладают и некоторыми необычными свойствами и это приводит к необычности действий над логическими многочленами. Разъясним это на примере.

Пример. Пусть необходимо умножить $(a + b)$ на $(a+c)$. Умножаем по обычным правилам умножения многочлена на многочлен:

$$(a + b)(a + c) = aa + ac + ab + bc.$$

Так как в алгебре высказываний $aa = a$, то

$$(a + b)(a + c) = a + ac + ab + bc.$$

Но работу над полученным произведением можно продолжить. Рассмотрим два первых слагаемых a и ac. Сгруппируем их и общий множитель a вынесем за скобки:

$$a + ac = a(1 + c),$$

и далее

$$a \cdot 1 = a.$$

Также поступим и с суммой $a + ab = a$. Тогда окончательно

$$(a + b)(a + c) = a + bc.$$

Результат оказался проще, чем мы ожидали, так как выражение $a + ab$, согласно закону поглощения, заменили множителем a.

Таким образом, если высказывание логически складывают с логическим произведением, в состав которого оно входит, то оно поглощает это произведение. Отметим еще одну важную особенность алгебры высказываний. Если в формуле $a + ab$ заменить знак + на знак × и знак × на знак +, то полученное новое высказывание $a(a + b)$ будет эквивалентно заданному. В этом легко убедиться, раскрыв скобки.

Это свойство распространяется на любую формулу алгебры высказываний.

Например,

$$a(b + c) = ab + ac.$$

К обеим частям применим упомянутую замену знаков и получим:

$$a + bc = (a + b)(a + c).$$

Еще пример. Дано $ab + a\bar{b}$. В левой части заменим знаки:

$$ab + a\bar{b} = (a + b)(a + \bar{b}) = a$$

или $$(a + b)(a + \bar{b}) = a.$$

Эту особенность преобразования формул в алгебре высказываний называют законом двойственности. Опираясь на закон двойственности, легко преобразовать эквивалентные высказывания.

Упрощение логических выражений

Следует иметь в виду, что каждое логическое высказывание можно воплотить с помощью логических элементов в конкретный действующий автоматический механизм. Для этого каждое логическое сложение, т. е. знак плюс в формуле высказывания, следует осуществить логическим элементом *ИЛИ,* каждое логическое умножение —- элементом *И,* а каждое отрицание или инверсию — элементом *НЕ* (рис. 3.1).

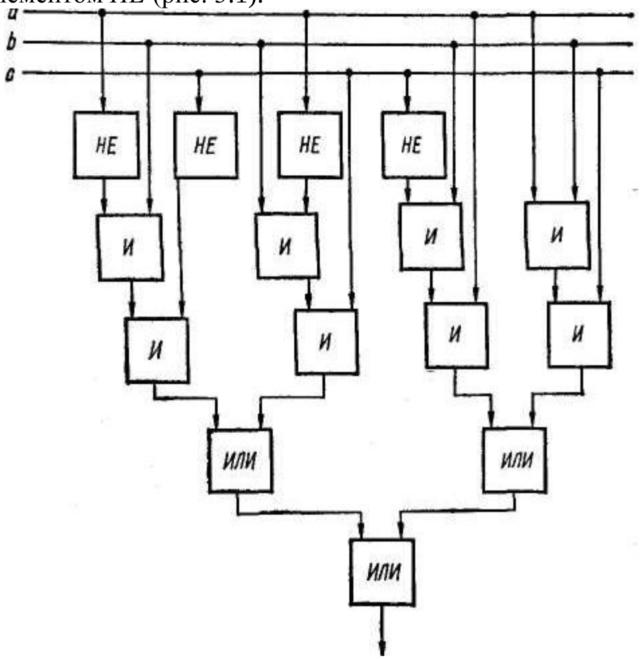

Рис. 3.1. Логическая схема.

Рассмотрим выражение

$$x = abc + \bar{a}bc + ab\bar{c} + \bar{a}b\bar{c}.$$

Заметим, что первое и третье, второе и четвертое слагаемые склеиваются по букве *a*.

Действительно,

$$abc + \bar{a}bc = bc$$

и

$$ab\bar{c} + \bar{a}b\bar{c} = b\bar{c}.$$

Имеем

$$x = bc + b\bar{c}.$$

В этой формуле можно произвести склеивание по букве *c* и тогда окончательно записанное ранее сложное выражение принимает очень простой вид: $x = b$.

Значит, математическая обработка выражения, построенного по законам математической логики, устранила необходимость в 15 логических элементах.

Мы произвели упрощение логического выражения, используя основные законы математической логики, путем последовательных рассуждений. Пример, который мы рассмотрели, относительно прост, да и то разные люди могли бы его решать по-разному и получать разные результаты. Тем более, такое явление может иметь место при упрощении более сложных логических выражений, в которых участвует большое число переменных, и которые выражаются более сложными зависимостями.

Задача упрощения логических выражений, или, как говорят, их «минимизация», является одной из наиболее сложных в алгебре логики. Есть много различных способов минимизации.

Рассмотрим один из распространенных способов минимизации логических выражений с помощью карт Карно.

Карты Карно (Karnaugh) наглядно изображают логические функции. Карта Карно (рис. 3.2) разделена на квадратики, и каждому из них отвечает определенная комбинация значений всех входных переменных. Кроме того, каждая сторона квадратика представляет собой границу между значениями переменных (верхний и нижний, равно как и боковые квадратики карты, являются соседними).

Обозначения входных переменных указываются сверху и сбоку карты и относятся ко всему столбику или строке квадратиков, причем значения этих входных переменных в них принимаются равными единице. В соседних с обозначенными в столбцах или строках входные переменные соответственно равны нулю. В квадратиках записывается значение самой функции при данных комбинациях значений входных переменных. Значения входных переменных не принято записывать в квадратиках, они подразумеваются, поэтому на карте остается только значение функции (рис. 3.2, *в, г, д, е)*. Из примеров, приведенных на рисунке для двух, трех и четырех переменных, видно, что прибавление каждой новой переменной удваивает карту.

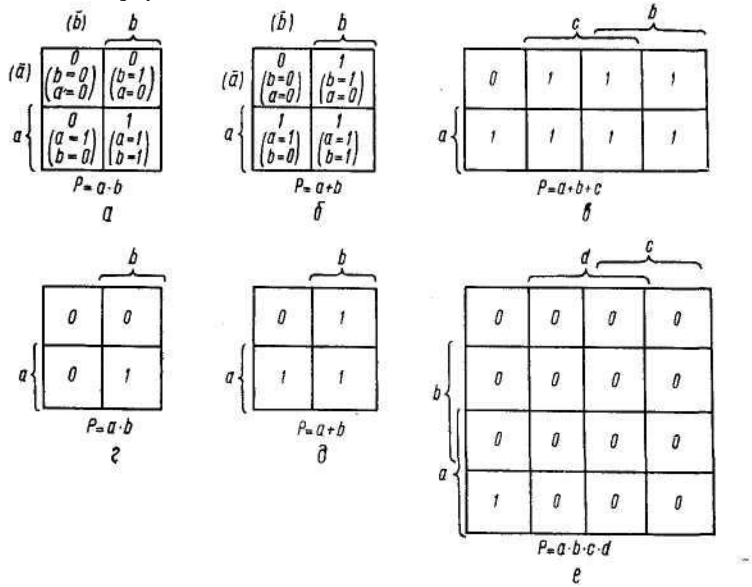

Рис. 3.2. Карты Карно.

Применяя карты Карно для изображения алгебраического выражения функций, можно записать функции либо в виде суммы произведений, либо в виде произведения сумм.

Выражение суммы произведений определяется суммой произведений значений всех входных переменных (прямых и инверсных) в каждом из квадратиков карты, содержащих единицу. Так, например, для карты, показанной на рис. 3.2*, а, г, P=ab,* а для карты, показанной на рис. 3.2*, б, д, —*$P = ab + ab + \bar{a}b$.

Выражения сомножителей в произведении сумм определяются суммами инверсных значений входных переменных в каждом из нулевых квадратиков. Так, например, для карты, представленной на рис. 3.2*, а, в,*

$$P=(a + b)(a + \bar{b})(\bar{a} + b);$$

для карты, показанной на рис. 3.3*, б, д,*

$$P = a + b.$$

С помощью карт Карно можно получить упрощенное выражение функций, для чего определяют суммы произведений и произведения сумм, объединяя квадратики, в которых значения функции соответственно равны 1 или 0, в контуры. Последние должны иметь форму прямоугольников и содержать четное число квадратиков или только один квадратик.

Из свойств карт Карно вытекает, что при переходе контура из одного квадратика к другому одна из переменных инвертируется. Поэтому выражение контура из двух квадратиков не зависит от этой переменной, а определяется только остальными переменными, т. е. выражения, соответствующие контурам, «не содержат тех переменных, чьи границы пересекаются данным контуром». Так, контур, ограничивающий четыре квадратика, пересекает две границы двух переменных и поэтому соответствующее ему выражение содержит n—2 переменных и т. д.

Для получения наиболее простых выражений, реализуемых минимальным количеством возможно более простых логических выражений, т. е. при минимизации, логическое выражение должно иметь как можно меньше членов, каждый из которых должен содержать как можно меньше переменных.

Правила минимизации выражения логической функции по карте Карно сводятся к следующему. Чем большее число квадратиков с одним значением функции объединяется в общем контуре на карте и чем меньше будет таких контуров, тем проще будет аналитическое выражение функции. При этом все квадратики с одним значением функции должны входить в какой-нибудь контур. Нужно также следить за тем, чтобы какой-либо контур не входил полностью в другие контуры.

Рассмотрим подробнее карту Карно для трех переменных (рис. 3.3).

Для наглядности в квадратиках указаны значения переменных. Пусть функция должна быть заложена в карту Карно и минимизирована.

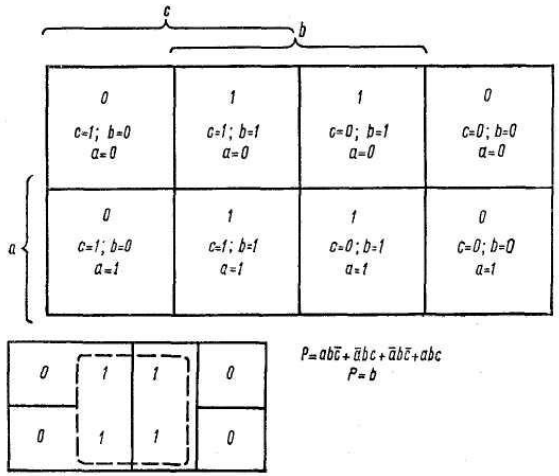

Рис. 3.3. Карта Карно для трех переменных.

Подставив значения переменных, соответствующие левому верхнему квадратику, в выражение функции, получим

$X= abc + ab\bar{c} + \bar{a}bc + \bar{a}b\bar{c} = 001 + 000 + 101 + 100 = 0 + 0 + 0 + 0 = 0$**.**

Это значит, что значение сложного выражения в этом квадратике равно 0, что и записываем.

Определяя таким образом значение выражения для всех квадратиков, получаем карту, показанную на рис. 3.3, *б.*

Охватываем контуром средние четыре квадратика со значениями 1. В этом контуре только лишь переменная сохраняет свое значение, равное единице во всех квадратиках. Следовательно, результат минимизации определяется выражением

$$X = b.$$

Такое же значение мы получили ранее более сложным приемом минимизации — аналитическим путем с использованием основных законов алгебры логики,

Разница в затратах труда на минимизацию становится тем значительнее, чем больше переменных в логическом выражении и чем оно сложнее.

Схемы систем атоматизированного консультирования основаны на так называемых двухпозиционных приборах, т. е. устройствах, способных занимать только одно из двух устойчивых положений. Сигнал, поступающий на вход в систему или снимаемый с выхода системы, может либо присутствовать (1), либо отсутствовать (0).

Поэтому в дальнейшем под переменными будем понимать сигналы на входе в схему, а под сложными высказываниями — сигналы на выходе, являющиеся логическими функциями этих переменных. Задача логической части схемы — выработать сигналы на выходе, являющиеся логическими функциями сигналов на входах.

Для решения задачи составления схемы необходимы электрические, электронные или пневматические устройства, осуществляющие элементарные логические связи *И, ИЛИ, НЕ.* Такие устройства называются элементами. Допустим, что наличию сигнала соответствует 1, т. е, истина, а отсутствию сигнала — 0, т. е. ложь. В электрических системах истине соответствует подача тока, а отсутствие тока — лжи. В пневматических системах наличие сигнала означает подачу сжатого воздуха под давлением, а отсутствие сигнала - соединение с атмосферой.

Логический элемент типа *И* должен иметь два или больше входа и один выход, с которого сигнал снимается.

В электрическом устройстве, в котором два реле *а* и *b* включены последовательно (рис. 3.4), сигнал на выходе появится при подаче напряжения на катушки обоих реле. В этом случае элемент выполнит логическую операцию умножения $P = ab$.

В пневматическом устройстве сигналы, т. е. сжатый воздух, поступают от двух пневматических кнопок *а* и *b*. Если нажать на одну из кнопок и подать воздух под давлением на один из входов, то две тарели клапана, сидящие на одной оси, передвинутся в одну сторону и свяжут выход клапана с атмосферой через вторую кнопку. Если нажать обе кнопки, то независимо от того, какое положение займут тарели, сжатый воздух пройдет на выход.

Элемент *ИЛИ* также имеет два входа и один выход. Если два реле *а* и *b* соединены параллельно, то сигнал на выходе появится, если подать напряжение на любое реле, и тогда элемент выполняет операцию логического сложения $P=a + b$.

В пневматическом варианте, если обе кнопки отпущены, выход элемента связан с атмосферой по крайней мере через одну из кнопок. Нажмем, например, кнопку *а.* Под действием давления сжатого воздуха шарик переместится вправо, прижмется к резиновому кольцу и не даст воздуху выходить в атмосферу через кнопку *b,* вследствие чего воздух поступит на выход клапана—элемента. Если отпустим кнопку *а* и нажмем кнопку *b,* то шарик переместится влево, не давая воздуху выходить в атмосферу через кнопку *а,* и на выходе тоже появится сигнал. Нажмем обе кнопки вместе, и, в каком бы ни был положении шарик, появится сигнал на выходе.

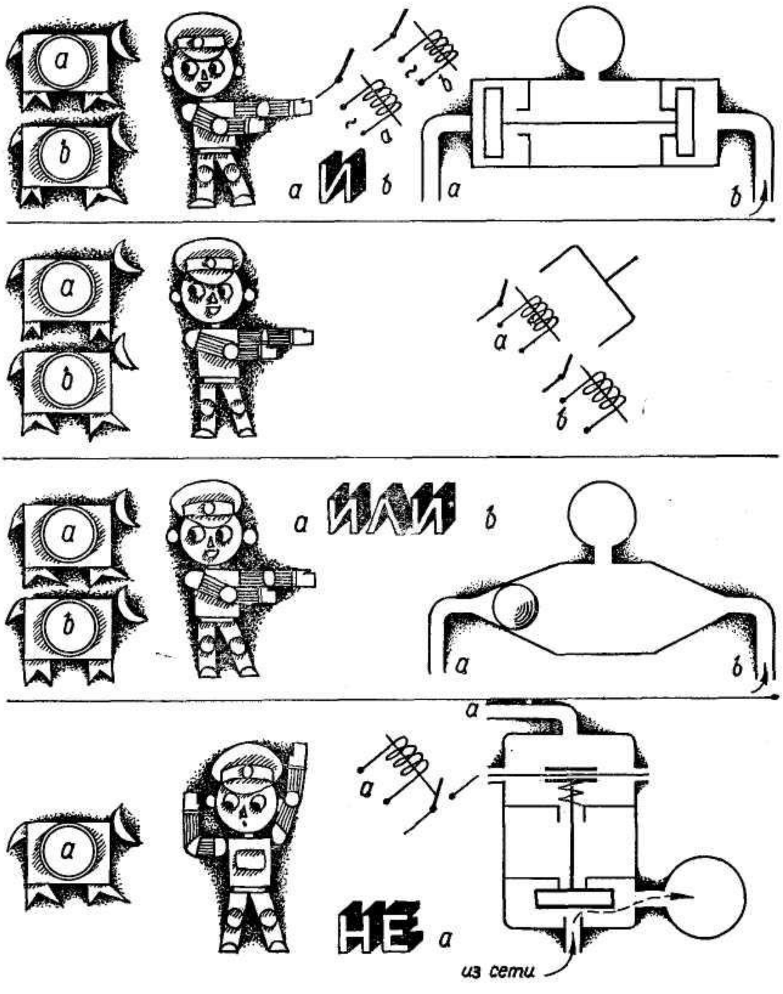

Рис. 3.4. Логические элементы И, ИЛИ, НЕ.

Логическим элементом *НЕ* в релейном варианте является переключатель. Когда напряжение в катушке *а* отсутствует, на выходе *Р* протекает ток, т. е. имеется сигнал на выходе. В пневматическом варианте этот элемент выглядит несколько сложнее; оно носит название пневматического реле. Полость над мембраной *1*, с которой соединяется пневматическая кнопка *а*, является полостью управления

реле: она же представляет собой вход. Если кнопка *а* отпущена, т. е. сигнал на входе отсутствует, то тарель *6* с толкателем *3* под действием пружины *2* находится в положении, показанном на рисунке. Резиновая прокладка *5* прижимается к верхнему седлу корпуса и закрывает проход к отверстию *4,* ведущему в атмосферу. Сжатый воздух, подводимый к нижнему отверстию 7, свободно проходит на выход реле *8*. При отсутствии сигнала на входе имеется сигнал на выходе.
Теперь нажмем кнопку *а.* Сжатый воздух начнет поступать в полость управления реле. Под действием силы давления воздуха мембрана прогибается, и тарель с толкателем движутся вниз, сжимая пружину. Резиновая прокладка отходит от верхнего седла, освобождая проход к отверстию *4,* а другая прокладка *6* перекрывает нижнее отверстие 7. Выход реле *8* оказывается связанным с атмосферой через отверстие *4.* Имеется сигнал на входе — нет сигнала на выходе.

Логические элементы, из которых складываются логические системы, чаще всего основываются на применении электронных, пневматических, пневмонических схем. В обычных пневматических устройствах процессы совершаются во много раз медленнее, чем в электронных, и поэтому логические машины на пневматике работают значительно медленнее. Однако применение пневмоники разрешает создавать устройства на сжатом воздухе, способные выполнять до двух тысяч и более операций в секунду. Вместе с тем при решении целого ряда задач автоматизации производства и, очень часто, в машиностроении, большая скорость выполнения отдельных операций вообще и не требуется. Зато пневматические устройства имеют ряд качеств, выгодно отличающих их от электронных. Они по самой своей природе взрывобезопасны, просты и надежны. Для их обслуживания и ремонта не требуется высокой квалификации.

Пример использования языка исчисления высказывний

При решении консультационной задачи на основе языка исчисления высказывний, в САК необходимо последовательно выполнить ряд следующих приемов:

1) составить подробные требования к формируемым рекомендациям;
2) установить число входов и выходов консультируемой проблемы;
3) составить таблицу функционирования консультируемой проблемы по форме;
4) на основании таблицы составить структурную формулу консультируемой проблемы;
5) осуществить минимизирование логической функции, т. е. структурной формулы консультируемой проблемы;

6) составить функциональную схему консультируемой проблемы по минимизированной логической функции.

Когда функциональная схема составлена, можно считать, что задача по формированию рекомендаций решения консультируемой проблемы будет решена.

Рассмотрим пример. На одном заводе имеются три цеха *А, В* и *С.* Электроэнергией их обеспечивает небольшая электрическая станция, на которой установлено два электрогенератора *X* и *Y.* Мощность генератора *X* в два раза выше, чем генератора *Y.*

Если в энергии нуждается один из цехов, то достаточно включить генератор *Y,* если же любые два цеха — генератор *X.* Снабжение электроэнергией всех трех цехов сразу обеспечивает одновременная работа двух генераторов. На электрической станции дежурный следит за сигналами из цехов *А, В* и *С* и соответственно регулирует включение того или иного генератора.

Стоит вопрос, нельзя ли сформировать рекомендации на создание автомата , который заменил бы дежурного по заводской электрической станции и, получая сигналы от цгхов *А, В, С,* сам бы решал, какой из генераторов включать? Приведенное задание консультанты рассматривают как словесное задание автомата, которое содержит лишние высказывание о его работеюю

Консультант, ознакомившись с таким заданием, анализирует его с помощью следующих рассуждений: будущий автомат должен получать сигналы из трех цехов *А, В* и *С,* а это значит, что у него три входа. Сигналы, вырабатываемые автоматом, направляются в два адреса: на генератор *X* и на генератор *Y.* Значит, у него два выхода. Теперь можно составить таблицу работы автомата.

A	B	C	X	Y	A	B	C	X	Y
1	1	1	1	1	0	1	0	0	1
1	1	0	1	0	0	1	1	1	0
1	0	1	1	0	0	0	1	0	1
1	0	0	0	1	0	0	0	0	0

Если в энергии нуждаются три цеха, включены оба генератора, если два цеха — только генератор *X* или генератор *Y.*

Пользуясь составленной таблицей, следует составить структурную формулу автомата. Для этого следует брать те строки в таблице, в которых выход имеет значение, равное единице. В таблице таких строк четыре для выхода *X* и четыре для выхода *Y*. Составим формулу для выхода *X*.

На выходе *X* появится сигнал при поступлении сигналов от цехов *A, B, C* одновременно или от любых двух цехов одновременно — всего в четырех случаях. Теперь несложно составить структурную формулу, по которой должен действовать дежурный или заменяющий его автомат:

$$X = abc + ab\bar{c} + a\bar{b}c + \bar{a}bc. \qquad (C)$$

Аналогично составляют формулу автомата, вырабатывающего сигнал на выход Y; эта формула будет иметь вид

$$Y = abc + a\bar{b}\bar{c} + \bar{a}b\bar{c} + \overline{ab}\,c. \qquad (Д)$$

Теперь следует минимизировать полученные выражения. Составим карту Карно для формулы *(C)*. Используя правила минимизации, получаем новую формулу для автомата, управляющего генератором *X* (рис. 3.5):

$$X = ab + ac + bc.$$

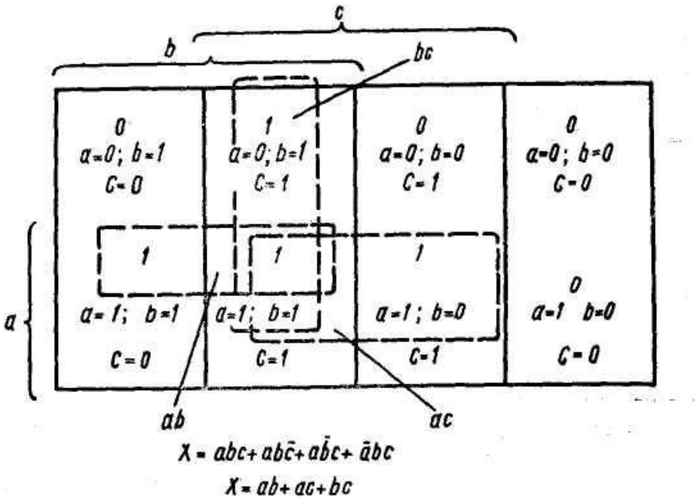

Рис. 3.5. Минимизация логического выражения для менее мощного генератора.

Наносим на карту Карно формулу *(Д)*. Так как единицы и нули на карте чередуются и нет ни одной пары смежных, которые можно было бы взять в контур, то выражение не поддается минимизации (рис. 3.6).

$$y = abc + \bar{a}b\bar{c} + \bar{a}\bar{b}c + a\bar{b}\bar{c}$$

1	0	1	0
0	1	0	1

Рис. 3.6. Карта Карно для более мощного генератора.

Структурная формула автомата, управляющего включением в работу генератора *Y*, остается прежней:

$$Y = abc + a\bar{b}\,\bar{c} + \bar{a}\,b\,\bar{c} + \overline{ab}\,c.$$

Функциональная схема автомата приведена на рис. 3.7.

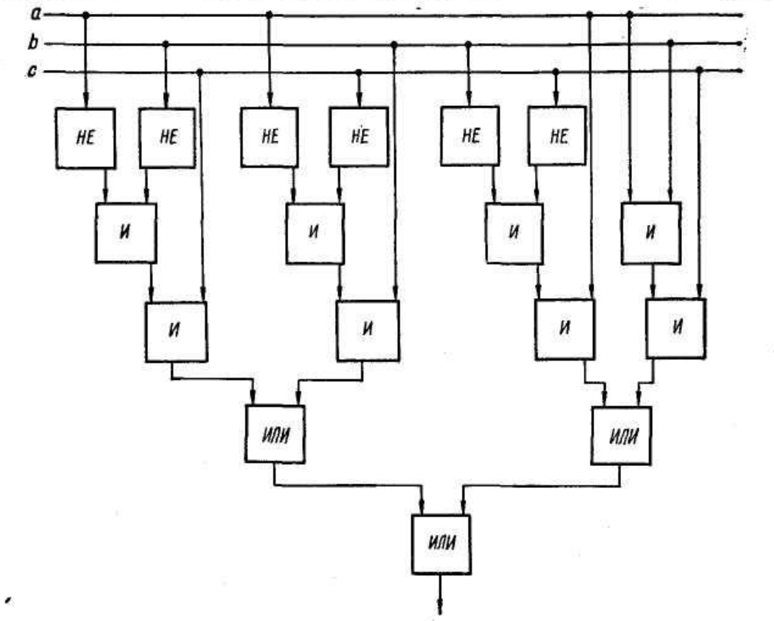

Рис. 3.7. Функциональная схема автомата, управляющего снабжением электроэнергией трех цехов

3.2.2. Исчисление предикатов как язык описания консультационных процессов

3.2.2.1. Введение в исчисление предикатов

Для осуществления автоматизации консультационных процессов необходим, как мы уже говорили, адекватный язык. Этот язык должен служить не только и не столько средством представления знаний (информации), сколько средством логического анализа консультационных задач.

Обычные человеческие языки, развивавшиеся под влиянием практических потребностей простоты общения (что далеко не всегда совместимо с точностью и надежностью логического анализа!), для этой цели плохо подходят. По этой причине желательно, даже практически необходимо, использовать в качестве языка логического анализа консультационных процессов специально созданный формализованный язык. Такой язык в противоположность обычному языку должен следовать за логической формой и воспроизводить ее даже в ущерб краткости и легкости общения, если это будет необходимо. Главной отличительной чертой такого формализованного языка является наличие в нем особой системы логического вывода или дедукции.

В качестве такого формальзованного языка, удовлетворяющего указанным требованиям, мы возьмем *исчисление предикатов.* В терминах исчисления предикатов можно сформулировать многие предложения и утверждения, выраженные на естественных языках, а также формализовать процесс рассуждений и доказательств, который является базовым консультационного процесса. Благодаря этому может быть устранен или во всяком случае резко снижен «языковый барьер» между САК и человеком.

Для того чтобы описать исчисление предикатов, мы должны воспользоваться какой-то частью обычного языка и в терминах этого языка образовать словарь и сформулировать правила формализованного языка, включая правила логического вывода. Исчисление предикатов (точнее, его синтаксис и семантика) определяется следующим очень экономным словарем символов и правилами их соединения и интерпретации:

1. **Имена**. Это — заимствованные из обычных языков выражения, служащие для непосредственного обозначения предмета. Примерами имен являются: «консультант», «мозг», «робот», «манипулятор», «консультационный процесс», «источник энергии», «искусственный интеллект» и т. д.

Следует отметить, что в одном или различных языках разные имена могут быть синонимами и выражать один и тот же смысл. С другой стороны, одно имя в различных языках или даже в одном языке (при омонимии) может выражать разный смысл.

Полное понимание языка требует знания смысла всех слов языка. Естественно потребовать, чтобы каждое имя имело точно один смысл. Такая однозначность обеспечивается в логике предикатов. А вот в обычных языках, как мы знаем, дело обстоит совсем не так.

2. Константы и переменные. *Константа* — это собственное имя. Примерами констант являются собственные имена чисел, людей, роботов, процессов, объектов.

Переменная — это символ, содержание которого совпадает с содержанием константы, за исключением лишь того, что единственный денотат константы заменен здесь возможностью различных значений переменной. С каждой переменной связана некоторая непустая область ее возможных значений. Поэтому к содержанию переменной относится в некотором смысле и содержание собственного имени области ее значений. Нужно особо подчеркнуть, что переменная в исчислении предикатов есть определенного рода символ, а не предмет (например, число), который этот символ обозначает.

3. Функции и термы. *Функция*— это операция, которая будучи применена к чему-то как к аргументу, дает некоторый объект в качестве значения функции для данного аргумента. В природе всякой функции лежит свойство быть применимой лишь к некоторым предметам.

Предметы, к которым функция применима, составляют область определения функции, а ее значения составляют область значений функции. Сама функция состоит в определении некоторого значения для каждого аргумента из области ее определения. Например, функция распознавания состоит в определении номера класса, к которому принадлежит объект, трактуемый как аргумент.

Для того чтобы обозначить значение функции для некоторого аргумента, обычно пишут имя этой функции и приписывают к нему справа имя аргумента, взятое в скобки. Так, если f — функция, а x принадлежит к области ее определения, то $f(x)$ есть значение функции f для аргумента x. Если функция применима к упорядоченной системе из n аргументов, то она называется n-арной.

Важную роль в дальнейшем играют выражения для функций, значения которых принадлежат той же области, что и их аргументы. Такие выражения называются термами. *Терм* — это выражение, построенное, исходя из символов предметных переменных и констант, с помощью символов функций. Например, если f есть n-арная функция и уже известно, что $x_1, ..., x_n$—термы, то $f(x_1, ..., x_n)$ есть терм. Содержательно терму соответствует имя некоторого предмета.

4. Предложения, высказывания и предикаты. Простейшим выражением в обычных языках является предложение. *Предложение* - это такое соединение слов, которое имеет самостоятельный смысл, т. е. выражает законченную мысль. Каждому предложению сопоставим *высказывание* (выражаемое этим предложением), предполагая при этом, что каждое высказывание или истинно, или ложно и не может быть одновременной истинно и ложно. Таким образом, высказывание можно рассматривать как величину, принимающую только два значения: «истина» (И) или «ложь» (Л).

Предположим теперь, что x представляет собой произвольный предмет из некоторого множества $\{x\}$, а $F(x)$ — какое-либо высказывание о x. Выражение $F(x)$ становится определенным, когда переменная x заменена определенным значением (именем предмета) из множества $\{x\}$. Например, выражение «x есть животное» становится вполне определенным высказыванием, если x— это робот (ложное высказывание) или если x — это собака (истинное высказывание).

Так как с нашей точки зрения каждое определенное высказывание представляет собой И или Л, то выражение $F(x)$ означает, что каждому предмету из $\{x\}$ поставлен в соответствие один из двух символов: И или Л. Иначе говоря, $F(x)$ представляет собой функцию, определенную на множестве $\{x\}$ и принимающую только два значения: И и Л. Аналогично неопределенное высказывание о двух, трех и более предметах представляет функцию со значениями И и Л от двух, трех и более переменных. Эти неопределенные высказывания (функции одной или нескольких переменных) вида $F(x_1, ..., x_{,,})$ мы будем называть *логическими функциями* или *предикатами.*

5. Элементарные (атомарные) и правильно построенные формулы. Какой бы ни был символ n-местного предиката и каков бы ни был выбор термов $x_1, ..., x_n$, (не обязательно различных), выражение $F(x_1, ..., x_n)$ мы будем называть *элементарной,* или *атомарной формулой.* Из этого определения следует, что, например, имена предметов не являются формулами.

Рассматривая элементарные формулы как величины, способные принимать только значения И и Л, мы определим над ними операции, которые позволяют из данных формул получать новые. Эти операции, по существу, выражают употребительные в обычных языках связки.

Если A и B — какие-либо данные формулы (т. е. либо элементарные формулы, либо уже построенные сложные формулы), то $A \wedge B$, $A \vee B$, $A \to B$, $A \leftrightarrow B$ также являются (сложными) формулами. Если A — данная формула, то $\neg A$ — также (сложная) формула. Первые четыре операции — бинарные (двухместные), пятая — унарная (одноместная).

Символы $\wedge$, $\vee$, $\to$, $\leftrightarrow$, $\neg$ называются соответственно *конъюнкцией, дизъюнкцией, импликацией, эквивалентностью* и *отрицанием.* Их можно читать, пользуясь словами, приведенными в правой части следующей таблицы:

$\wedge$ — «и»;

$\vee$ — «или», «... или, ... или», «и/или»;

$\to$ — «влечет», «если..., то...,» «только если»;

$\leftrightarrow$ — «равносильно», «эквивалентно», «тогда и только тогда»,

$\neg$ — «не», «неверно, что».

Прочтение сложных формул может стать неоднозначным, если не ввести скобок, указывающих, в каком порядке формулы связываются между собой. Поэтому мы будем писать $(A \to B) \to C$ или $A \to (B \to C)$, а не $A \to B \to C$. Впрочем, число скобок можно уменьшить, приписав нашим связкам убывающие «ранги» в следующем «порядке старшинства»:

$$\leftrightarrow, \to, \vee, \wedge, \neg .$$

Там, где возможны были бы два способа построения формулы, связка более высокого ранга имеет большую область действия. Так, $A \to B \wedge C$ означает $A \to (B \wedge C)$. Связка $\neg$ имеет наименьший ранг, так что, например, $\neg A \vee B$ означает $(\neg A) \vee B$, а не $\neg (A \vee B)$.

При построении сложных формул возникает вопрос, как определить значения сложных формул, зная значения простых формул, которые их составляют? Ответ на этот вопрос дается нижеследующей таблицей истинности.

A	B	$\neg A$	$\neg B$	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$
И	И	Л	Л	И	И	И	И
Л	И	И	Л	Л	И	И	Л
И	Л	Л	И	Л	И	Л	Л
Л	Л	И	И	Л	Л	И	И

Таким образом, $A \leftrightarrow B$ истинно тогда и только тогда, когда A и B имеют одинаковые значения (почему $\leftrightarrow$ и называют «эквивалентностью»); $A \to B$ ложно тогда и только тогда, когда A истинно, а B ложно; $A \vee B$ истинно тогда и только тогда, когда и A, и B истинны; $A \vee B$ ложно тогда и только тогда, когда и A, и B ложны; наконец, $\neg A$ истинно тогда и только тогда, когда A ложно.

Кроме пяти упомянутых символов-связок, в исчислении предикатов употребляются еще два символа, выражающие операции утверждения всеобщности и существования . Символ $\forall x$ называется *квантором всеобщности,* а символ $\exists x$ — *квантором существования.*

Формула $\forall xF(x)$ истинна, когда $F(x)$ истинно для каждого элемента x области $\{x\}$, и ложна в противном случае. Соответствующее ей словесное выражение будет: «для всякого x $F(x)$ истинно». Формула $\exists xF(x)$ истинна, если существует элемент области $\{x\}$, для которого $F(x)$ истинно, и ложна в противном случае. В обычном языке этой формуле соответствует выражение: «существует x такое, что $F(x)$ истинно».

Мы будем говорить, что в формулах $\forall xF(x)$ и $\exists xF(x)$ переменная x связана соответствующим квантором. Ясно, что сами эти формулы от x не зависят. Заметим, что $\neg(\forall xF(x)) \leftrightarrow \exists x \neg F(x)$.

Теперь мы можем дать определение *правильно построенной формулы* (ППФ) на языке консультационного процесса. ППФ называется выражение, которое может быть построено исходя из элементарных (атомарных) формул с помощью операций перехода от формулы A к формулам $\forall xF(x)$ и $\exists xF(x)$, от формул A и B к формулам $A \wedge B$, $A \vee B$, $A \to B$, $A \leftrightarrow B$, $\neg A$, $\neg B$. Элементарная формула или ее отрицание, входящие в ППФ, называются литерами (или литералами), а дизъюнкция литер называется простым дизъюнктом.

6. Интерпретации. ППФ имеет смысл только тогда, когда имеется какая-нибудь интерпретация входящих в нее символов. Под *интерпретацией* мы будем понимать всякую систему, состоящую из непустого множества D, называемого предметной областью (областью консультирования проблем), и какого-либо соответствия, относящего каждому символу n-местного предиката некоторое n-арное отношение в D, каждому символу функции от n аргументов некоторую n-местную операцию в D и каждой константе — некоторый элемент из D. Например, если D есть множество всех проблем, то отношение между двумя проблемами, состоящее в том, что первая из них «проблемнее» (по каким-то определенным параметрам) второй, можно отождествлять с множеством всех упорядоченных пар проблем (x, y) таких, что x проблемнее y. Таким образом, интерпретация осуществляет связь между языком консультационного процесса и описываемой им

предметной областью (консультируемой проблемой) реального мира. Она позволяет придать ППФ содержательный смысл.
При заданной интерпретации всякая ППФ (не содержащая свободных переменных) представляет собой высказывание, которое истинно или ложно. Если при данной интерпретации каждая из ППФ A_i, $i = 1, ..., n$, имеет значение И, то будем говорить, что данная интерпретация удовлетворяет системе ППФ $\{A_i\}^n_{i=1}$. ППФ A выводима (логически следует) из некоторой системы ППФ $\{A_i\}^n_{i=1}$, если каждая интерпретация, удовлетворяющая $\{A_i\}^n_{i=1}$, удовлетворяет также и A. Так, очевидно, что ППФ $\forall xF(x)$ выводима из системы ППФ $\{\forall x\neg R(x) \vee F(x), .\, yxR(x)\}$.

Согласно теореме Гёделя, если некоторая интерпретация удовлетворяет заданной системе ППФ $\{A_i\}^n_{i=1}$, то она удовлетворяет и любой ППФ *А,* выводимой из этой системы. Умение продемонстрировать, что, ППФ A выводима (логически следует) из системы ППФ $\{A_i\}^n_{i=1}$, когда это на самом деле так, играет важную роль при логическом анализе, и мы сосредоточим на нем свое внимание. Предположим, что A выводима из $\{A_i\}^n_{i=1,}$ Тогда любая интерпретация, удовлетворяющая $\{A_i\}^n_{i=1}$, удовлетворяет A, но не удовлетворяет $\neg A$. Следовательно, никакая интерпретация не удовлетворяет объединению $\{A_i\}^n_{i=1} \vee \neg A$. Если некоторая система ППФ не удовлетворяется ни при какой интерпретации, то она называется неудовлетворимой. Так, если ППФ A выводима из $\{A_i\}^n_{i=1}$, то объединение $\{A_i\}^n_{i=1} \vee \neg A$ неудовлетворимо. И наоборот, если $\{A_i\}^n_{i=1} \vee \neg A$ неудовлетворимо, то ППФ A должна логически следовать из системы ППФ $\{A_i\}^n_{i=1.}$ Именно эта концепция выводимости лежит в основе понятия логического вывода, или дедукции, в исчислении предикатов.

Универсальным методом логического вывода является так называемый *метод резолюций*, предложенный в 1965 г. Дж. Робинсоном. Этот метод замечателен тем, что он сложный процесс логического вывода сводит к последовательности очень простых операций, каждая из которых может быть легко запрограммирована.
В основе метода резолюций лежат три простых правила вывода *(резольвенции):*

1) если истинны ППФ A и $\neg A \vee B$, то истинна ППФ B (правило *modus ponens);*

2) если истинна ППФ $A \vee A$, то истинна ППФ A (правило факторизации);

3) если истинна ППФ $A(x)$, то истинна ППФ $\forall yA(y)$.

Эти правила применяются к простым дизъюнктам, на которые предварительно «раскладывается» система ППФ $\{A_i\}^n{}_{i=1} \vee \neg A$, из неудовлетворимости которой следует, что A выводима из $\{A_i\}^n{}_{i=1}$. Новые дизъюнкты, получаемые в результате применения указанных правил, называются *резольвентами*. При образовании резольвент существенную роль играет процедура *унификации,* которая для двух данных предикатов осуществляет подстановку термов вместо переменных, делающую предикаты одинаковыми. После этого к полученным ППФ применяются правила резольвенции. Например, для неудовлетворимой системы ППФ вида $\{A(x) \vee B(x), \neg B(f(z)), \neg A(f(z))\}$, используя первое правило вывода после подстановки терма $f(z)$ вместо переменной x, получим из первых двух ППФ резольвенту A ($f(z)$), которая в сочетании с третьей ППФ системы дает нулевую формулу. Таким образом, если выбрано два простых дизъюнкта и по одной литере в каждом из них, то применение правила унификации и затем правил вывода дает резольвенту. При доказательстве выводимости ППФ A, рассматриваемой как заключение (теорема), из заданной системы ППФ $\{A_i\}^n{}_{i=1}$, рассматриваемых как посылки (аксиомы), процесс образования резольвент (в котором могут принимать участие и ранее полученные резольвенты) продолжается, пока не будет получена пустая формула, означающая неудовлетворимость системы $\{A_i\}^n{}_{i=1} \vee \neg A$ и успех доказательства. Важно отметить, что число резольвент, формируемых при доказательстве любой теоремы из заданной конечной системы аксиом, конечно.

В ряде задач, которые должны решаться с использованием автоматизированных консультационных процессов, простое доказательство выводимости ППФ A, формулирующей консультационное задание, из системы ППФ $\{\{A_i\}^n{}_{i=1}$, описывающих условия выполнения этого задания, оказывается недостаточным. Примером такой задачи является консультационная задача по формировию рекомендаций планирования поведения робота. В подобного рода задачах нужно знать то значение переменной x, при котором данная ППФ $A(x)$ логически выводима из некоторой системы ППФ $\{A_i\}^n{}_{i=1}$. Иными словами, консультанты (вместе с роботом) хотели бы знать, следует ли логически ППФ $\exists x A(x)$, и если да, то каково то значение x, при котором существует решение. Заметим, что умение отыскивать такие значения для переменной, связанной квантором существования, позволяет ставить роботу вопросы весьма общего характера и осуществлять диалог с ним. Например, консультант мог бы рекомендовать спросить у робота: «Какие действия и в какой последовательности нужно совершать, чтобы собрать из деталей определенную конструкцию?». Ответом на этот вопрос будет не просто констатация факта, что сборка данной конструкции возможна, а развернутый план рекомендаций (технологический маршрут) сборки.

Рассмотрим на простейшем примере, как можно рекомендовать решать подобного рода задачи. Пусть роботу известно, что его манипулятор жестко закреплен на подвижной платформе, а платформа находится в цехе. Рекомендуется спросить: «где находится манипулятор?». В этой задаче сформулированы два «факта», которые можно записать в виде двух правильно построенных формул:

$A_1 \leftrightarrow \forall\, xP$ (платформа, $x) \rightarrow P$(манипулятор, x), $A_2 \leftrightarrow P$(платформа, цех),
где двухместному предикату $P(y, z)$ придана очевидная интерпретация: «y находится в z». На вопрос «где находится манипулятор?» робот может дать ответ, если сначала докажет, что правильно построенная формула

$A_1 \leftrightarrow \forall\, xP$ (манипулятор, x)
выводима из системы ППФ $\{A_i\}^n_{i=1}$, и затем найдет то значение x (константу), которое на самом деле «существует» и служит ответом.
Используя описанный выше рекомендованый метод резолюций, робот сначала попытается доказать неудовлетворимость системы $\{A_i\}^n_{i=1} \vee \neg P$ (манипулятор, x). (Заметим, что отрицание ППФ A есть ППФ $\neg P$ (манипулятор, x)). Процесс доказательства неудовлетворимости $\{A_i\}^n_{i=1} \vee \neg A$ представлен на дереве вывода, изображенном на рис. 3.8.

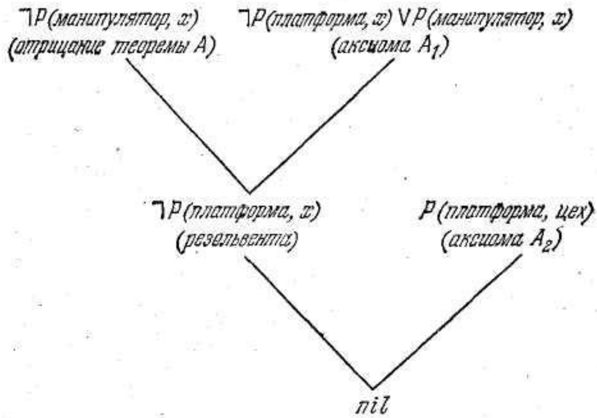

Рис. 3.8. Дерево вывода.

Из этого дерева вывода можно извлечь также ответ на вопрос консультанта: «где находится манипулятор?». Это осуществляется следующим образом. Сначала к отрицанию теоремы добавляется ее отрицание, т. е. сама теорема. В результате получается тавтология (т.е. ППФ, тождественно истинная при всех интерпретациях) вида

$$\neg P \text{ (манипулятор, } x) \vee P \text{ (манипулятор, } x).$$

Затем в соответствии со структурой дерева вывода, изображенного на рис. 3.8, вновь формируются резольвенты до тех пор, пока в корне дерева не получится некоторая ППФ, играющая роль ответа на языке робота. В нашем примере получим одну резольвенту

$$\neg P \text{ (платформа, } x) \vee P \text{ (манипулятор, } x),$$

а в корне дерева — ППФ *Р* (манипулятор, цех), в которой содержится ответ на вопрос «где находится манипулятор?». Заметим, что форма ответа на языке предикатов близка к форме теоремы-вопроса. В нашем случае единственное отличие состоит в том, что в теореме-вопросе содержится переменная, связанная квантором существования, а в ответной ППФ — константа (ответный терм).

Таким образом, описанная система логического вывода, основанная на методе резолюций, представляет собой эффективное средство для автоматического поиска доказательств (отыскания логических следствий) и извлечения ответа в терминах исчисления предикатов. Мы рассмотрели основные понятия этого исчисления и связанного с ним метода резолюций не ради них самих, а чтобы понять и продемонстрировать, как консультант, используя этот язык, может логически рассуждать, обучаться новому и адаптироваться в процессе решения консультационных задач в системе автоматизированного консультирования.

3.2.2.2. Адаптация в консультационном процессе логического вывода

Первым и неизбежным этапом применения описанной выше системы логического вывода для автоматизированного формирования рекомендаций по решения задач консультируемой проблемы, требующих логического анализа, является формулировка этих задач в терминах исчисления предикатов. Для этого нужно, прежде всего, задать предметную область (консультируемую проблему), т. е. совокупность относящихся к решаемой задаче объектов (или процессов), и выделить их существенные свойства, от которых в наибольшей степени зависит успех формирования рекомендаций по решению задач консультируемой проблемы. Далее нужно, присвоив определенный содержательный (семантический) смысл предикатным и функциональным символам, формализовать данные и условия задачи в виде ППФ, которые должны на них выполняться (т. е. истинность которых считается не требующей доказательства). Очевидно, что эти ППФ выделяют из всевозможных систем объектов (консультируемых проблем), их свойств и отношений между ними такие системы (консультируемые проблемы), для которых они выполнены.

ППФ, посредством которых мы таким образом выделяем совокупность объектов (консультируемых проблем), называются аксиомами. Если для какой-либо совокупности объектов (консультируемых проблем), их свойств и отношений некоторые аксиомы истинны, то говорят, что данная совокупность объектов (консультируемых проблем) удовлетворяет системе этих аксиом или является интерпретацией данной системы аксиом.

Таким образом, аксиомы можно рассматривать как определения системы объектов (консультируемых проблем), их свойств и отношений между ними. Делая логические выводы из аксиом, мы будем получать ППФ, истинные для любой системы объектов (консультируемых проблем), удовлетворяющей данным аксиомам.

Ясно, что соответствие между аксиомами и предметами консультируемой реальности, т. е. предметной областью, всегда имеет приближенный характер. Поэтому возникает вопрос, как узнать, действительно ли данная система аксиом определяет именно то, что было задумано, что требуется для решения задачи?

Ответ на этот вопрос связан с понятием *непротиворечивости системы аксиом.* Мы должны быть уверены, что делая всевозможные выводы из данной системы аксиом, не придем к противоречию, т. е. не выведем какие-либо несовместимые ППФ. Появление противоречия означало бы, что рассматриваемой системе аксиом не может удовлетворять никакая совокупность объектов (консультируемых проблем), и, таким образом, эти аксиомы ничего не описывают. Мы будем говорить, что система аксиом $\{A_i\}^n{}_{i=1}$ противоречива, если в ней выводима какая-либо ППФ *А,* а также и ее отрицание $\neg A$. Для проверки (доказательства) непротиворечивости системы аксиом достаточно построить какую-нибудь точную интерпретацию этой системы.

Весьма важным является свойство *независимости аксиом.* Какая-либо аксиома *А* называется *независимой* в данной системе аксиом $\{A_i\}^n{}_{i=1}$, если она невыводима из остальных аксиом этой системы. Для проверки (доказательства) независимости какой-либо аксиомы достаточно найти совокупность объектов (консультируемых проблем), удовлетворяющую всем аксиомам, кроме исследуемой, и не удовлетворяющей этой последней. Иными словами, для проверки независимости аксиомы A достаточно найти интерпретацию следующей системы аксиом: $\{A_i\}^n_{i=1}$, ┐ A.

Таким образом, система аксиом, которой пользуется робот, должна иметь точную интерпретацию в том мире объектов, свойств и отношений, в котором он функционирует. Этому требованию можно удовлетворить путем травильной формулировки тех консультационных задач, которые робот должен решать. Остановимся на этом вопросе подробнее.

Формулировка задачи на языке предикатов— это первый и наиболее ответственный этап организации его целенаправленного поведения. На этом этапе от консультанта требуются глубокие знания не только и не столько исчисления предикатов, сколько существа решаемой консультационнойзадачи, ее специфических черт, консультируемой проблемы, той цели, которая должна быть достигнута в результате реализации сформированной рекомендации. Возможна, но совершенно бессмысленна постановка на языке предикатов таких, например, задач: «рекомендую переместиться туда, сам не знаю куда», или « рекомендую найти то, сам не знаю что».

Практически весьма важно, чтобы формулировка консултационной задачи (связанная с заданием системы аксиом и теорем-заданий) была по возможности простой, не «засоренной» массой мелких, второстепенных факторов, так как учет их существенно осложняет логический анализ и делает трудно обозримыми результаты решения. Отметим две типичные трудности, которые всегда подстерегают «формулировщика» задачи. Первая — это возможность «утонуть в деталях и подробностях», т. е. «из-за деревьев не увидеть леса»; вторая — слишком огрубить задачу, или, как принято говорить в подобных случаях, «вместе с водой выплеснуть и ребенка». Ниже на примерах формулировки задач по формированю рекомендаций планирования поведения робота и распознавания сложных ситуаций мы увидим, что искусство формулировки консультационных задач на языке предикатов есть именно искусство. Здесь нет общих рецептов, а опыт консультанта в этом трудном деле приобретается постепенно.

При построении аксиом будем различать два типа предикатов, использование которых по-разному сказывается на быстроте формирования рекомендаций. Предикаты первого типа описывают простейшие свойства конкретных объектов (например, «робот находится в точке *x»,* «объект z большой» и т. п.). Предикаты второго типа определяют общую картину отношений между различными объектами (консультируемыми проблемами) и их свойствами. Один такой предикат может описывать набор свойств большого числа объектов (консультируемых проблем) (например, «если между точками *a* и *b* нет препятствий, то робот может проехать между этими точками по прямой» и т. п.). Количество подобных предикатов, необходимое для описания (с требуемой степенью подробности) данных и условий задачи, обычно невелико. Однако для сложных предикатов существенно возрастает сложность термов, участвующих в их определении.

Многие консультационные задачи часто связаны с изменением во времени свойств консультируемых проблем, которые обычно известны в начальный момент времени. В таких задачах удобно ввести предикат позиции, определяющий все «интересные» свойства всех консультируемых проблем. При этом аксиомы, описывающие изменение предиката позиции во времени, в наиболее простой форме могут быть составлены из трех литер, а именно, если имеется некоторая позиция (ситуация, проблема) и если выполняется некоторое дополнительное условие, характеризующее принципиальную возможность применения данной аксиомы, то получится новая позиция (ситуация, проблема). Основным преимуществом такого способа построения аксиом является то, что свойства, связанные между собой, определяются одним предикатом и поэтому меняются одновременно. При этом на каждом шаге формирования рекомендации учитывается все многообразие сложившейся ситуации, вследствие чего уменьшается число резольвент в процессе логического вывода.

Из дальнейшего изложения (и, в частности, из примеров) будет ясно, что эффективность системы логического вывода можно увеличить путем уменьшения числа предикатов и аксиом, определяющих данные и условия задачи. С этой целью разумно использовать ранее доказанные теоремы или ввести более сложные предикаты, образующие новые аксиомы, которые можно рассматривать как результат обучения консультанта в процессе формирования рекомендаций. Такие аксиомы, описывающие на языке исчисления предикатов приобретаемый консультантом опыт, мы будем называть *аксиомами обучения.* Введение аксиом-обучения как бы моделирует феномен мышления, о котором еще Р. Декарт писал в своем «Рассуждении о методе»: «Каждая решенная мною задача становилась образцом, который служил впоследствии для решения других задач». Образно говоря, аксиомы обучения играют роль лемм при доказательстве новых теорем, определяющих целевые условия задачи. Тем самым они позволяют оперировать более крупными «блоками» (фрагментами) доказательств, освобождая от рассмотрения многочисленных деталей, имеющих в данном доказательстве лишь вспомогательное значение. Заметим, что введение аксиом обучения позволяет консультанту увеличивать и улучшать знания о решаемом классе консультационных задач в процессе их непосредственного решения. Таким образом, аксиомы обучения являются средством обучения новым понятиям и фактам и уточнения старых.

Скорее всего, в будущем консультантам так и не удастся прийти ни к какой определенной конечной системе аксиом, рассматриваемой как окончательная. Напротив, подобно тому, как это происходит в мире живого, будут появляться (автоматически формироваться) все новые аксиомы обучения, отображающие изменения в окружающем консультанта мире и в решаемом им классе задач.

Построение системы аксиом в каждой задаче важно не само по себе, а имеет целью выявление оптимальных путей логического вывода. Под *эффективностью системы логического вывода* мы будем понимать меру успешности и поиска рекомендации. Для того чтобы выбрать количественный показатель эффективности рекомендации, нужно прежде всего спросить себя: чего мы хотим от системы логического вывода, к чему стремимся при формированни рекомендаций? Выбирая (формируя) рекомендацию, мы предпочитаем такую, которая при ее реализации обращает показатель эффективности в максимум или же в минимум.

Очень часто в качестве показателя эффективности систем логического вывода фигурируют затраты на формирование рекомендации (доказательства), которые, естественно, нужно минимизировать. Заметим, что неправильный выбор показателя эффективности очень опасен, так как он может привести к плохим рекомендациям. Рекомендации, выбранные под углом зрения неудачно выбранного показателя эффективности, могут привести к большим неоправданным потерям и затратам.

В рассматриваемом круге задач под *эффективностью* системы логического вывода будем понимать число шагов доказательства (возможно, усредненное по классу решаемых задач), т. е. число резольвент, формируемых в процессе поиска рекомендаций. С целью увеличения эффективности системы логического вывода введем некоторые ограничения на процесс образования резольвент, связанные с выбором стратегии формирования рекомендаций.

Стратегией логического вывода называется способ выбора очередной пары дизъюнктов и литер в них для образования резольвент. Именно стратегия определяет, в каком порядке будут образовываться резольвенты и, следовательно, насколько быстро будет найдена (сформирована) рекомендация. Стратегия «запускает» процесс доказательства — *начинается дедукция*: с помощью аксиом, резольвент и теорем строится та или иная конструкция доказательства. При этом стратегия решает, какие понятия и факты (аксиомы и литеры в них) несущественны, а какие — необходимы для доказательства. Таким образом, выбор и подстройка стратегии являются основным средством увеличения эффективности системы логического вывода, определяющим быстроту сходимости реализуемого ею метода поиска доказательства. Если правила резольвенции есть правила дедуктивного вывода следствий, то стратегия — это та активная часть, способная к обучению и адаптации, которая имитирует способ «мышления», например, искусственного интеллектуального робота, уровень его познаний в логике, степень его интеллектуальности.

Образно говоря, *стратегия системы логического вывода — это идея формирования рекомендации (доказательства), исходя из заданной системы аксиом, в которой заключены вес необходимые для решения задачи знания*. Если идея (стратегия) хороша, то рекомендация будет сформирована быстро. Однако рассчитывать на хорошую рекомендацию (стратегию) консультант может лишь тогда, когда в системе аксиом достаточно полно отражены не только необходимые знания о задаче (консультируемой проблеме), но и прошлый опыт решения задач в подобных проблемах. Хорошие рекомендации (стратегии) имеют своим источником прошлый опыт и ранее приобретенные знания.

Стратегия называется *полной,* если она находит (в конечное число шагов) доказательство любой ППФ, выводимой из аксиом. Примерами полных стратегий являются стратегия опорного множества, стратегия предпочтения единичным элементом, а также тривиальная стратегия полного перебора. Все перечисленные стратегии характеризуются тем, что для них предикат является «неразложимым» понятием, а критерием выбора очередной пары дизъюнктов могут быть количество предикатов в дизъюнкте, порядок их расположения и т. п. Такие стратегии, не зависящие от внутренней структуры, смысла используемых предикатов, будем называть *синтаксическими*.

Стратегию будем называть *адаптивной,* если она целенаправленно меняется (подстраивается) в процессе логического вывода в зависимости от приобретаемого опыта. Примерами адаптивных стратегий могут служить *семантические* стратегии, в которых критерий выбора очередных дизъюнктов зависит от вхождения в них определенного терма. *Согласно семантической стратегии сначала* *выбираются термы, соответствующие «интересным» проблемам, затем — предикаты, описывающие их свойства, и, наконец, ППФ, содержащие эти свойства.*

В предыдущем пункте мы отмечали, что процессу поиска доказательства может быть поставлено в соответствие дерево вывода, которое заканчивается пустым дизъюнктом, означающим конец и успех доказательства. Дерево вывода строится и ветвится под каждой резольвентой так же, как и под теоремой (точнее, ее отрицанием). Отсюда ясно, что резольвенты, не приводящие к пустому дизъюнкту, резко увеличивают число шагов доказательства, и их получение крайне нежелательно. В связи с этим задача построения адаптивной стратегии может быть переформулирована как задача отсечения ненужных (лишних) ветвей на дереве вывода. Для решения этой задачи необходимо указать критерий предпочтения ветвей.

Действительно, в процессе доказательства теоремы можно указать, как правило, несколько подходящих аксиом и несколько путей (ветвей) доказательства. Если нет критерия предпочтения одной ветви другой, приходится действовать по методу случайного поиска, что соответствует образованию пучка ветвей на дереве вывода. Введение подходящего критерия предпочтения позволяет исключить лишние тупиковые ветви и благодаря этому существенно увеличить эффективность системы логического вывода. При этом особую роль играют критерии предпочтения, формируемые в процессе решения задач. Примером такого критерия является критерий предпочтения аксиом обучения (хранящихся в памяти наряду с исходной системой аксиом), которые позволяют уменьшить исходную неопределенность относительно условий формирования рекомендаций.

Введение аксиом обучения, о которых шла речь выше, особенно эффективно в тех случаях, когда в них либо раскрывается неопределенность (т. е. содержится новая необходимая для формирования рекомендации информация), либо «запоминается» в компактной форме часто встречающийся в рассматриваемом классе задач «фрагмент» рекомендации (доказательства). В самом деле, если в процессе формирования рекомендаций потребуется доказать уже доказанную ране теорему, то критерий предпочтения аксиом обучения сократит общее число шагов доказательства по крайне мере на длину доказательства соответствующей аксиом обучения.

Важной особенностью адаптивной системы логического вывода является ее способность логически рассуждать, т. е. сводить сложное заключение к последовательности утверждений, истинность каждого из которые проверяется очень просто и чисто механически. Такая система может также не только автоматически доказывай теоремы, трактуемые как некоторые задания или вопросы, но и обучаться способам их доказательства.

Рассмотрим теперь применение адаптивной системы логического вывода для автомазированного формирования рекомендаций по решению задач планирования поведения робота, распознавания и описания сложных изображений трехмерной среды, получаемых с помощью системы атвоматизированного консультирования в условиях неопределенности.

3.2.3.3. Логические алгоритмы формирования рекомендаций по планированию поведения робота

Содержание задачи формирования рекомендаций по планированию поведения робота поясним на примере того, как эту задачу решает человек. Первое, с чем мы сталкиваемся ежедневно, — это задача утреннего одевания. Мы должны сформировать для себя рекомендации выработки плана действий, который позволит нам одеться, причем так, чтобы выполнялись естественные общепринятые ограничения (рубашку надевать необходимо, но не поверх пиджака, и т. п.). При этом время — наш основной ресурс, и выбранный план должен быть наилучшим в том смысле, в каком каждый понимает расход своего утреннего времени.

Если отбросить некоторые «несущественные» детали, план одевания должен оперировать такими предметами, как туфли, носки, брюки, рубашка, галстук, пиджак и пальто. *Рекомендация по реализации плана действий представляет собой любой порядок, в котором можно надеть эти предметы.* Всего в этом случае существует 7! = 5040 различных вариантов плана. Многие из них недопустимы, так как либо не удовлетворяют общепринятым ограничениям (рубашка поверх пиджака, носки поверх ботинок), либо непрактичны (галстук под рубашкой) и являются нереализуемыми рекомендациями. Но даже после того, как эти недопустимые рекомендации будут отброшены, все равно придется исследовать некоторое количество допустимых планов (рекомендаций). Как же выбрать окончательный (желательно, оптимальный) рекомендуемый план? Прежде всего заметим, что в рассматриваемой задаче имеется некоторая мера эффективности, некий критерий, позволяющий нам сравнивать эффективность рекомендуемых допустимых планов. Если мы можем каким-то образом сравнить значение этого критерия для различных планов, то мы сможем тем самым выбрать из них оптимальный. В данной конкретной задаче естественно рекомендовать минимизировать время, необходимое для того, чтобы одеться. Это и есть та мера эффективности рекомендации, с помощью которой можно сравнивать рекомендуемые допустимые планы. Тогда в качестве рекомендованного оптимального плана, позволяющего одеться, не нарушая общепринятых ограничений, можно выбрать следующий план: носки, рубашка, брюки, галстук, туфли, пиджак, пальто. Ясно, что при другом критерии эффективности поведения оптимальным может оказаться иной план.

Характерной особенностью консультационных задач планирования является наличие многих допустимых рекомендаций. После того, как эти рекомендации сформированы, возникает следующая задача: выбрать среди них по крайней мере одну оптимальную (в смысле определенного критерия) рекомендацию по реализации плана действий.

В рассмотренном нами простейшем примере рекомендации формировались без специальных обоснований, просто на основе опыта и здравого смысла консультанта. Оптимизация таких рекомендаций происходит как бы сама собой, в процессе жизненной практики. Если порой выбранная рекомендация окажется не самой удачной, так что же? На ошибках учатся. Нередки, правда, ситуации (соответствующие, например, планированию мероприятий, осуществляемых в первый раз), когда использовать эвристические решения, основанные на опыте и здравом смысле, просто невозможно. В подобного рода ситуациях «опыт» молчит, а «здравый смысл» легко может обмануть, если не будет опираться на математический расчет.

Но бывают рекомендации несравненно более сложные, а главное ответственные — при их реализации от них очень многое зависит. Конечно, при планировании поведения в подобного рода ситуациях можно действовать интуитивно, опираясь опять-таки на опыт и здравый смысл. Но гораздо более разумными могут оказаться рекомендации, подкрепленные количественными, математическими расчетами соответствующего плана действий. Эти предварительные расчеты помогут избежать длительного и накладного поиск рекомендации «на ощупь».

«Семь раз примерь, один раз отрежь», — говорип известная поговорка. *Формирование рекомендаций по планированию поведения* как раз и представляет собой своеобразное математическое «примеривание» к потребному будущему, позволяющее заранее оценить последствия каждой рекомендации, заранее отбросить недопустимые планы и рекомендовать наиболее удачные. Эти последние позволят установить, достаточна ли имеющаяся у нас информация для правильного формирования рекомендации, и если нет— какую информацию нужно дополнительно получать и обрабатывать. Все это позволяет при реализации рекомендованного плана экономить время, энергию и материальные средства.

Необходимость в формировании рекомендаций по планированию поведения возникает у робота при выполнении им сложных заданий в условиях большой априорной неопределенности (например, сборка сложного изделия по чертежу, поиск и транспортировка нужного объекта на неизвестной местности с препятствиями и т. п.). Задача автоматизированого формирования рекомендаций по планированию поведения решаемая на втором уровне иерархии системы управления робота, может быть переформулирована на языке исчисления предикатов как задача логического вывода (автоматического доказательства теорем). При таком подходе априорные сведения о свойствах и функциональных возможностях робота и окружающей его среды необходимо прежде всего представить в виде правильно построенных формул (ППФ). Совокупность таких ППФ мы будем называть *априорными аксиомами* и разобьем их на четыре класса:

1) сенсорные аксиомы (СА);
2) моторные аксиомы (МА);
3) аксиомы среды (АС);
4) аксиомы начальных условий (АНУ).

Сенсорные аксиомы описывают функциональные возможности информационно-измерительной системы робота, а *моторные аксиомы* — функциональные возможности исполнительных механизмов робота. *Аксиомы среды* определяют состояние и эволюцию среды, а *аксиомы начальных условий* описывают начальные состояния робота и среды. В табл. 3.1 и 3.2 приведены типичные для задачи планирования поведения робота на местности с препятствиями: термы, функции, предикаты и априорные аксиомы вместе с их интерпретацией на обычном (русском) языке.

Таблица 3.1

Функции и предикаты	Интерпретация на естественном языке
$f(a, b, s)$	Ситуация, наступающая после выполнения роботом, находящимся в ситуации s, действия «переехать из a в b по прямой».
$p(s)$	Ситуация, наступающая после выполнения манипулятором, находящимся в ситуации s, действия «погрузить объект на тележку».
$q(s)$	Ситуация, наступающая после выполнения манипулятором, находящимся в ситуации s, действия «сгрузить объект с тележки».
$G(a, b)$	Истинен, если из a в b можно проехать по прямой.
$Pos(s, x, y)$	Истинен, если в ситуации s робот находится в состоянии x, а объект — в состоянии y (предикат позиции).
$Pos(s, x, fin)$	Истинен, если в ситуации s объект находится на складе, указанном в задании, а робот — в состоянии x.
$Pos(s, fin, y)$	Истинен, если в ситуации s робот находится в конечной точке заданного маршрута, а объект — в состоянии y.

Наряду с априорными аксиомами введем *аксиомы обучения* (АО), автоматически формируемые по мере накопления роботом опыта и знаний в процессе выполнения тех или иных заданий. Задания роботу, формулируемые консультантом, будем трактовать как заключения теорем, посылками которых служат априорные аксиомы и аксиомы обучения. Заметим, что формирование рекомендаций по выбору аксиом и теорем диктуется окружающим робота миром, который он воспринимает своими органами чувств и на который воздействует своими исполнительными механизмами, а также структурно-функциональными особенностями робота и целями (задачами) его функционирования.

Таблица 3.2

Тип	Аксиомы. Логическое представление	Смысл на естественном языке
МА	$\forall s \forall x \forall y \forall z (\neg Pos(s, x, y) \vee \neg G(x, z) \vee Pos(f(x, z, s), z, y))$	Если в ситуации s, робот находится в состоянии x, а объект — в состоянии y, и из x можно переехать по прямой в z, то в ситуации $f(x, z, s)$ (т. е. после реализиции функции движения «переехать из точки x в точку z») робот окажется в точке z.
АС	$G(a, b)$	Из точки a в точку b можно проехать по прямой.
СА	$K(x)$	Объект x принадлежит k-му классу.
АНУ	$Pos(s_0, O, C_1)$	В начальной ситуации s_0 робот находится в точке O, а объект — на складе C_1.

Для автоматического доказательства теорем-заданий и извлечения ответа на языке робота целесообразно применить адаптивную систему логического вывода, описанную выше. Такая система в результате доказательства теоремы-задания (или теоремы-вопроса) сформирует рекомендации по указанию, какие действия и в какой последовательности нужно роботу совершить для выполнения задания, т. е. выдаст рекомендации по реализации искомого план поведения робота.

Продемонстрируем работу адаптивной системы логического вывода в задаче формирования рекомендаций по планированию поведения робота на примере. Пусть робот находится в цехе с оборудованием (трактуемом как препятствия), где имеются склад заготовок C_1 и склад готовых изделий C_2 (см. рис. 3.9). Вначале робот находится в точке O и перед ним ставится задача: перевезти определенный объект (который еще нужно распознать) со склада C_1 на склад C_2 (местоположение складов известно) и после этого покинуть цех через выход. Предполагается, что выход задан набором признаков (его координаты роботу неизвестны), а сенсорная система может измерять значения признаков и координаты видимых ею точек, причем она не может «видеть» сквозь препятствия. Кроме перечисленного и исходной системы аксиом, представленной в табл. 3.2, роботу ничего неизвестно.

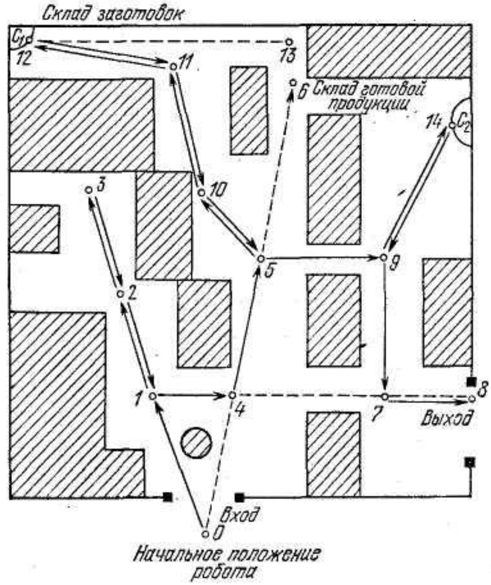

Рис. 3.9. Планирование поведения робота в незнакомом помещении.

Поскольку решение этой задачи требует дополнительной информации об обстановке в цехе, робот вначале опрашивает сенсорную систему. При этом отыскиваются все видимые границы препятствий и около них фиксируются некоторые точки *1-6*, к которым робот может проехать по прямой. В результате автоматически строятся аксиомы среды: $G(0, 1)$, $G(1, 2)$, $G(2, 3)$, $G(0, 4)$, $G(4, 5)$, $G(5, 6)$. По этим данным, а также по аксиоме начальных условий $Pos(s_0, O, C_1)$ робот пытается доказать теорему-задание $\exists s\ Pos(s, fin, fin)$ (где s— переменная, описывающая ситуацию). Однако, поскольку знаний о среде, заключенных в построенных АС, явно недостаточно (теорема не выводима из АС), ответ, т. е. рекомендация по реализации искомого плана поведения, не будет получен. В процессе логического вывода робот убеждается, что маршруты через точки *O, 1, 2, 3* и *0, 4, 5, 6* к выполнению задания не приводят. Далее, используя критерий близости к складу C_1, робот принимает решение переместиться в точку *3*. Последовательность действий на этом этапе определяется термом ситуации в резольвенте $f(2,\ 3,\ f(1,\ 2,\ f(0,\ 1,\ s_0)))$, который расшифровывается в обратном порядке и в соответствии с определением функции *f* означает: переехать из точки *O* в точку 1, затем в точки *2* и *3*. Передвигаясь согласно выбранному маршруту, робот останавливается в каждой из них и пополняет свои знания о среде посредством опроса сенсорной системы. Так возникают новые аксиомы среды (АС): $G(1, 4), G(4, 7), G(7, 8), \neg Pos(s, 8, y) \vee Pos(s, fin, y)$ (последняя аксиома среды означает, что точка *8* находится у выхода).

Приехав в точку *3*, робот не формирует ни одной новой АС, а поэтому и новой резольвенты. Он в тупике. «Осознав» это, робот вынужден развернуться и исправить те действия, которые привели его в «тупиковую» точку *3*, в обратном порядке, пока не появится первая возможность образования новой резольвенты. В результате он выбирает маршрут через точки 1, *4*, *5*, используя ранее построенные аксиомы обучения (АО) (к числу которых относятся АС и СА, формируемые в процессе функционирования робота). Так, формируя и корректируя рекомендации по реализации локальных планов поведения на основе целенаправленной переработки новой информации, робот, в конце концов, решает поставленную задачу: отыскивает (распознает) нужный объект на складе C_1, погружает (с помощью манипулятора) его на тележку, подвозит к складу C_2, сгружает объект и покидает цех через выход. При этом адаптивная система логического вывода строит 47 резольвент. Окончательный маршрут робота, реализующий выработанный план поведения, изображен на рис. 3.9 сплошными линиями со стрелками.

Рассмотренный пример формирования рекомендаций по планированию поведения робота в условиях большой априорной (начальной) неопределенности замечателен тем, что он ясно демонстрирует, что обычная (неадаптивная) система логического вывода принципиально не способна решать такого рода задачи без использования элементов обучения и адаптации. Важно отметить, что автоматическое формирование АО и адаптивная подстройка стратегии, использующей АО, не только делает задачу разрешимой, но и существенно сокращает (за счет отсечения многих тупиковых ветвей на дереве вывода) число шагов в процессе поиска плана поведения как при полной, так и при частичной информированности об условиях функционирования робота.

3.2.2.4. Алгоритмы распознавания ситуаций

Задача формирования рекомендаций по описанию, распознаванию и анализу ситуаций, котрыми являются консультируемые проблемы, является одной из центральных проблем формирования рекомендаций с использованием автоматизированных консультационных процессов. Для формализации и автоматизации решения этой задачи, как мы увидим ниже, опять-таки удобно использовать язык исчисления предикатов и адаптивную систему логического вывода. В качестве примера, на котором мы будем демонстрировать использование языка исчисления предикатов и адаптивной системы логического вывода, рассмотрим функционирование робота. Поскольку для робота наиболее важное значение имеет зрительная информация, мы сосредоточим внимание на методах формированию рекомендаций по описанию, распознаванию и анализу трехмерных сцен по их изображениям. Что же касается задач переработки речевой, тактильной и другой сенсорной информации, то они могут решаться по существу теми же методами. Трудности, возникающие при формировании рекомендаций для решения задачи распознавания отдельных предметов на сложной сцене, связаны с наличием «порочного круга»: для того чтобы распознать некоторый предмет на сцене, нужно прежде всего его «выделить», а для того чтобы «выделить» этот предмет, нужно его распознать. Это приводит к тому, что классические методы распознавания «перцептронного» или статистического типа, в этой задаче практически не применимы. На первый взгляд кажется, что выход из указанного «порочного круга» только один — полный перебор элементов изображения предмета и сцены. Однако более глубокий анализ этой задачи позволяет сформировать рекомендации по ее формулировке и решению как задачи логического вывода.

Идея предлагаемого решения основана, во-первых, на том, что применяется предварительное обучение робота путем показа ему отдельных предметов из различных классов и сообщения ему не только названия предмета, но и, возможно, его описания. Это обучение ведется консультантом в диалоговом режиме, позволяющем оперативно выявлять и исправлять ошибки робота. Во-вторых, специфика задачи ярко проявляется в большой вариативности изображений реальных предметов и сцен, которая имеет двоякую природу. С одной стороны, она порождается естественной вариативностью характеристик самих предметов, с другой стороны, — перемещением предметов в пространстве. Вариативность второго рода можно трактовать как результат действия некоторых известных преобразований изображения. Априорное знание этих преобразований позволяет построить алгоритм распознавания, инвариантный по отношению к этим преобразованиям. Благодаря этому удается не только «избавиться» от вариативности второго рода, но и существенно облегчить задачу переработки зрительной информации.

Сама эта задача подразделяется на следующие подзадачи: *описание классов (формирование понятий о классах объектов), распознавание изображения данного предмета, анализ изображения сцены (распознавание всех предметов на сцене).* Результаты решения перечисленных подзадач могут использоваться для формирования рекомендаций по моделированию внешней среды (на четвертом уровне иерархии) путем преобразования изображений предметов и сцен в их пространственное представление, а также для описания сцен на естественном языке или, наоборот, для синтеза изображения сцены по ее описанию.

Рассмотрим сначала *задачу описания классов* (формирования понятий). Эта задача решается в режиме обучения. Роботу последовательно предъявляют предметы из различных классов (и, возможно, в различных ракурсах) с указанием, к какому классу каждый такой предмет принадлежит. Эти предметы (а также их изображения) называют *эталонными*, а совокупность классифицированных предметов — *обучающей выборкой*. По этим данным робот должен автоматически сформировать описание классов в терминах тех свойств предметов, которые непосредственно измеряются сенсорной системой. Примерами этих свойств, которые мы будем называть первичными признаками, являются следующие: «красное», «ближе», «правее», «выше», «две точки соединены отрезком», «два отрезка параллельны», «зоны одинаковой яркости» и т. п. Таким образом, *изображения предметов и сцен задаются полным набором своих первичных признаков.*

Каждому признаку поставим в соответствие предикат $\xi_i(x_1, ..., x_{ni})$, где $x_1, ..., x_{ni}$ — элементы изображения ω, определяющие наличие на нем i-го признака. Тогда каждому эталону ω_h (предмету из обучающей выборки) соответствует набор значений предикатов $\xi_i^{(h)}(c_1, ..., c_{hi})$, истинных на изображении данного предмета ω_h. Здесь c_j — предметные константы, означающие фиксированные части изображения. Описанием изображения эталона ω_h будем называть конъюнкцию

$$\bigwedge_{i=1}^{P_h} \xi_i^{(h)}(c_1, \ldots, c_{ni}).$$

Поскольку к одному и тому же классу могут принадлежать несколько эталонов (например, предмет из этого класса, показанный в разных ракурсах), то описанием всех эталонных изображений, принадлежащих данному классу Ω_k, является дизъюнкция

$$\bigvee_{\omega_h \in \Omega_k} \bigwedge_{i=1}^{P_h} \xi_i^{(h)} (c_1, \ldots, c_{ni}).$$

Если теперь в этой дизъюнкции все предметные константы c_j заменить на соответствующие предметные переменные x_j, то получим ППФ, которую естественно назвать описанием класса Ω_k. Введем предикат $\sigma(k)$, означающий принадлежность изображения классу Ω_k. Тогда каждый класс Ω_k описывается аксиомой класса (АК) вида

$$\bigvee_{\omega_h \in \Omega_k} \bigwedge_{i=1}^{P_h} \xi_i^{(h)} (x_1, \ldots, x_{ni}) \rightarrow \sigma(k), \tag{3.1}$$

(которая по существу является логическим определением класса (или соответствующего ему понятия). Из вышеизложенного ясно, что АК вида (3.1) могут строиться роботом автоматически в режиме обучения по мере последовательного предъявления ему эталонов.

Практически важно, чтобы система АК обладала свойствами полноты, независимости и инвариантности в естественных смыслах. Дадим развернутое определение этих всвойств.

Систему АК будем называть *полной* на множестве изображений $\{\omega\}$, если для всякого изображения ω из этого множества найдется АК, принимающая на нем значение «истинно». Следует отметить, однако, что полнота системы АК не исключает того, что для некоторого изображения могут найтись две АК, принимающие на нем значение «истинно».

В некоторых случаях требуется, чтобы ни одно исследуемое изображение не было отнесено одновременно к нескольким классам (например, если априори известно, что распознаваемые классы изображений не пересекаются). Это требование должно быть отражено в АК. Систему АК будем называть *непротиворечивой,* если существует только одна АК, истинная на любом данном изображении. Из приведенного определения следует, что непротиворечивая система АК исключает возможность пересечения классов, описываемых этой системой аксиом. Очевидно, что непротиворечивость системы АК всегда можно эффективно проверить.

Вариативность изображений, порождаемая пространственными преобразованиями воспринимаемых предметов, а также действием разного рода помех и искажений, требует, чтобы система обладала определенной инвариантностью и помехозащищенностью. Например, всевозможные изображения стола, отличающиеся от эталонного (по которому строится соответствующая АК) сдвигом, поворотом, масштабом, а также некоторыми незначительными искажениями или помехами (лишние линии, незначительные изменения пропорций и т. п.), должны описываться одной и той же АК «стол», т. е. должны классифицироваться как эквивалентные. Систему АК будем называть *инвариантной* по отношению к заданной группе преобразований, если каждая входящая в нее аксиома принимает одно и то же значение на изображениях, отличающихся преобразованиями из этой группы. Таким образом, инвариантность системы АК позволяет «снять» охарактеризованную выше вариантность изображений и тем самым облегчить распознавание сцен по их изображениям.

Задачи распознавания и анализа изображений могут быть переформулированы как задачи логического вывода. Эти задачи решаются в режиме распознавания. Роботу предъявляются сцена, изображение $\tilde{\omega}$ которой может содержать одно или несколько изображений предметов. Эти изображения отличаются от эталонов некоторыми преобразованиями и даже могут частично перекрываться, Описание изображения сцены $S(\tilde{\omega})$ представляет собой конъюнкцию всех первичных предикатов, истинных на данном изображении $\tilde{\omega}$. В этих условиях задача отыскания на изображении $\tilde{\omega}$ изображения из определенного класса Ω_k, т. е. задача распознавания, сводится к нахождению доказательства теоремы $S(\tilde{\omega}) \to \sigma(k)$. Саму эту теорему можно трактовать как вопрос: имеется ли на данном изображении $\tilde{\omega}$ предмет из k -го класса? Ясно, что в процессе ответа на этот вопрос описание $S(\tilde{\omega})$ может быть использовано не полностью. Поэтому измерение тех или иных первичных признаков и предикатов должно производиться по мере необходимости в процессе логического вывода. Задача анализа изображения сцены заключается в распознавании на ней всех изображений предметов из различных классов. Формально эта задача сводится к последовательному доказательству теорем $S(\tilde{\omega}) \to \exists\, \sigma(k)$, $i = 1, ..., N — 1$, где $S_1(\tilde{\omega}) = S(\tilde{\omega})$, а $S_{i+1}(\tilde{\omega})$ получается из $S_i(\tilde{\omega})$ вычеркиванием всех предикатов, участвовавших в выводе i-й теоремы. Содержательно это означает, что, как только выделяется очередное изображение предмета из некоторого класса, оно при дальнейшем анализе не рассматривается. Полный анализ изображения сцены заканчивается распознаванием (и тем самым выделением) всех видимых изображений предметов, составляющих сцену.

В режиме обучения и распознавания на различных изображениях может встречаться один и тот же набор первичных признаков, характеризующих, например, фрагмент изображения. В таких случаях естественно ввести вторичные признаки, каждый из которых представляет собой некоторую совокупность из первичных признаков, а также вторичные предикаты, определяющие соответствующие фрагменты изображения как новые понятия.

Аксиомами обучения (АО) будем называть ППФ вида

$$\bigwedge_{j=1}^{r} \xi_j (x_{j1}, \ldots, x_{jm}) \to \alpha_i, \tag{3.2}$$

где $\xi_j(x_{j1}, \ldots, x_{jm})$, $j = 1, \ldots, r$ — первичные предикаты, дающие полное описание вторичного предиката α_i, т. е. определяющие некоторый фрагмент изображения как новое понятие. Использование АО позволяет не только более экономно представить АК, но и повысить эффективность системы логического вывода в процессе распознавания и анализа.

Как мы уже отмечали, универсальным средством логического вывода в исчислении предикатов является метод резолюций. Поэтому любой конкретный алгоритм распознавания или анализа определяется стратегией метода резолюций. Важно отметить, что для распознавания на изображении сцены нужного предмета не обязательно строить доказательство теоремы $S(\tilde{\omega})\to\sigma(k)$ полностью, т. е. перебирать все элементы искомого простого изображения на изображении сцены $\tilde{\omega}$. Вместо этого достаточно найти фрагмент искомого изображения, который содержится лишь в изображениях k -го класса и не содержится ни в одном изображении предметов из других классов. Именно это обстоятельство позволяет сильно ограничить число шагов логического вывода, а также распознавать частично закрытые изображения предметов.

Качество работы алгоритмов распознавания и анализа естественно характеризовать числом обращений к информационно-измерительной (сенсорной) системе с целью определения нужных признаков. Заметим, что в нашей формализации это в точности совпадает с числом шагов логического вывода, т. е. с числом резольвент, формируемых в процессе распознавания и анализа. Стратегию логического вывода будем называть *оптимальной,* если число шагов доказательства (число резольвент) минимально.

Важно, отметить, что система логического вывода способна совершенствовать алгоритмы распознавания и анализа за счет использования элементов обучения, а именно: АО и адаптивной подстройки стратегии. Адаптивная подстройка стратегии осуществляется путем ее перестройки (например, путем перестройки оптимального распознающего графа) по мере распознавания новых изображений и формирования аксиом обучения (АО). Использование АО сокращает логический вывод на длину ее описания, а процесс построения оптимального распознающего графа — на значение экспоненциальной функции от этой длины.

Проиллюстрируем описанный метод на примере формирования рекомендаций по решению задачи описания, распознавания и анализа обстановки в цехе. Предположим, что в результате предварительной фильтрации исходные изображения предметов и сцен превращаются в контурные изображения, составленные из отрезков. Введем необходимые для дальнейшего первичные признаки и предикаты, представленные в табл. 3.3. Вторичные признаки и предикаты (аксиомы обучения) представлены в табл. 3.4.

Таблица 3.3

Первичный признак	Предикат	Интерпретация
Вертикальный отрезок	$ВТ(x, y)$	Истинен, если точки x и y соединены вертикальным отрезком.
Горизонтальный отрезок	$ГР(x, y)$	Истинен, если точки x и y соединены горизонтальным отрезком.
Параллельность двух отрезков	$ПРЛ(x, y, u, v)$	Истинен, если отрезки (x, y) и (u, v) параллельны.

Таблица 3.4

Вторичный признак-понятие	Символ понятия	Логическое описание понятия
Параллелограмм	$ПР(x_1, x_2, x_3, x_4)$	$ПРЛ(x_1, x_2, x_3, x_4) \wedge ПРЛ(x_1, x_4, x_2, x_3)$
Горизонтальный параллелограмм	$ГПР(x_1, x_2, x_3, x_4)$	$ПР(x_1, x_2, x_3, x_4) \wedge ГР(x_1, x_2) \wedge ГР(x_2, x_3)$
Вертикальный прямоугольник	$ВПР(x_1, x_2, x_3, x_4)$	$ПР(x_1, x_2, x_3, x_4) \wedge ВТ(x_1, x_2) \wedge ГР(x_2, x_3)$

В режиме обучения роботу предъявляются эталонные контурные изображения токарного и сверлильного станков, представленные на рис. 3.10.

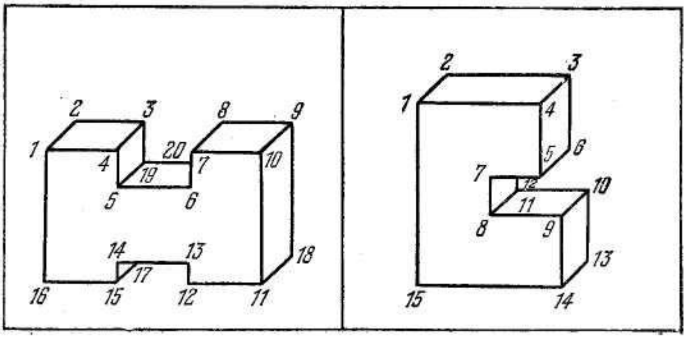

Рис. 3.10. Эталонные изображения (режим обучения).

По этим данным он строит АК, общий вид которых приведен в табл. 3.5.

Таблица 3.5

Класс-понятие	Символ класса	Логическое описание класса
Токарный станок	σ (*ТС*)	$ГПР\ (x_1, x_2, x_3, x_4) \wedge ВПР\ (x_5, x_4, x_3, x_{19}) \wedge ГР\ (x_{19}, x_{20}) \wedge ГР\ (x_5, x_6) \wedge ВТ\ (x_6, x_7) \wedge ГПР\ (x_7, x_8, x_9, x_{10}) \wedge ВПР\ (x_{11}, x_{10}, x_9, x_{18}) \wedge ГР\ (x_{12}, x_{11}) \wedge ВТ\ (x_{12}, x_{13}) \wedge ГР\ (x_{14}, x_{13}) \wedge ВТ\ (x_{15}, x_{14}) \wedge ГР\ (x_{15}, x_{17}) \wedge ГР\ (x_{16}, x_{15}) \wedge ВТ\ (x_{16}, x_1)$
Сверлильный станок	σ (*СС*)	$ГПР\ (x_1, x_2, x_3, x_4) \wedge ВПР\ (x_5, x_4, x_3, x_6) \wedge ГР\ (x_7, x_5) \wedge ВТ\ (x_8, x_7) \wedge ГПР\ (x_8, x_{11}, x_{10}, x_9) \wedge ВТ\ (x_{11}, x_{12}) \wedge ВПР\ (x_{14}, x_9, x_{10}, x_{13}) \wedge ГР\ (x_{15}, x_{14}) \wedge ВТ\ (x_{15}, x_1)$

Так в режиме обучения автоматически формируется описание классов

В режиме распознавания роботу предъявляется изображение произвольной сцены — обстановки в цехе, попавшей в «поле зрения» робота. Для определенности пусть это будет изображение сцены, представленное на рис. 3.11.

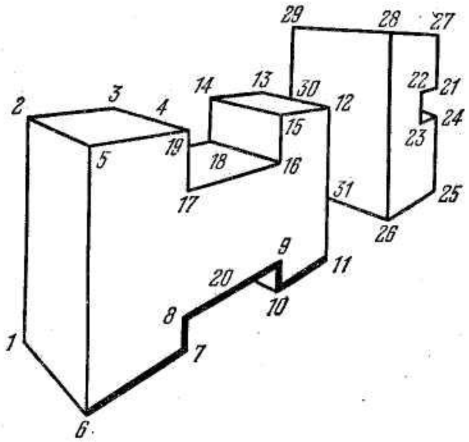

Рис. 3.11. Изображение сцены (режим распознавания).

Описание этой сцены (на языке первичных и вторичных предикатов) имеет вид

1. *ВПР (6, 5, 2, 1);* 2. *ГПР (5, 2, 3, 4);* 3. *ВТ (17, 4);* 4. *ГР (17, 16);* 5. *ВПР (16, 15, 14, 18);* 6. *ГР (19, 18); 7. ГПР (15, 14, 13, 12);* 8. *ВТ (11, 12);* 9. *ГР (10, 11);* 10. *ВТ (10, 9);* 11. *ГР (10, 20);* 12. *ГР (8, 9,);* 13. *ВТ (7, 8);* 14. ГР *(6, 7);* 15. *ВТ (30, 28);* 16. *ГР (29, 28);* 17. *ГР (29, 27);* 18. *ВТ (21, 27);* 19. *ГР (22, 21);* 20. *ВТ (23, 22);* 21. *ГР (23, 24);* 22. *ВТ (25, 24);* 23. *ГР (26, 25);* 24. *ВТ (26, 29);* 25. *ГР (26, 31).*

Предположим, что мы спрашиваем робота (или он сам задается этим вопросом): «есть ли на изображении сцены токарный станок?». Для ответа на вопрос робот пытается доказать теорему

$$\bigwedge_{j=1}^{25} \xi_j (c_{j1}, \ldots , c_{jk}) \rightarrow \sigma (\text{ТС}),$$

где ξ_j $(c_{j1}, \ldots , c_{jk})$, $j = 1, \ldots , 25$, — предикаты, входящие в приведенное выше описание изображения сцены.

Доказательство этой теоремы с помощью стратегии лозы потребовало в зависимости от порядка записи АК от 8 до 54 резольвент. Однако, как уже отмечалось, для распознавания токарного станка на изображении сцены полное доказательство соответствующей теоремы не обязательно, — возможно распознавание по фрагменту. Использование оптимальной стратегии в этом примере позволило сократить число резольвент до 5. При этом был автоматически выделен и существенно использовался в процессе логического вывода фрагмент изображения, обведенный на рис. 3.11 жирной линией. Интересно отметить, что решение той же задачи без использования АО приводит к существенному снижению эффективностисистемы логического вывода. В этом случае описание изображения сцены требует уже не 25, а 47 первичных предикатов. В процессе распознавания токарного станка с помощью стратегии лозы строится по меньшей мере 73 резольвенты.

Резюмируя содержание п. 3.2.2.1—3.2.2.4, отметим, что *язык исчисления предикатов и адаптивная система логического вывода* позволяют эффективно решать не только задачу автоматизированного формирования рекомендаций для планирования поведения робота в условиях априорной неопределенности, но и задачи описания классов, распознавания и анализа ситуаций в окружающей робота среде.

3.2.2.5. Моделирование внешней среды

Любая система, вырабатывающая целесообразное поведение, — будь то человек, животное или робот, — всегда использует знания о своих собственных функциональных возможностях и о свойствах внешней среды. О таких системах принято говорить, что они обладают «внутренней моделью внешнего мира». Наличие модели среды позволяет быстро и глубоко проникать в механизмы внешних явлений, намного облегчает процессы обучения на опыте и адаптации.

Каковы же особенности и способы представления знаний о внешнем мире в памяти робота? Прежде всего заметим, что способность к моделированию внешней среды (включая свое собственное «я») присуща лишь интеллектуальным роботам.

Окружающий нас мир настолько сложен, что не только робот, но и человек бывает доволен, если ему удается уловить и понять хотя бы некоторые самые простые из присущих этому миру закономерностей. Для этого человек строит упрощенные и идеализированные модели, освобожденные от маловажных подробностей и отражающие, как он надеется, наиболее существенные свойства рассматриваемых реальных объектов. По такой же схеме должен действовать и интеллектуальный робот, желающий создать в себе адекватную модель внешнего мира.

Восприятие внешнего мира осуществляется с помощью искусственных органов чувств робота. Это значит, что реальные ситуации описываются в памяти робота с помощью набора показаний сенсорных датчиков. Именно в терминах этих показаний — элементарных высказываний о свойствах среды и самого робота — и формируется «внутренняя модель внешнего мира».

Однако совокупность показаний сенсорных датчиков — это «сырая» информация, которую мы будем называть первичным описанием реальной ситуации. Анализ и обработка этой информации, как мы видели в предыдущем разделе, позволяют строить обобщенные описания ситуаций, называемые понятиями. В эти информационные представления о мире могут входить не только показания органов чувств (например, «красное», «горячее», «твердое» и т. п.), но и утверждения о соотношениях между этими показаниями. Одним из наиболее удобных средств для формирования и обработки этих представлений является описанное выше исчисление предикатов; вместе с адаптивной системой логического вывода. На языке исчисления предикатов свойства внешнего мира задаются с помощью элементарных предикатов и функций от показаний сенсорных датчиков, а также с помощью логических средств формирования из этих элементов сложных утверждений (описаний) в зависимости от имеющегося опыта и поступающей информации. В двух предыдущих пунктах мы видели, каким образом осуществляется уточнение (коррекция) и расширение представлений робота о внешнем мире в процессе обучения на опыте и адаптации к существующим (заранее неизвестным) условиям.

Моделирование внешнего мира в памяти робота не является самоцелью. Оно служит главным образом для «мысленного» планирования поведения и принятия решений о тех или иных целенаправленных действиях робота. При этом весьма важно, чтобы в модели внешнего мира был отражен сам робот, его структурные и функциональные свойства, особенности взаимодействия с окружающей средой. Благодаря этому робот получает возможность анализировать не только окружающую среду, в которой он функционирует, но и свое поведение в этой среде. Мы можем смело сказать, что интеллектуальный робот обладает «сознанием» и «самосознанием». Уточним эти интуитивно понятные термины.

Следуя терминологии, предложенной Д. А. Поспеловым, «сознанием» робота мы будем называть его способность отображать (моделировать) внешнюю среду в своей памяти и анализировать закономерности этой среды, а также результаты своих воздействий на среду. Под «самосознанием» будем понимать свойство робота отображать себя в модели среды и анализировать закономерности воздействия среды на свою структуру и функционирование. Именно «сознание» и «самосознание» принципиально отличают интеллектуальных роботов.

3.2.2.6. Алгоритмы построения программных движений

Задача построения программных движений (ПД) робота решается на пятом уровне иерархии его системы управления. Особенность этой задачи заключается в том, что движение исполнительных механизмов робота задается законом изменения обобщенных координат, а происходит в реальной внешней среде. При этом допустимость тех или иных конфигураций исполнительных механизмов непосредственно наблюдается в реальной же среде и никаким простым способом не выражается «на языке» пространства обобщенных координат. Другой особенностью является наличие двигательной избыточности, присущей роботам с большим числом степеней свободы. Эта избыточность, однако, полезна. Она позволяет среди множества возможных ПД отобрать наиболее «экономные», оптимальные программные движения.

Рассмотрим кратко некоторые идеи и алгоритмы построения ПД. Эти алгоритмы служат для того, чтобы робот мог, еще не совершая каких-либо реальных движений, «мысленно» рассчитать, а если нужно — скорректировать программу своих целенаправленных движений. В качестве примера рассмотрим задачу управления многозвенным манипулятором. Цель управления в этом случае заключается в отслеживании схватом некоторой траектории, рассчитанной на более высоком уровне планирования поведения. Задача осложняется наличием разного рода препятствий и конструктивных ограничений.

Программным движением называется такой закон изменения обобщенных координат, при котором достигается цель управлений, удовлетворяются конструктивные ограничения и обеспечивается обход препятствий.

Идея «глобального» подхода к построению ПД заключается в следующем. Вначале на основе информации о траектории схвата с учетом конструктивных ограничений и препятствий формируется *план движения — последовательность промежуточных конфигураций, ведущая к цели.* Затем по «кускам» (например, в классе кусочно-полиномиальных функций) строится само ПД в соответствии с этим планом.

Большой интерес представляет также «локальный» подход, основанный на моделировании так называемого тропизма.

Тропизм — это свойство, присущее простейшим живым организмам; оно заключается в направленных движениях организмов в результате действия односторонних раздражителей. Применительно к уровню построения ПД под таким *«раздражителем» будем понимать информацию о взаимном расположении манипулятора и препятствий в реальном трехмерном пространстве, а под тропизмом — движения манипулятора, направленные на удаление от препятствий*. Не приводя (ввиду громоздкости) явные формулы алгоритма построения ПД, имитирующего свойства тропизма, отметим только, что синтезируемые им целенаправленные движения реализуются за счет двигательной избыточности манипулятора, в то время как схват движется по заданной траектории.

Мы бегло охарактеризовали различные способы построения ПД манипулятора. А как строить ПД колесного или гусеничного шасси для робота, функционирующего на местности с препятствиями? Эта задача естественным образом распадается на две*: прокладка безопасного маршрута и построение закона изменения обобщенных координат, удовлетворяющего конструктивным ограничениям и обеспечивающего движение по этому маршруту.*

Рассмотрим один из подходов к построению *оптимального маршрута* на местности с препятствиями, основанный на методе динамического программирования. Для простоты предположим, что местность представляет собой плоскость, а препятствия задаются ломаными линиями. Пусть заданы координаты *исходной точки* (где находится робот) и *целевой точки* (куда он должен передвинуться). ***Задача заключается в построении маршрута (ломаной линии) из исходной точки в целевую, который не пересекает ломаных препятствий и имеет наименьшую длину***. Такой маршрут будем называть оптимальным.

Прежде всего заметим, что если сформулированная задача не имеет тривиального решения (когда маршрутом является просто отрезок, соединяющий исходную и целевую точки), вершинами ломаной наименьшей длины должны быть вершины ломаных — препятствий. Поэтому в дальнейшем будем рассматривать только эти вершины. Введем функцию f_i определяющую минимальную длину пути из некоторой точки i в целевую точку. Сразу же возникает вопрос — как найти эту функцию? Один из классических способов нахождения функции состоит в том, чтобы задать уравнения, определяющие эту функцию.

Приступая к поиску такого уравнения, зададимся вопросом: существует ли такое внутреннее свойство процесса поиска оптимального пути, которое можно использовать для получения уравнения? Ответ, к счастью, положительный. В самом деле, анализируя задачу, мы видим, что отправляясь из точки i, робот на первом шаге попадет в точку j, пока неясно, какую именно. Далее, мы замечаем, что, если робот ищет кратчайший путь от точки i до целевой точки, то в какую бы точку он ни попал, путь из точки j в целевую должен иметь наименьшую длину. Это почти очевидное свойство докажем от противного. Пусть путь из точки i в целевую точку является кратчайшим, тогда часть пути из точки j в целевую точку также должна иметь минимальную длину, так как, если бы эта часть пути не была кратчайшей, то ее можно было бы заменить более кратким путем и тем самым сократить общую длину пути, а это противоречит тому, что f_i по определению есть минимальная длина пути.

Таким образом, мы установили существенное внутреннее свойство процесса поиска оптимального маршрута: «хвост» (конец) процесса — это всегда кратчайший маршрут. Что все это дает для определения функции f_i? Обозначим через s_{ij} — длину пути между точками i и j (хотя все еще неизвестно, что это за точка j), а через f_j — наименьшую длину пути между точкой j и целевой точкой. Выбирая в качестве точки j такую точку, которая минимизирует сумму s_{ij} + f_j, получаем уравнение

$$f_i = \min_{j \neq i} [s_{ij} + f_j]. \tag{3.3}$$

Это — основное уравнение динамического программирования. Оно обладает тем свойством, что задает две функции f_i и $j(i)$. В самом деле, если мы нашли f_i, то значение j, на котором достигается минимум (3.3), как раз и указывает ту точку, в которую роботу нужно двигаться дальше. Важно отметить, что функция $j(i)$ представляет собой по существу стратегию поиска оптимального маршрута, т. е. правило, которое позволяет роботу, находящемуся в произвольной точке i, определить, куда ему следует двигаться дальше.

Трудность решения уравнения (3.3) заключается в том, что неизвестная функция входит в обе части равенства. В такой ситуации приходится прибегать к классическому методу последовательных приближений в функциональном пространстве, используя рекуррентную формулу

$$f_i^{(k+1)} = \min_{j \neq i} [s_{ij} + f_j^{(k)}], \tag{3.4}$$

где $f_i^{(k)}$ — k-е приближение искомой функции. Можно доказать сходимость алгоритма (3.4) для произвольной неотрицательной начальной функции f_i^0 с единственным ограничением, что значение функции f в целевой точке равно нулю.

Решение уравнения (3.3) можно искать и в пространстве стратегий. Этот путь представляется более естественным: «мыслить» в терминах пространства стратегий роботу более удобно, чем в терминах искусственно выбранных функций f_i. К тому же понятие стратегии сохраняет смысл и в тех случаях, когда функция не может быть определена. Какую же приближенную стратегию можно выбрать в задаче прокладки оптимального маршрута? Одна из возможных стратегий такова: робот решает непосредственно двигаться из точки i в целевую точку. Тогда получается приближение $f_i^{(0)} = s_{iц}$, где $s_{iц}$ — длина пути между точкой i и целевой точкой. Следующее приближение получится, если робот будет искать решение в классе двухзвенных ломаных. Дальнейшие приближения ищутся в классе трехзвенных, четырехзвенных и т. д. ломаных.

Одно из преимуществ выбора приближений в пространстве стратегий состоит в следующем: выбрав из общих соображений стратегию, мы тут же получаем соответствующую функцию $f_i^{(k)}$. Понятно, что $f_i^{(k)} \leq f_i$. В заключение, не входя в детали, отметим только, что описанный метод последовательных приближений в пространстве стратегий действительно сходится к решению уравнения (3.3) и допускает чрезвычайно простую программную реализацию.

После того, как оптимальный безопасный маршрут построен, можно тем или иным методом (например, с помощью кусочно-полиноминальной аппроксимации) построить программное движение колесного или гусеничного шасси робота. Алгоритмическое решение этой последней задачи обычно не вызывает никаких принципиальных затруднений.

3.2.2.7. Алгоритмы адаптивного управления движением

После того как программное движение (ПД) робота построено, возникает следующая задача: *сформировать рекомендации по нахождению закона управления исполнительными приводами, реализующий ПД*. Решение этой задачи существенно осложняется *неопределенностью* условий функционирования робота (проблемы). Природа этой неопределенности многообразна: изменение характеристик исполнительных двигателей и механизмов (например, в результате старения и износа), технологические допуски, возможные неисправности, а также изменение условий, внешних по отношению к роботу, но оказывающих на него полностью или частично неконтролируемые воздействия. Если степень неопределенности велика, то для формирования рекомендаций по построению законов управления ПД робота классические методы теории автоматического управления могут оказаться недостаточными. В подобных условиях рекомендации по широко используемым следящим приводам, параметры которых выбираются из априорных соображений о «средних» условиях функционирования, зачастую не обеспечивают реализацию ПД с требуемой точностью. Поэтому возникает необходимость сформировать рекомендации по использованию адаптивного управления, восполняющем недостающую информацию в процессе функционирования робота. Синтез *адаптивного управления ПД* осуществляется в два этапа.

Вначале в предположении, что уравнения движения робота полностью известны, рекомендуется строить закон управления, обеспечивающий близость (с любой наперед заданной точностью) реального и программного движений. Эту «неадаптивнаю» задачу можно рекомендовать решаеть методами классической теории управления. Однако воспользоваться таким «идеальным» законом управления практически нельзя, так как он зависит от ряда *варьируемых параметров* уравнения движения, значения которых неизвестны. Варьируемыми параметрами могут быть, например, масса и моменты инерции объекта манипулирования, распределение нагрузки на шасси робота, коэффициенты сцепления с грунтом и т. п. Поэтому на втором этапе на основе «идеального» закона рекомендуется построить адаптивное управление, обеспечивающее близость реального и ПД при любых возможных изменениях варьируемых параметров.

Идея адаптивного управления проста. Она заключается в замене неизвестных параметров «идеального» закона управления их оценками, которые должны целенаправленно «настраиваться» в процессе функционирования робота. Алгоритмы, по которым осуществляется «настройка» параметров управления, принято называть *алгоритмами адаптации.* Многие известные в настоящее время алгоритмы адаптации с математической точки зрения представляют собой *разностные или дифференциальные уравнения*, определяющие закон целенаправленного изменения параметров управления на основе сенсорной информации.

Ярким примером эффективных алгоритмов адаптации, допускающих простую реализацию, могут служить *конечно-сходящиеся* алгоритмы решения целевых неравенств. Смысл целевых неравенств заключается в том, что если они выполнены, то закон управления, в котором вместо неизвестных параметров используется их оценка, обеспечивает требуемую близость реального и ПД. В этом случае коррекция («настройка») параметров не производится. Если же целевые неравенства в некоторый момент времени нарушаются, то осуществляется адаптивная коррекция параметров по некоторому алгоритму. Этот алгоритм может быть выбран так, что число коррекций будет конечным (и даже минимально возможным).

Описанный конечно-сходящийся процесс адаптации хорошо согласуется с содержательным представлением об адаптации. В самом деле, с интуитивной точки зрения адаптация должна проявляться в том, что в неизменяющихся («стационарных») условиях функционирования робота (проблемы) алгоритм адаптации «работает» не все время, а с течением времени «отключается», осуществляя переход на автоматическое управление без «настройки» параметров. Лишь значительное изменение условий функционирования робота (проблемы) вызывает необходимость «включения» алгоритма адаптации для коррекции параметров управления.

3.2.3. Описанин консультационных процессов сетями Петри

3.2.3.1. Сети Петри и их модификация

Существуют различные типы консультационных систем, однако независимо от их структуры и выбранного принципа консультирования исходным материалом для консультанта является выполняемый им консультационный процесс. После распределения функций между консультируемой проблемой и консультантом и выявления взаимодействия между этими двумя компонентами консультационной системы консультационный процесс может быть описан как набор процедур, соответствующих операциям функционального процесса, который определяет и порядок их выполнения.

Выбор формализованного языка, в наибольшей степени учитывающего особенности консультационного процесса, является основной задачей начального этапа формирования рекомендаций. Одно из характерных свойств консультирования, влияющих на структуру консульационного процесса, заключается в наличии в нем множества блоков, часть которых может работать одновременно, взаимодействуя друг с другом. Поэтому формализованный язык должен прежде всего обладать средствами для описания параллелизма. Широкое распространение различного типа систем вызвало в свое время развитие различных языков, предназначенных для их описания и моделирования. Часть из них является расширением известных языков программирования, другие основаны на использовании различных графовых моделей. Не имея возможности дать полный обзор этих языков и провести их сравнение, рассмотрим более подробно язык сетей Петри, получивший широкое распространение и обладающий средствами, которые позволяют в значительной степени учитывать структуру и характер функционирования консультируемых проблем.

Сети Петри разрабатывались специально для описания и моделирования систем, состоящих из отдельных взаимодействующих блоков. При этом допускается одновременная работа нескольких блоков, и данный язык отражает присущий такой системе параллелизм. Другой характерной особенностью является то, что каждый отдельный блок сам может быть целой системой, однако ее поведение возможно описать независимо от других блоков, за исключением точно определенного взаимодействия с этими блоками. Именно такими свойствами обладают различные типы консультируемых проблем, так как ясно, что даже консультационный пункт с моноблочной структурой состоит из отдельных взаимодействующих компонент, что необходимо учитывать при более детальном описании консультационных структур.

Основными понятиями, на которых базируется представление рассматриваемой системы сетью Петри, являются события и условия. *Событие* — это некоторое действие, происходящее в системе (консультационном процессе), т. е. определенный этап функционирования или управления, выполняемый в соответствующем блоке системы. Сеть Петри описывает структуру консультационного процесса, она определяет последовательность процедур этого процесса и не предназначена для моделирования функционирования или получения консульационных параметров. Для того чтобы в консультационном процессе произошло определенное событие, необходимо выполнение соответствующих условий, множество которых задает состояние консультационного процесса. Реализация события может привести к возникновению других условий и, следовательно, к изменению состояния консультационного процесса. Такое представление консультационного процесса полностью совпадает с рассмотренными ранее правилами его функционирования, что также говорит о целесообразности использования языка сетей Петри для описания консультационного процесса, реализуемого консультантом или САК.

Наличие уже достаточно обширной в настоящее время литературы по сетям Петри, содержащей богатую библиографию, избавляет от необходимости подробного описания этого языка, его свойств и областей применения. Поэтому дадим лишь основное определение сети Петри, введем обозначения, которые далее будут использоваться, и интерпретацию символов этого языка применительно к данной структуре консультационного процесса.

В сетях Петри *события* и *условия* представляются абстрактными символами двух непересекающихся множеств — множества *переходов* и множества *позиций,* которые будем обозначать $T=\{t_0, t_1,...,t_r\}$ и $A=\{a_0, a_1,...,a_f\}$ соответственно. Переходы и позиции связаны между собой входной I и выходной O функциями. Функция I отображает переход $t_v(v = 0, 1,\ldots, r)$ в множество позиций $I(t_v)$, которые называются *входными позициями перехода.* Функция O отображает переход t_v в множество позиций $O(t_v)$, называемых *выходными позициями перехода.* Позиция a_μ является входной позицией перехода

t_v, если $a_\mu \in I(t_v)$, позиция a_μ — выходной позицией перехода, если $a_\mu \in I(t_v)$.

Таким образом, сеть Петри является четверкой $N=(A, T, I, 0)$, где A и T — конечные непересекающиеся множества.

Например, простейшая сеть Петри с пятью позициями и переходами может быть задана в таком виде:

$$A=\{a_0, a_1, a_2, a_3, a_4\};\ T=\{t_0, t_1, t_2, t_3, t_4\};$$
$$I(t_0)=a_0;\ I(t_1)=a_1;\ I(t_2)=a_2;\ I(t_3)=a_3;\ I(t_4)=a_4;$$
$$O(t_0)=a_1;\ O(t_1)=a_2;\ O(t_2)=a_3;\ O(t_3)=a_4.$$

Ясно, что функции I и O могут быть заданы также в виде матриц, которые называют матрицами *следования и предшествования* соответственно, либо в виде одной объединенной матрицы.

Наряду с приведенным способом задания сети Петри в виде системы входных и выходных функций широко используется графическое представление, которое более наглядно, но менее компактно. Граф сети Петри имеет два типа вершин, один из них соответствует позициям, обозначаемым кружочками, другой — переходам, последние обозначаются вертикальными черточками. Направленные дуги графа соединяют вершины разных типов. Если позиция a_μ является входной для перехода t_v, то дуга направлена от a_μ к t_v, если a_μ является выходной позицией перехода t_v, то дуга направлена от t_v к a_μ. В общем случае в сети Петри допускается существование кратных дуг от одной вершины графа к другой.

Таким образом, граф $G=(V, W)$ сети Петри является ориентированным двудольным мультиграфом, где V—множество вершив; W—множество (точнее, комплект) направленных дуг. Очевидно, $V=A \bigcup T$, $A \cap T = \varnothing$.

Как отмечалось, позиция сети Петри соответствует условию. Выполнение условий отмечается маркировкой позиций, так что позиция сети Петри содержит метку, если сопоставленное с ней условие выполняется, и не содержит метки в противном случае. На графе сети Петри каждая метка позиции отмечается точкой в кружке этой позиции. Однако для сетей Петри общего вида допускается наличие более одной метки в позиции, и тогда требуется иная интерпретация меток. Некоторые авторы, например, вводят понятие емкости условия, поэтому далее рассмотрим некоторые конкретные примеры использования маркированных позиций.

Маркировка сети Петри может быть задана в виде f-вектора, который определяет число меток в каждой позиции сети Петри. Таким образом, маркированная сеть Петри задается пятеркой

$N=(A, T, I, O, \mathbf{M}_0)$, где $\mathbf{M}_0$ — вектор начальной маркировки. На рис. 3.12 приведен граф маркированной сети Петри заданной (2.1), для $\mathbf{M}_0$= (1, 0, 0, 0, 0).

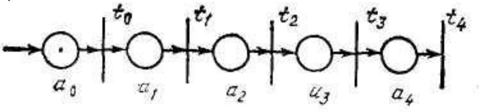

Рис. 3.12.

При больших значениях чисел в векторе маркировки вместо такого количества точек в позиции указывается просто их число. Точно так же при большой кратности дуг между двумя вершинами графа сети Петри указывается только одна дуга, которая помечается числом, равным ее кратности. Заметим, что часто, когда используется графическое изображение, говорят о сети Петри, опуская слово «граф». Распределение меток в позициях сети Петри определяет порядок ее выполнения, который зависит от последовательности реализации переходов. Переход может быть реализован, только если он становится активным. Переход является активным, когда каждая из его входных позиций содержит число меток не меньшее, чем число дуг, соединяющих ее с этим переходом. *Метки,* число которых равно кратности дуги, называют *разрешающими.* В результате реализации активного перехода удаляются разрешающие метки из всех его входных позиций, и в каждую выходную позицию помещается по одной метке для каждой дуги. Таким образом, реализация перехода заменяет маркировку сети Петри **M** в общем случае на новую маркировку **M'**, которую называют непосредственно достижимой из **M**.

Из правил реализации переходов следует, что когда один переход имеет несколько выходных позиций, после его реализации все они получат метки, т. е. произойдет распараллеливание процесса. После этого активизированные параллельные участки процесса могут выполняться независимо. Переход сети Петри, имеющий несколько входных позиций, может быть реализован, только если все эти позиции содержат метки, что соответствует выполнению всех соединяющихся в данном переходе параллельных участков процесса.

Достоинством языка сетей Петри является не только то, что он позволяет описывать параллельные консультационные процессы, но и имеет средства для задания конфликтных состояний, когда необходимо запретить одновременное развитие нескольких процессов. Если позиция сети Петри содержит q меток и является входной для $w>q$ переходов, то одновременно может быть реализовано не более q этих активизированных переходов. Вопрос о разрешении возникающего здесь конфликта, при котором одновременно несколько переходов готовы использовать метки одной и той же позиции, должен решаться дополнительными средствами.

Реализация переходов сети Петри осуществляется до тех пор, пока при очередной маркировке существует хотя бы один активный переход. Таким образом, при выполнении сети Петри возникают две связанные последовательности — реализуемых переходов и маркировок $\mathbf{M}_0$, $\mathbf{M}_{1,}$ $\mathbf{M}_2$ .... Маркировка сети Петри определяет ее состояние, так что можно говорить как о векторе маркировки, так и о состоянии сети Петри, которое соответствует состоянию задаваемого ей процесса.

Следует иметь в виду, что в первоначально разработанном в 1962 г. доктором Петри языке, т. е. в оригинальных сетях Петри, запрещалось существование кратных дуг между позициями и переходами и вектор маркировки мог содержать лишь 0 и 1. Кроме того, реализация активного перехода разрешается только в том случае, если ни одна из его выходных позиций не содержит меток. Очевидно, тогда число меток в любой позиции не может быть больше 1. Такие *позиции* называют *безопасными,* а *сеть* Петри, все позиции которой безопасны, является *безопасной.* Она имеет простые правила выполнения и конечное число состояний, равное 2^f при f позициях, однако ее описательные возможности весьма ограничены. Поэтому при необходимости отразить более сложные явления, например очередь заявок или число клиентов, ожидающих обслуживания, приходится отказываться от безопасных сетей Петри и допускать наличие нескольких меток в позиции. Если число меток в позиции не превышает некоторое значение k, то позицию называют k-безопасной или ограниченной. Для $k'\geq k$ k-безопасная позиция является и k'-безопасной. Хотя для разных позиций величина k может быть различной, всегда можно найти такое число $k_{\text{макс}}$, для которого все позиции $k_{\text{макс}}$ — безопасны. Это говорит о том, что число меток в позиции ограничено, т. е. сама позиция является ограниченной. Сеть Петри называют *ограниченной,* если все ее позиции ограничены. Для описания многих консультационных процессов приходится использовать не только позиции, содержащие множество меток, но и допускать наличие кратных дуг между позициями и переходами.

В результате этого и сформировалась рассмотренная общая модель, в соответствии с которой сеть Петри представляется ориентированным двудольным мультиграфом с произвольной начальной маркировкой позиций и кратными дугами. Однако и эта модель является ограниченной в том смысле, что она представляет только консультационную структуру, но не отражает ее связь с консультантом. Поэтому потребовались различные расширения сети Петри, в которых по-разному можно моделировать взаимодействие сети Петри с консультациооным процессом. Наиболее удачным можно считать подход, когда в сеть Петри вводится специальное множество позиций, каждая из которых представляет отдельный входной или выходной символ, характеризующий рассматриваемое взаимодействие. В позицию, соответствующую входному символу, метка помещается извне, а не в результате реализации переходов сети Петри. В свою очередь метка, появившаяся при выполнении сети Петри в позиции, соответствующей выходному символу, удаляется при выполнении определенных операций в консультируемой проблеме.

Другим существенным ограничением оригинальных сетей Петри является то, что реализация перехода рассматривается как мгновенное событие, занимающее нулевое время. Моделируемое таким переходом *событие* называют *примитивным.* Однако в реальном мире большинство событий занимают некоторое ненулевое время, т. е. они не являются примитивными и, строго говоря, не могут моделироваться переходами сети Петри. В связи с этим вводятся различные дополнения для того, чтобы моделировать сетями Петри реальные консультационные процессы с непримитивными событиями. Например, непримитивное событие может быть представлено в виде двух примитивных событий, соответствующих началу и концу непримитивного события, и условия, которое принимает истинное значение, когда происходит непримитивное событие. Другое предложение состоит в том, чтобы непримитивное событие изображать на графе сети Петри не вертикальной чертой, как примитивный переход, а прямоугольником. Но чаще такое обозначение используется в тех случаях, когда моделируются сложные иерархические консультационные пороцессы. Тогда прямоугольником обозначается не отдельный переход, а целый уровень иерархии, который может быть представлен своей сетью Петри. При описании сетями Петри реальных консультационных пороцессов, несмотря на введенное первоначально ограничение, обычно все же предполагается, что для реализации каждого перехода требуется некоторое время и отдельные события могут пересекаться во времени, но никаких ограничений на относительные времена реализации переходов не накладывается. В этом смысле говорят об асинхронности, одновременности и недетерминированности при выполнении сети Петри.

На основе предложенной первоначально оригинальной модели развилась целая теория сетей Петри, в которой исследуются их модификации, различающиеся как свойствами используемых переходов, дуг, позиций и их маркировкой, так и правилами выполнения сети Петри.

Оригинальную сеть Петри называют иногда *сетью позиция — переход.* Если в такой сети каждый переход имеет одну входную и одну выходную позиции, сеть описывает последовательный процесс и является *автоматной.* Сеть Петри, в которой каждая позиция имеет один входной и один выходной переходы, является *маркированным графом,* который задает *параллельно-последовательный процесс* без разветвлений. Если выполняются свойства автоматной сети и маркированного графа, то такая сеть Петри описывает *последовательный линейный* процесс, в котором нет ни *параллельных участков,* ни *альтернативных ветвей.*

Следует отметить, что если переходы имеют несколько входных и выходных позиций, то сеть Петри описывает не только параллелизм, но и ожидание. Необходимым условием реализации перехода является лишь наличие разрешающих меток в его входных позициях.

Более интересными и необходимыми для описания реальных консультационных пороцессов являются *сети Петри с предикатами* на переходах. В такой сети каждому переходу ставится в соответствие предикат, определяющий отношения между некоторыми переменными. Активизированный переход может быть реализован только при истинном значении сопоставленного с ним предиката. Дальнейшее расширение описательных возможностей сети Петри состоит в том, что каждому переходу сопоставляется некоторое действие, или операция. В зависимости от свойств рассматриваемого консультационного пороцесса каждая операция может иметь разный содержательный смысл. В простейшем случае используются операции над множеством переменных, от которых зависит значение предиката. В таких сетях Петри типа *предикат — действие* переход реализуется, если он активизирован и сопоставленный с ним предикат имеет истинное значение, при этом неделимым образом выполняется соответствующая переходу операция и создается новая маркировка сети Петри.

Таким образом, сетью Петри оказалось возможным описывать консультационные пороцессы, которые выполняют какие-то действия. Очевидно, эти действия занимают некоторое время, которое желательно ввести в формализованное описание консультационного пороцесса. Для этого предлагается два подхода. Первый из них используется во временных сетях Петри, где каждому переходу ставится в соответствие определенный интервал времени, равный длительности его реализации, т. е. продолжительности сопоставленной с ним операции. Когда переход становится активным, он сразу же переходит в стадию реализации, захватывая метки входных позиций и запрещая тем самым реализацию других конфликтующих с ним переходов. По истечение времени реализации перехода его выходные позиции получают метки.

Второй подход состоит в том, что каждому переходу ставится в соответствие пара величин a и b таких, что $0 \le a \le b$. Если переход t активизирован в момент времени q, то он не может быть реализован ранее, чем в момент $(q+a)$, и должен быть реализован до или в момент $(q+b)$, если только условия его активизации не изменились из-за реализации других переходов. Такие сети Петри называют *тайм-аутными*, их используют для описания и проверки протоколов вычислительных сетей, в которых применяется механизм тайм-аута, и для описания консультационных пороцессов со свойством восстановления. *Временные сети Петри в большей степени предназначены для оценки времени выполнения исследуемого консультационного пороцесса*.

В рассмотренных сетях Петри все метки, содержащиеся в позициях, являются одинаковыми, поэтому для того, чтобы задать разные типы проблемных условий, проблемных объектов или проблемных ресурсов, которые характеризуют состояние консультируемой проблемы, необходимо соответственно увеличивать число позиций сети. Избежать этого позволяют так называемые *помеченные сети Петри*, или *сети с раскрашенными позициями*. В таких сетях каждой метке сопоставляется определенный признак (цвет), так что одна позиция сети Петри может содержать метки разных цветов. Теперь для активизации перехода его входная позиция должна содержать метки с определенными признаками, которыми помечается дуга, направленная от позиции к переходу.

Обобщением сетей предикат — действие с раскрашенными позициями являются *численные сети Петри*. В этих сетях метки позиций могут иметь любую природу и величину, условия активизации и результат реализации перехода независимы, при реализации переходов изменяется как маркировка его входных и выходных позиций, так и содержимое памяти данных. Каждая входная дуга помечается условием активизации и условием реализации перехода, которые могут не совпадать, т. е. не все метки, необходимые для активизации перехода, удаляются из входной позиции после его реализации. На каждой выходной дуге указывается, сколько меток и с какими признаками помещаются в выходную позицию в результате реализации перехода. Переходу может сопоставляться некоторый предикат и операция над данными.

Очевидно, такие сети обладают большими описательными возможностями, но слишком сложны для анализа и преобразования реальных консультационных пороцессов.

Другая разновидность сетей Петри использует *дуги разных типов*. При этом различают *простые дуги*, к которым относятся активизирующая, сдерживающая, входная, выходная, и *составные дуги* — это активизирующая входная и сдерживающая выходная. *Активизирующая дуга* приводит к реализации перехода, когда позиция заполнена, и запрещает его, когда позиция пуста.

Сдерживающая дуга запрещает реализацию перехода, если позиция содержит метку, и активизирует его, если позиция пуста.

Входная дуга удаляет метку из позиции при реализации перехода; если позиция была пуста, она остается неизменной.

Выходная дуга помещает метку в позицию после реализации перехода; если позиция была заполнена, она остается неизменной. *Активизирующая входная дуга* является обычной дугой оригинальной сети Петри: для активизации перехода в позиции должна быть метка, которая удаляется после его реализации.

Сдерживающая выходная дуга также соответствует правилам выполнения оригинальной сети Петри: для реализации активного перехода необходимо, чтобы его выходная позиция была пуста, она заполняется меткой после реализации перехода.

Введение сдерживающих дуг позволяет осуществлять проверку на нуль значения какой-либо переменной, что существенно расширяет описательные возможности сетей Петри и в этом смысле приближает их к машинам Тьюринга. Однако большинство вопросов анализа самих сетей Петри с сдерживающими дугами для неограниченных позиций становится неразрешимым.

Таким образом, выбор из всего существующего множества разновидностей сетей Петри той модели, которая наиболее адекватно описывает рассматриваемый консультационный пороцесс и, обладая необходимыми описательными возможностями, сама может быть проанализирована, является достаточно сложной задачей.

Сравнение разных модификаций сети Петри позволило выделить те их свойства, которые необходимы для описания консультационных процессов и разработать некоторую обобщенную модель для описания консультационных процессов общего типа. Основной задачей при этом было отразить в исходном описании структуру и свойства консультационного процесса и максимально сохранить простые правила выполнения и анализа оригинальной сети Петри.

3.2.3.2. Консультационные процессы и их формализованное описание

Описание консультационного процесса (КП), реализуемого в САК, может быть осуществлено лишь после того, как определено суть проблемы и ее структура, т. е. число и состав функциональных блоков (ФБ) проблемы, выполняемые ими функции и их взаимодействие. При этом полнота получаемого описания зависит как от имеющихся в распоряжении консультантаа сведений об КП, так и от цели, которую он преследует, составляя описание процесса и проблемы. Для формирования рекомендаций решения некоторых консультационных задач достаточно знать только общую структуру проблемы, для других требуется задавать взаимодействие процедур консультационного процесса с ресурсами. Составление полного описания, которое необходимо иметь при консультировании проблемы, обычно является сложной итеративной консультационной процедурой, позволяющей постепенно выявлять и уточнять характер взаимодействия функциональных блоков консультируемой проблемы.

Структура консультируемой проблемы, содержащиеся в ней ФБ и, порядок их работы определяются свойствами КП, которые, в свою очередь, зависят от характеристик консультируемой проблемы. Очевидно, используемый формализованный язык должен обладать средствами для описания различных типов КП.

Простейшим можно считать последовательный линейный консультационный процесс. Такой процесс состоит из некоторого числа консультационных процедур A_0, A_1,...,A_k, последовательность выполнения которых определяется консультационным процессом и не зависит от значений и изменений каких-либо параметров консультируемой проблемы. САК, реализующая линейпый консультационный процесс, содержит оператоные функциональные блоки (ОФБ), последовательность включения которых задает КП Считая выполнение консультационной процедуры некоторым событием и сопоставив поэтому каждой консультационной процедуре переход сети Петри, линейный КП можно описать оригинальной сетью Петри, которая будет содержать t_0, t_1 ...,t_k переходов и a_0, a_1,..., a_f позиций. Появление метки в позиции a_0, соответствующее активизации консультационного процесса, вызывает последовательную реализацию всех переходов сети Петри. Хотя такое описание КП возможно, оно мало отражает действительный характер функционирования консультируемой проблемы, поскольку в оригинальной сети Петри все переходы являются примитивными, т. е. предполагается, что они реализуются мгновенно. В действительности каждая процедура имеет определенную длительность, так что можно выделить отдельные ее фазы: *начало, выполнение и окончание процедуры*. Это сртветствует *включению, работе и отключению ОФБ*, в котором выполняется процедура.

Такую процедуру A_i можно представить подсетью Петри (рис. 3.13,*а),* содержащей переходы $t_{н_i}$ и $t_{к_i}$ и дополнительные позиции z_i и w_i, которые в отличие от *внутренних* позиций назовем *внешними,* так как их маркировка зависит не только от состояния сети Петри, но и от внешних событий.

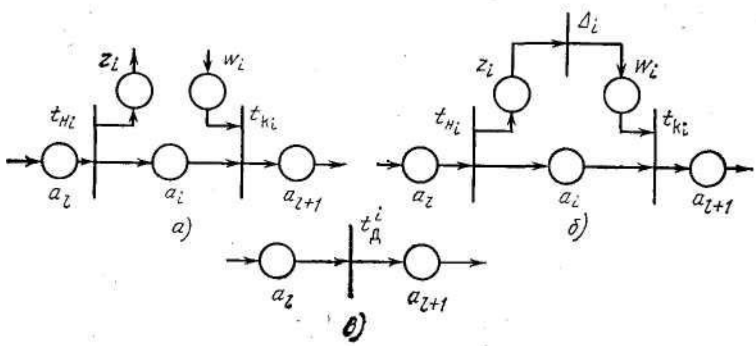

Рис. 3.13.

В результате реализации перехода $t_{н_i}$, который определяет начало выполнения процедуры A_i, появляются метки в его внешней z_i и внутренней a_i выходных позициях. Первая из них соответствует включению *ОФБ*$_i$, вторая — тому, что происходит событие, состоящее в выполнении процедуры A_i. Окончание работы *ОФБ*$_i$ отмечается появлением метки во внешней входной позиции перехода $t_{к_i}$. Этот переход реализуется, в результате чего появляется метка в его внутренней выходной позиции, означающая завершение процедуры A_i. Следует отметить, что при многопрограммном принципе реализации КП такое описание взаимодействия с ОФБ не только наглядно и удобно, но и необходимо, так как позволит разделить переходы $t_{н_i}$ и $t_{к_i}$.

Как уже отмечалось, ОФБ может иметь местное управление, что позволяет ему после получения включающего сигнала работать автономно. В это время могут выполняться процедуры других процессов, которым сопоставлены свои переходы. При этом переходы $t_{н_i}$ и $t_{к_i}$ уже не являются переходами, связанными отношением непосредственного следования. Использование внешних выходных и входных позиций позволяет отразить специфику многопрограммного управления консультационным процессом.

Однако разделение позиций на внутренние и внешние достаточно условно, поскольку, расширив рамки рассматриваемого явления, можно перевести часть внешних позиций во внутренние. Так, в нашем случае описания взаимодействия с ОФБ работу этого блока можно представить как непримитивное событие и промоделировать непримитивным переходом Δ_i. Если теперь включить этот переход в рассматриваемую сеть Петри, то внешние позиции z_i и w_i станут внутренними, так как заполнение их метками и удаление меток будет связано только с реализацией переходов этой сети Петри (рис. 3.13, *б*). В тех случаях, когда выполнение процедуры можно рассматривать как неделимое событие, для упрощения графа сети Петри ее фрагмент, содержащий переходы $t_{н_i}$, Δ_i, $t_{к_i}$ и позиции z_i, a_i, w_i, целесообразно представить одним переходом $t_д$, отметив его индексом сопоставленной с ним процедуры $t^i_д$ (рис. 3.13, *в*). Такой отмеченный переход будем называть *длительным.* Следует иметь в виду, что в отличие от примитивного для реализации длительного перехода требуется некоторое время, в течение которого уже нет метки в его входной позиции, но еще не появилась метка в выходной позиции. Никаких ограничений на длительность реализации перехода не накладывается, известно только, что это время конечно. При необходимости можно вновь заменить длительный переход подсетью Петри (рис. 3.13, *б*).

Реализация длительных переходов связана с использованием соответствующих ОФБ, которые являются ресурсами, необходимыми для выполнения консультационного процесса. Кроме ОФБ в состав САК могут входить элементы памяти и регистры для хранения промежуточных данных, вентили, шины передачи данных и управляющих сигналов и. др. Все аппаратурно-программные средства, необходимые для выполнения процедур консультационного процесса, будем называть *функциональными ресурсами* (ФР). Функциональный ресурс назовем *собственным* ресурсом процедуры A_i, если он используется только этой процедурой и не доступен для других.

Функциональный ресурс назовем *разделяемым,* если он может потребоваться нескольким процедурам. При описании консультационного процесса будем рассматривать только разделяемые ФР, так как именно они могут влиять на ход развития консультационного процесса.

Пусть для выполнения процедур КП необходимо l функциональных разделяемых ресурсов $C_1, C_2,..., C_l$. Если, имеется q экземпляров одного и того же ФР (например, q одинаковых ОФБ), то этот ресурс имеет кратность $q(C_j^q)$ и может использоваться одновременно $\alpha \leq q$ процедурами. Когда q=1, ресурс имеет два состояния: «свободен» и «занят». В общем случае ресурс имеет $q+1$ состояние.

Ресурсы, которые разделяются только процедурами рассматриваемого консультационного процесса, назовем *внутренними* или *собственными* ресурсами этого процесса. Далее будем считать, что все ФР являются внутренними.

Для выполнения процедур линейного консультационного процесса необходимы только функциональные ресурсы, и каждую процедуру A_i такого консультационного процесса можно охарактеризовать следующими *тремя множествами*: $\{C_в^i\}$—множество ФР, которыми процедура A_i уже владеет; $\{ C_з^i \}$ — множество ФР, которые процедура A_i еще запрашивает; $\{C_о^i\}$ — множество ФР, которые освобождаются после выполнения A_i.

Если, например, для процедуры A_i $\{ C_в^i \} = C_1$, $\{ C_з^i \} = C_3, C_4$ и $\{ C_о^i \} = C_1, C_4$, т. е. $A_i (\{C_1\}, \{C_3, C_4\}, \{ C_1, C_4\})$, это означает, что при выполнении процедуры A_i ей уже предоставлен ресурс C_1, еще необходимы ресурсы C_3 и C_4, а в результате ее выполнения ресурсы C_1 и C_4 освобождаются.

Будем считать, что процедура не запрашивает повторно тот ресурс, который ей предоставлен в результате выполнениядругих процедур, поэтому $\{ C_з^i \} \cap \{ C_в^i \} = \emptyset$. В частном случае, когда все необходимые ресурсы запрашиваются данной процедурой и сразу освобождаются, $\{ C_в^i \{ = \emptyset$ и $\{ C_з^i \} = \{ C_о^i \}$. Очевидно, во время выполнения процедура владеет всеми запрошенными ресурсами.

Пусть перед началом выполнения консультационного процесса и после его окончания все ресурсы свободны. Функциональному ресурсу C_j поставим в соответствие позицию c_j сети Петри и назовем ее *ресурсной позицией.* Если q_j — кратность ФР C_j, позиция c_j в начальной маркировке содержит q_j меток. Позиция c_j должна быть входной позицией длительного перехода $t_д^i$ если $C_j \in \{C_з^i\}$, и выходной позицией этого перехода, если $C_j \in \{C_о^i\}$. Ясно, что когда ресурс C_j занимается только на время выполнения процедуры A_i то позиция C_j является одновременно входной и выходной позицией перехода $t_д^i$.

Таким образом, сеть Петри, описывающая линейный последовательный консультационный процесс, содержит длительные переходы $t_д^i$, основные внутренние $a_μ$ и ресурсные внутренние c_j позиции. Будем считать, что заданное количество и распределение ресурсов позволяют последовательному консультационному процессу развиваться от начальной процедуры A_0 до конечной A_k.

На рис. 3.14 приведен пример сети Петри для консультационного процесса, состоящего из пяти последовательно выполняемых процедур A_0, A_4, A_1, A_2, A_3 при следующем распределении трех ФР C_1, C_2, C_3: $A_1(\{C_2\}, \{-\}, \{-\})$; $A_2(\{C_2\}, \{C_1\}, \{C_2\})$; $A_3(\{C_1\}, \{C_3\}, \{C_1, C_3\})$;
$A_4(\{-\}, \{C_2, C_3\}, \{C_3\})$.

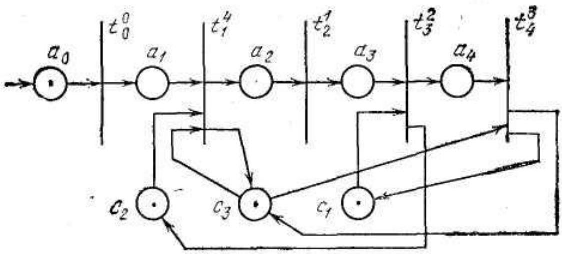

Рис. 3.14.

Рассмотрим особенности описания параллельного линейного консультационного процесса. Такой КП может иметь сложную параллельно-последовательную структуру, определяемую консультационным процессом. Сеть Петри, задающая параллельный консультационный процесс, кроме длительных переходов, соответствующих процедурам, содержит примитивные переходы для распараллеливания консультационного процесса и объединения его параллельных участков. Эти переходы будем называть соответственно *переходами распараллеливания* t_R и *соединения* t_S, число таких переходов зависит от структуры КП.

Участок консультационного процесса, соответствующий части сети Петри, заключенной между двумя последовательными переходами распараллеливания и (или) соединения будем называть *элементарным консультационным подпроцессом.* Такие консультационные подпроцессы могут быть параллельными либо последовательными, но в пределах одного элементарного консультационного подпроцесса все процедуры выполняются последовательно и его подсеть не содержит переходов распараллеливания и соединения.

Функциональные ресурсы, которые используются только процедурами одного и того же элементарного консультационного подпроцесса, будем называть *собственными* ФР этого консультационного подпроцесса. При описании консультационного процесса учитывают лишь ресурсы, разделяемые разными параллельными консультационными подпроцессами.

На рис. 3.15 приведен пример сети Петри, задающей параллельный линейный консультационный процесс при следующем распределении ФР, используемых процедурами параллельных консультационных подпроцессов:

$A_1(\{C_1\}, \{C_3\}, \{-\})$; $A_2(\{-\}, \{C_3\}, \{-\})$; $A_3(\{C_3\}, \{C_2\}, \{C_3\})$; $A_4(\{C_1, C_2\}, \{-\}, \{C_1\})$; $A_5(\{-\}, \{C_1\}, \{-\})$; $A_6(\{C_1, C_3\}, \{-\}, \{C_3\})$; $A_7(\{C_2\}, \{-\}, \{C_2\})$; $A_8(\{-\}, \{C_2\}, \{-\})$; $A_9(\{C_2\}, \{C_1\}, \{-\})$; $A_{10}(\{C_1, C_2\}, \{-\}, \{C_2\})$; $A_{11}(\{C_1\}, \{-\}, \{C_1\})$.

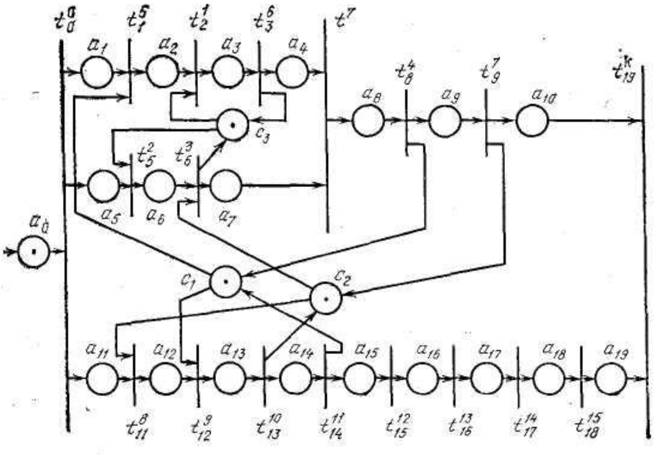

Рис. 3.15.

Заметим, что разделение ресурсов процедурами независимых параллельных консультационных процессов превращает их в один консультационный процесс с параллельными консультационными подпроцессами, взаимодействующими черезэти ресурсы. Сеть Петри (рис. 3.16) иллюстрирует этот случай.

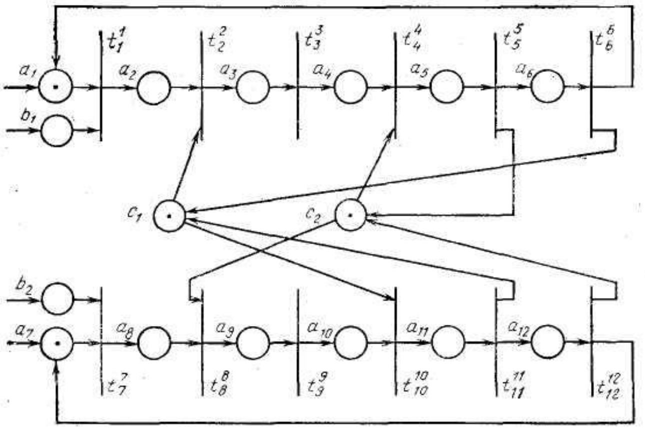

Рис. 3.16.

Разветвленный консультационный процесс в отличие от линейного содержит не одну, а несколько последовательностей выполнения процедур, каждая из которых соответствует определенному набору значений логических условий, проверяемых в ЛФБ САК. Следует четко различать альтернативные ветви последовательного консультационного процесса и параллельные участки (консультационные подпроцессы) параллельного консультационного процесса, которые могут развиваться одновременно.

Если для сети Петри параллельного консультационного процесса характерно наличие переходов распараллеливания и соединения, имеющих несколько выходных и входных позиций соответственно, то для сети Петри разветвленного консультационного процесса характерно наличие *позиций альтернативного разветвления* и *альтернативного соединения.* Позиция альтернативного разветвления имеет несколько переходов-последователей, но одновременно может быть реализован только один из них. Позиция альтернативного соединения имеет несколько переходов-предшественников и получает метку в результате реализации одного из них. Следует отметить, что позиции альтернативного соединения имеются и в сети Петри консультационного процесса, содержащего циклы.

В разветвленном консультационном процессе конфликт, соответствующий позиции альтернативного разветвления, разрешается путем сопоставления каждому из конфликтующих переходов определенного набора значений логических условий. Очевидно, эти наборы должны быть ортогональны, чтобы при выполнении консультационного процесса активизировалась только одна из альтернативных ветвей.

Как уже отмечалось, проверка значений логических условий осуществляется в ЛФБ САК. По аналогии с функциональными ресурсами ресурсы САК, которые необходимы для определения последовательности процедур консультационного процесса, назовем *логическими* ЛР. В отличие от ФР для ЛР нужно знать не только его состояние, но и значение проверяемого в нем логического условия (ЛУ), которое может изменяться либо процедурами рассматриваемого КП, либо внешними событиями.

Пусть для выполнения консультационного процесса необходимо m ЛР $D_1, D_2, \ldots, D_m$ таких, что в ЛР D_s проверяется значение логического условия p_s. Если значение логического условия p_s изменяется только процедурами КП, ЛР D_s будем называть *внутренним* ЛР. Рассмотрим вначале случай, когда все ЛР являются внутренними.

Каждую процедуру A_i будем характеризовать двумя множествами ЛУ:

$$\{P_1{}^i\} \text{ и } \{P_2{}^i\} : A_i(\{P_1{}^i\},\ \{P_2{}^i\}),$$

где $\{P^i{}_1\}$ — номера ЛУ и их значения, необходимые для выполнения процедуры A_i ; $\{P^i{}_2\}$— номера ЛУ и их значения, которые они приобретают в результате выполнения процедуры A_i.

Если, например, A_i $(\{\overline{p}_1, p_2\}, \{\overline{p}_2, p_3\})$, то это означает, что для выполнения процедуры A_i необходимо, чтобы логическое условие p_1 имело нулевое, а логическое условие p_2 — единичное значение. После выполнения A_i значение логического условия p_2 изменится и станет нулевым, а логическое условие p_3 примет единичное значение, хотя само выполнение процедуры A_i не зависит от значения этого логического условия.

Если логическое условие p_s входит в множество $\{P^i{}_2\}$, т. е. его значение изменяется процедурой A_i то эта процедура монопольно занимает ЛР D_s и во время ее выполнения он находится в состоянии *занято.* После завершения процедуры все ЛР освобождаются, в отличие от ФР, которые могут быть заняты при выполнении нескольких последовательных процедур.

Если логическое условие p_s входит только в множество $\{P^i{}_1\}$, т. е. его значение просто используется для определения активной ветви консультационного процесса, то ЛР D_s занимается немонопольно. Допускается одновременно несколько таких обращений к одному ЛР, поскольку его состояние и значение проверяемого им логического

условия при этом не изменяются. Это соответствует тому, что для выполнения процедуры A_i необходимо определенное значение логического условия p_s, которое не изменяется этой процедурой.
Если логическое условие p_s входит только в множество $\{P^i{}_2\}$, процедура A_i выполняется независимо от значений логического условия p_s, но после этого p_s принимает определенное значение, которое и указывается в $\{P^i{}_2\}$.

Очевидно, логическое условие p_s может входить как в $\{P^i{}_1\}$, так и в $\{P^i{}_2\}$. Это соответствует тому, что выполнение процедуры A_i зависит от значения p_s и, в свою очередь, влияет на это значение, т. е. монопольно занимает ЛР D_s.

Таким образом, для описания разветвленного консультационного процесса необходимо иметь средства моделирования существующих видов взаимодействия консультационного процесса с ЛР на языке сетей Петри. Прежде всего нужно представить каждый ЛР в терминах этого языка так, чтобы задать и его состояние, и значение проверяемого им ЛУ. Для этого ЛР D_s поставим в соответствие не одну, как в случае ФР, а три позиции сети Петри d_s, d_s^1, d_s^0. Позиция d_s содержит метку, если ресурс D_s свободен, т. е. ни одна из процедур консультационного процесса не владеет им монопольно. Позиция $d_s^1(d_s^0)$ содержит метку, если логическое условие p_s, проверяемое в ЛР D_s, имеет единичное (нулевое) значение.

Самым простым для описания сетью Петри является случай, когда процедура A_i изменяет значение того ЛУ, от которого и зависит ее выполнение. Например, если $A_i(\{p_s\}, \{\overline{p}_s\})$ и $A_j(\{\overline{p}_s, \{p_s\}),_j$ то входными позициями перехода $t_д^i(t_д^j)$ кроме основной внутренней позиции должны быть позиции d_s и d_s^1 $(d_s$ и $d_s^0)$. Выходными позициями перехода $t_д^i(t_д^j)$ будут кроме внутренней позиции позиции d_s и d_s^0 $(d_s$ и $d_s^1)$ (рис. 3.17, *а),* что соответствует освобождению логического ресурса D_s и изменению значения логического условия p_s.

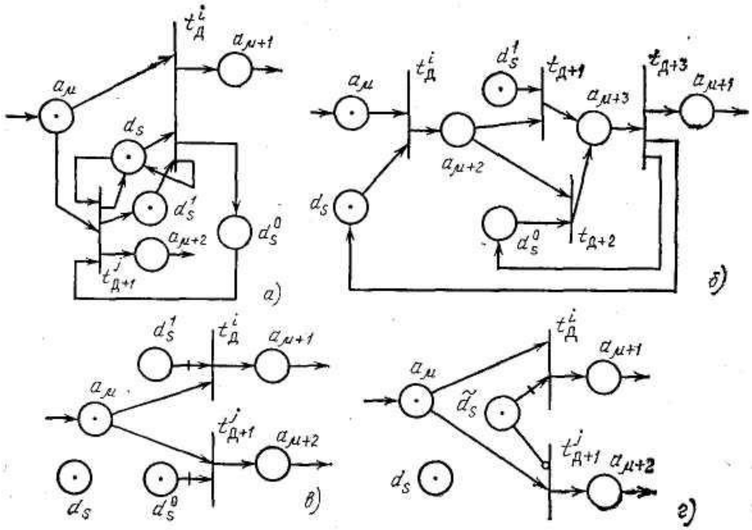

Рис. 3.17.

Другой случай, когда выполнение процедуры A_i не зависит от значения логического условия p_s, но в результате ее выполнения p_s принимает определенное значение, также может быть промоделирован существующими средствами сети Петри. Поскольку ЛР монопольно занимается на время выполнения A_i, позиция d_s является входной позицией перехода $t_д^i$. Так как перед выполнением A_i значение ЛУ p_s не проверяется, необходимо удалить метку из той позиции (d_s^0 или d_s^1), в которой она находилась, а затем поместить ее в d_s^0 (d_s^1), если в результате выполнения $A_i p_s$ принимает нулевое (единичное) значение, одновременно освободив ЛР D_s (рис. 3.17, *б*).

Наконец, третий случай, когда для выполнения процедуры A_i необходимо определенное значение ЛУ p_s, но это значение не меняется, не удается промоделировать существующими в сети Петри средствами. Кажущийся естественным способ сделать позицию d_s^1 (или d_s^0) входной и выходной позицией перехода $t_д^i$ не соответствует такому немонопольному занятию ЛР, когда одновременно несколько консультационных процессов могут считывать значение одного и того же ЛУ, не изменяя его.

Для описания этого взаимодействия введем в сеть Петри новый тип дуг, которые назовем *неизменяющими.* Если переход t_v соединен с входной позицией a_μ неизменяющей дугой, то для его реализации необходимо наличие метки в позиции a_μ, которая не удаляется из a_μ при реализации t_v. На рисунках неизменяющую дугу будем обозначать перечеркнутой стрелкой. Теперь нетрудно описать возможность немонопольного занятия ЛР, которое не приводит к изменению значения проверяемого им ЛУ. Если $A_i(\{p_s\}, \{—\})$, то позиция d_s^1 должна быть соединена с $t_д^i$ неизменяющей дугой. Когда $A_j(\{\overline{p}_s\},\{—\})$, позиция d_s^0 является входной для $t_д^i$ и соединяется с ним неизменяющей дугой (рис. 3.17, *в*). Заметим, что в этом случае при описании последовательного консультационного процесса метка позиции d_s не используется.

Очевидно, если A_i ({—}, {—}), то процедура A_i не взаимодействует с ЛР и переход $t_д^i$ может быть реализован независимо от маркировки позиций логических ресурсов.

Рассмотренный способ представления ЛР тремя позициями сети Петри хотя позволяет промоделировать все случаи взаимодействия консультационных процессов с ресурсами, однако является достаточно громоздким, так как требует введения большого числа дополнительных позиций. Это число можно сократить, если применить так называемые *сдерживающие,* или *тормозящие,* дуги. Эти дуги были введены специально для того, чтобы использовать информацию об отсутствии метки в позиции. Если переход t_v соединен тормозящей дугой с входной позицией a_μ, то он может быть реализован, только когда эта позиция не содержит метку. При этом ЛР D_s можно задать двумя позициями: d_s и $\overline{d}_s$, последняя содержит метку, когда логическое условие p_s имеет единичное значение, и не содержит ее при нулевом значении этого ЛУ.

На графе сети Петри сдерживающая дуга заканчивается не стрелкой, а маленьким кружком. При использовании таких дуг взаимодействие $A_i(\{p_s\}, \{—\})$ с логическим ресурсом D_s, показанное на рис. 3.17, *в*, будет моделироваться сетью Петри, приведенной на рис. 3.17, *г*.

Следует отметить, что позиции функциональных ресурсов и позиции состояний логических ресурсов могут не вводиться в описание последовательного процесса, тогда как маркировка позиций значений ЛУ может существенно влиять на развитие процесса.
На рис. 3.18 показана сеть Петри разветвленного последовательного процесса, все процедуры которого используют собственные ФР, выполнение процедур A_1, A_3, A_4, A_5, A_6, A_7 зависит от значений логических условий p_1, p_2 и процедуры A_1, A_3, A_7 меняют значения

логических условий: $A_1(\{p_1\}, \{\overline{p}_1\})$; $A_3(\{p_2\}, \{\overline{p}_2\})$; $A_4(\{\overline{p}_1\}, \{-\})$; $A_5(\{p_1\}, \{-\})$; $A_6(\{\overline{p}_1\}, \{-\})$; $A_7(\{\overline{p}_2\}, \{p_2\})$. Для упрощения графа сети Петри не введены позиции состояний $\overline{d}_1$ и $\overline{d}_2$.

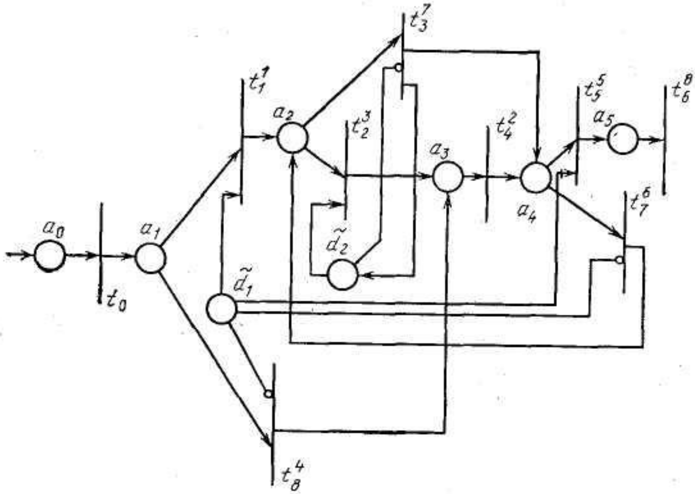

Рис. 3.18.

В общем случае консультационный процесс содержит как альтернативные ветви, так и параллельные участки; пример сети Петри такого процесса приведен на рис. 3.19.

Если все логические и функциональные ресурсы КП являются внутренними, процесс можно назвать автономным, поскольку его развитие полностью определяется начальным состоянием, т. е. начальной маркировкой его сети Петри.

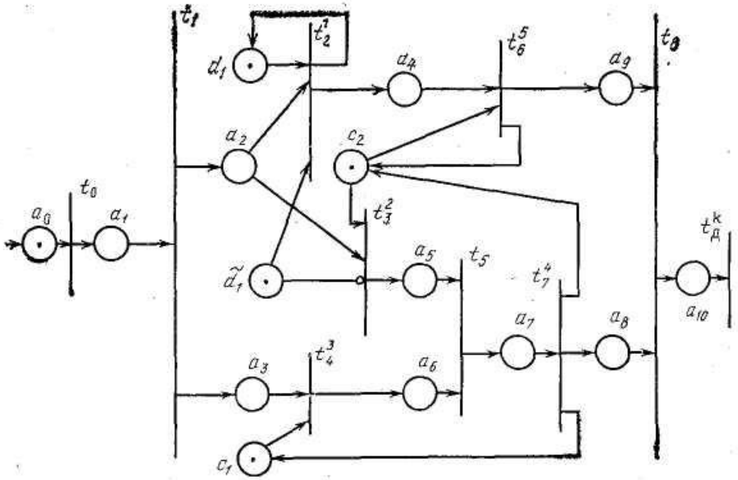

Рис. 3.19.

3.2.3.3. Обобщенная сеть Петри для описания неавтономного консультационного процесса

Для автономного консультационного процесса характерно то, что все ФР используются только процедурами данного процесса и значения всех ЛУ изменяются этими процедурами, что отражается в описании процесса. Однако во многих случаях значения ЛУ, влияющих на развитие процесса, зависят от состояния консультируемой проблемы или от состояний других процессов, т. е. определяются некоторыми внешними событиями. Такие ЛР являются внешними, их состояния и значения проверяемых ими ЛУ зависят не только от процедур рассматриваемого КП. Для описания такого процесса каждому внешнему логическому условию p_u сопоставим внешнюю позицию h_u, которая содержит метку, если p_u=1, и не содержит ее при p_u=0. В описание процедур могут входить как внутренние, так и внешние ЛУ; очевидно, последние входят только в множество $\{P_i\}$. Если процедура A_i выполняется при единичном (нулевом) значении логического условия p_u, то внешняя позиция h_u соединяется с переходом $t_д^i$ неизменяющей (сдерживающей) дугой. Позиция состояния внешнего ЛР не включается в сеть Петри КП, так как ее маркировка не может быть полностью определена развитием этого процесса. Все процедуры, выполнение которых зависит от значения внешнего ЛУ, только считывают его значение, следовательно, может быть одновременно несколько обращений к внешнему ЛР, а значение ЛУ должно оставаться неизменным на все время выполнения этих процедур. Консультационный процесс, который использует наряду с внутренними и внешние ресурсы, является неавтономным. Развитие такого процесса зависит не только от начальной маркировки внутренних позиций сети Петри, но и от текущей маркировки внешних входных позиций. Поскольку маркировка этих позиций не изменяется в результате выполнения процедур процесса, для упрощения графа сети Петри внешние входные позиции можно заменить предикатами, зависящих от внешних ЛУ. Переход $t_д^i$ помечается тем предиктом, истинное значение которого необходимо для выполнения процедуры A_i. Таким образом, сеть Петри неавтономного процесса содержит позиции внутренних ресурсов и помеченные предикатами переходы. Для простоты будем считать, что внешними могут быть только ЛР, а все ФР являются внутренними

Следует отметить еще один случай задания взаимодействия процедур процесса с ЛР когда не определено конкретное влияние выполнения процедуры на значение логического условия p_s, но известно, что такая зависимость существует. Обычно эта ситуация возникает, если ЛР является сложным ЛФБ, имеющим множество внутренних состоянии, которые изменяются под воздействием процедуры A_i, но не все эти изменения приводят к изменению значения проверяемого ЛУ. Без учета функционирования ЛР приходится считать, что процедура A_i влияет на значение ЛУ т. е. всегда монопольно занимает ЛР. Такое возможное изменение значения логического условия уровня p_s будем отмечать его безразличным ($\overline{p}_s$) значением в множестве $\{P_2^i\}$. Позицию состояния логического ресурса D_s в этом случае необходимо вводить в описание параллельного процесса, чтобы запретить одновременное выполнение процедур, использующих ресурс D_s Метка из позиции d_s удаляется на время реализации перехода $t_д^i$. Позиция $\overline{d}_s$ аналогична внешней позиции, так как маркировка ее не может быть определена состоянием процесса.

На рис. 3.20, *а* показан пример сети Петри консультационного процесса, все процедуры которого используют собственные ФР: ЛР D_1 вляется внутренним.

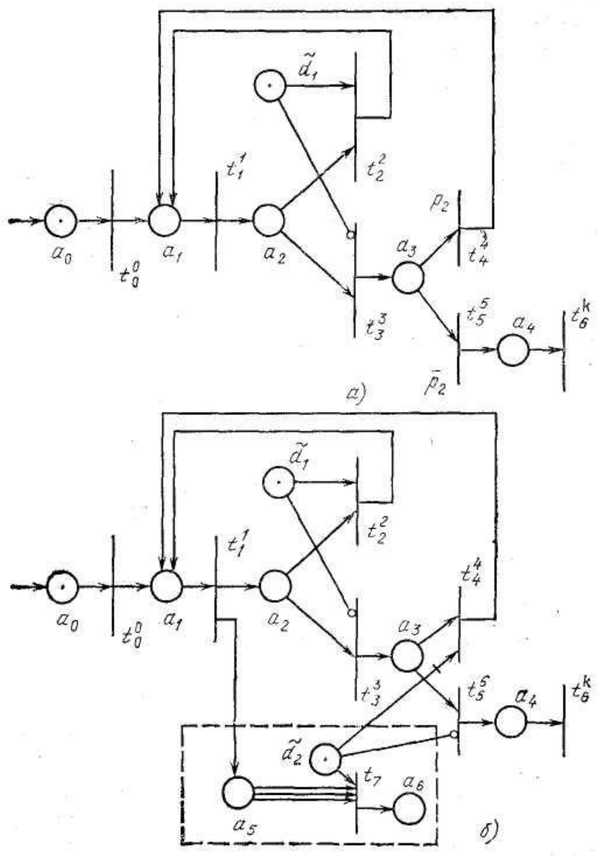

Рис. 3.20.

Состояние ЛР D_2 изменяется процедурой A_1, что может привести к изменению значения логического условия p_2, поэтому A_1 ({-}, { $\overline{p}_2$ }). Кроме того, задано следующее взаимодеиствие процедур с ЛР:

$A_2(\{p_1\}, \{\overline{p}_1\})$; $A_3(\{\overline{p}_1\}\{—\})$; $A_4(\{p_1\}, \{—\})$; $A_5(\{\overline{p}_2\}, \{—\})$. Если бы процесс был параллельным, в его описание необходимо было бы ввести позицию состояния ресурса D_2. Переходы t_4^4 и t_5^5 помечены соответствующими значениями логического условия p_2, которое приходится считать внешним.

Однако если известен алгоритм функционирования ЛР, то можно представить его в виде отдельной сети Петри и составить ее композицию с сетью Петри консультационного процесса. Пусть, например, ЛР D_2 является счетчиком, состояние которого изменяется каждый раз при выполнении процедуры A_1. Состояние счетчика сравнивается с константой k, так что при k=3 логическое условие p_2 принимает ложное значение (рис. 3.20, *б*). Теперь позиция d_2 является внутренней, так как ее маркировка полностью определяется состоянием сети Петри, состоящей из двух подсетей. Как видно, здесь потребовалось использовать k -кратную дугу между позицией a_5 и переходом t_7.

При описании КП общего вида необходимо также учитывать то обстоятельство, что одни и те же ресурсы могут запрашиваться разными параллельными подпроцессами. При этом может оказаться, что одни подпроцессы монопольно занимают ЛР D_s, а другие только используют его, т. е. занимают немонопольно. Возникает ситуация, аналогичная задаче о чтении-записи с ограниченным числом процессов чтения, которое в нашем случае не может быть больше числа параллельных подпроцессов. Любой подпроцесс записи должен исключать все другие подпроцессы чтения и записи, тогда как несколько процессов чтения могут выполняться одновременно.

Для моделирования такой ситуации позиция d_s в начальной маркировке должна содержать δ меток, если δ —максимальное число подпроцессов, немонопольно владеющих ресурсом D_s. Позиция d_s является входной и выходной для δ переходов сети Петри этих подпроцессов. Переход сети Петри подпроцесса, который занимает логический ресурс D_s монопольно, соединяется с позицией d_s дугой кратности δ. При этом не нарушается критерий взаимного исключения процессов и допускается одновременная проверка значения логического условия p_s, несколькими параллельными подпроцессами. Ясно, что для этого соответствующие переходы сетей Петри соединяются с позицией $\overline{d}_s$ либо сдерживающими, либо неизменяющими дугами.

На рис. 3.21 показан рассмотренный выше процесс.

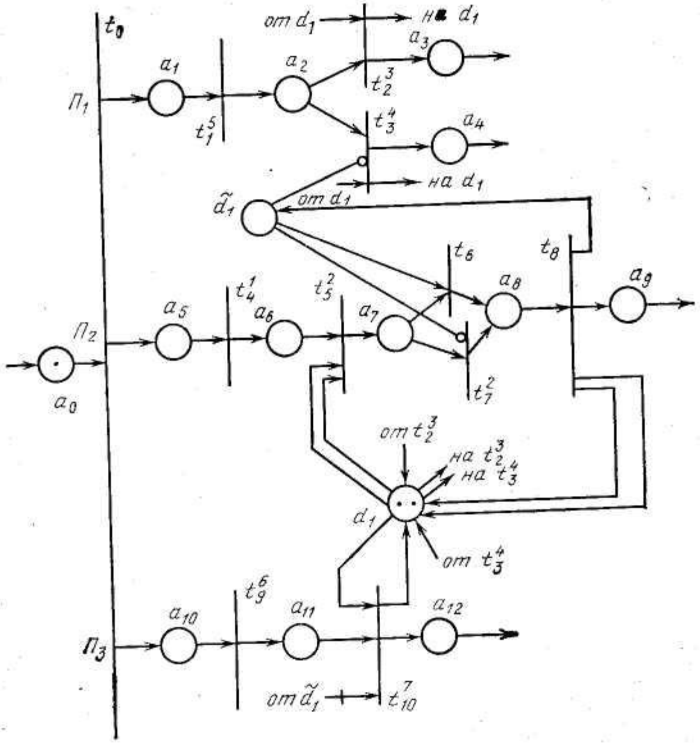

Рис. 3.21.

Здесь подпроцессы $П_1$ и $П_2$ немонопольно владеют логическим ресурсом D_1 при выполнении процедур A_3 или A_4 и A_7, в то же время выполнение процедуры A_2 подпроцесса $П_2$ не зависит от значения логического условия p_1, но присваивает ему единичное значение, т. е. $A_3(\{p_1\}, \{—\}); A_4(\{\overline{p}_1), \{—\}); A_7(\{p_1\}, \{—\}); A_2(\{-\}, \{p_1\})$.

Наличие двух меток в начальной маркировке позиции d_1 и двухкратных дуг, соединяющих эту позицию с переходом t_7^2, обеспечивает требуемое взаимодействие параллельных подпроцессов. Действительно, допустима одновременная реализация переходов t_2^3 и t_9^7, так как каждый из них удаляет по одной метке из позиции d_1. Реализация перехода t_7^2 удаляет обе метки из этой позиции, запрещая использование логического ресурса D_1 другими подпроцессами. Маркировка позиции $\overline{d}_1$ не изменяется при реализации переходов t_2^3,

t_3^4 и t_9^7, но после реализации t_7^2 эта позиция получает метку — единичное значение логического условия p_1.

На рисунке показан частный, но довольно распространенный случай разветвленного процесса с состоянием ожидания. При наличии метки в позициях a_{10} и d_1 подпроцесс $П_3$ не может развиваться (изменяться), если логическое условие p_1 имеет нулевое значение; он будет находиться в этом состоянии до тех пор, пока значение p_1 не изменится. Это частный случай процесса с циклом.

Таким образом, граф обобщенной сети Петри, предназначенной для описания неавтономного консультационного процесса общего вида, содержит длительные и примитивные переходы, основные и ресурсные внутренние позиции, основные, неизменяющие и сдерживающие дуги требуемой кратности. Каждому длительному переходу сопоставлена процедура КП. Если выполнение процедуры A_i зависит от значений внешних ЛУ, то переход $t^i_{д}$ помечается соответствующим предикатом. Реализация длительного перехода занимает некоторое время, в течение которого отсутствуют метки в его входных и выходных позициях. Примитивные переходы, к которым относятся переходы распараллеливания и соединения, реализуются мгновенно, они предназначены для задания структуры процесса. Полное состояние автономного КП определяется маркировкой основных (aμ) и внутренних ресурсных позиций сети Петри (c_j, d_s, $\bar{d}_s$). Дуги графа сети Петри определяют последовательность выполнения процедур процесса и их взаимодействие с функциональными и логическими ресурсами.

Такая обобщенная сеть Петри обладает свойствами временных сетей с переходами, помеченными предикатами и операциями, и дугами разных типов. Ее отличительной особенностью является то, что в описание процесса вводятся используемые им ресурсы и учитывается влияние процедур процесса на состояния ресурсов. Это дает возможность осуществить более полный анализ процесса и получить правильный RП с учетом ресурсов.

3.2.3.4. Получение правильного управляющего процесса

3.2.3.4.1. Граф достижимых маркировок сети Петри

Формализованное описание КП на языке сетей Петри прежде всего используется для *анализа консультационного процесса, выявления его свойств, обнаружения и устранения недопустимых состояний*. К числу таких состояний, наиболее характерных для параллельных процессов с разделяемыми ресурсами, относятся тупиковые состояния, в которых процесс прекращает развитие, не достигнув своего конечного состояния. Причины возникновения тупикового состояния могут быть различны — это и некорректно составленная структура параллельно-последовательного процесса, и неправильное распределение ФР, и недопустимое взаимодействие процедур процесса с ЛР. Поскольку возникновение тупиковых состояний в КП приводит к преждевременной остановке консультационного процесса, выявление и устранение тупиков является одной из основных задач начального этапа формирования рекомендаций.

Анализ КП основан на анализе и выявлении свойств задающей его сети Петри. Для анализа сетей Петри используют различные методы: *построение дерева достижимости, составление матричных уравнений, получение различных инвариантов для позиций и переходов.* Каждый из этих методов имеет свои достоинства и ограничения. Наиболее наглядным и удобным для пояснения является метод, основанный на применении дерева достижимости.

При выполнении сети Петри возникают две взаимно связанные последовательности — маркировок и реализуемых при этом переходов. Дерево достижимости представляет множество достижимости сети Петри, можно называть его также *графом достижимых состояний* сети Петри. Построение такого графа вытекает непосредственно из правил выполнения сети Петри. Начальная вершина графа соответствует начальной маркировке $\mathbf{M}_0$. Если маркировка $\mathbf{M}_1$ непосредственно достижима из $\mathbf{M}_0$ при реализации перехода t_l, то вершина графа $\mathbf{M}_0$ соединяется с вершиной, соответствующей маркировке $\mathbf{M}_1$, дугой, помеченной переходом t_l. Затем рассматриваются все маркировки, достижимые из $\mathbf{M}_1$, и т. д. Поскольку при одной и той же маркировке может быть активизировано несколько альтернативных переходов, то образуется целое дерево достижимости.

В процессе построения такого графа достижимых маркировок может возникнуть маркировка, уже порожденная на предыдущем шаге, тогда в графе образуется цикл. Кроме того, в графе могут возникать маркировки, различающиеся только числом меток в одной и той же позиции. Предлагается возможное бесконечное число меток в маркировках такого типа представить с помощью специального символа ω, описывается алгоритм построения дерева достижимости, доказывается, что этот алгоритм всегда заканчивает свою работу и что дерево достижимости сети Петри конечно.

Очевидно, наличие символа ω в графе достижимых маркировок свидетельствует о том, что число различных маркировок может быть бесконечным, несмотря на конечное число вершин графа. Сеть Петри, для которой получен такой граф, является неограниченной. Если сеть

Петри ограничена и символ ω отсутствует в графе ее достижимых состояний, то эта сеть задает систему с конечным числом состояний. Именно к таким системам относится рассматриваемая нами САК, поэтому сеть Петри, описывающая КП, реализуемый в САК, должна иметь конечное число состояний.

Граф достижимых маркировок сети Петри N будем обозначать G_N. В вершине графа G_N, соответствующей маркировке $\mathbf{M}_i$, будем выписывать те позиции сети Петри, которые содержат метки при этой маркировке, и те переходы, которые при этом активизированы. Если активизированные переходы входят в параллельные участки процесса, т. е. могут быть реализованы одновременно, в вершине графа G_N они связываются знаком конъюнкции. Активизированные переходы, входящие в альтернативные ветви, соединяются знаком дизъюнкции. Если позиция a_μ в маркировке $\mathbf{M}_i$ имеет более одной метки, число этих меток указывается в виде верхнего индекса этой позиции.

Поскольку каждая маркировка сети Петри определяет состояние рассматриваемой системы, можно считать, что вершина $\mathbf{M}_i$ графа достижимых маркировок сети Петри соответствует состоянию S_i процесса, задаваемого этой сетью. Для каждого процесса задается начальное и конечное состояния; если процесс является циклическим, то выделяется состояние, после которого процесс повторяется. В общем случае может быть несколько начальных и конечных состояний. Начальное состояние соответствует начальной вершине графа G_N, конечное — его терминальной вершине. Состояние процесса, не являющееся конечным, но не допускающее дальнейшего его развития, является *тупиковым.* Вершина графа $G_{N\nu}$ соответствующая тупиковому состоянию, характеризуется тем, что при наличии меток в основных внутренних позициях сети Петри нет ни одного активизированного перехода.

3.2.3.4.2. Влияние структуры процесса на наличие тупиковых состояний

Рассмотрим различные случай возникновения тупиковых состояний. На рис. 3.22 приведен граф достижимых маркировок простого параллельно-последовательного консультационного процесса, заданного сетью Петри рис. 3.19.

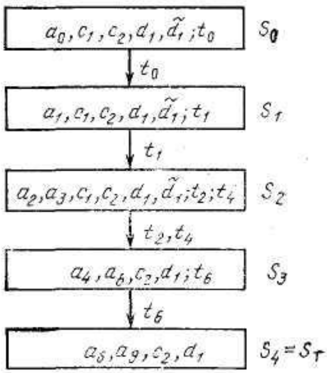

Рис. 3.22.

Для простоты на графе переходы будем выписывать без верхних индексов. Вначале построим граф G_N в предположении, что время реализации всех переходов одинаково, поэтому одновременно активизированные переходы реализуются также одновременно. При этом процесс переходит из состояния S_2 сразу в состояние S_3. Как видно из рис. 3.22, при заданной начальной маркировке, когда ФР c_1 и c_1 свободны и логическое условие $p_1 = 1$, процесс, не достигнув конечного состояния, соответствующего реализации перехода $t_{д}^{к}$, попадает в состояние S_4, которое является тупиковым, так как ни один из переходов не является активным и дальнейшее развитие процесса невозможно. Таким образом, при начальном истинном значении логического условия p_1 процесс попадает в тупиковое состояние.

Аналогичный граф достижимости можно построить для начального ложного значения логического условия p_1 когда позиция $\bar{d}_1$ в начальной маркировке не содержит метку. При этом вместо перехода t_2 будет активизирован переход t_3. Чтобы не составлять отдельные графы для различных допустимых начальных маркировок, можно построить один полный граф, предусматривающий все возможные пути развития процесса. Очевидно, при этом позиции ЛР в вершинах графа не выписываются. Такой граф для сети рис. 3.19 представлен на рис. 3.23; как видно, левая ветвь графа соответствует $p_1=1$ и полностью совпадает с графом рис. 3.22.

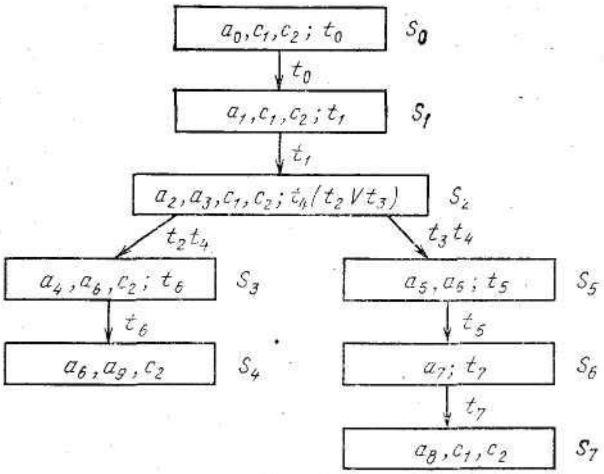

Рис. 3.23.

При p_1 = 0 реализуется другая последовательность переходов, но процесс, не достигнув конечного состояния, вновь попадает в тупиковое состояние S_7. Этот граф также построен в предположении, что реализация активизированных переходов завершается одновременно. Такой граф можно назвать *графом статических состояний* процесса. В статическом состоянии ни один из переходов сети Петри не находится в стадии реализации.

Очевидно, можно построить граф, содержащий не только статические, но и все промежуточные состояния, которые возникают из-за различной скорости реализации переходов. На рис. 3.24 приведен граф, задающий полное множество достижимых состояний рассматриваемого процесса, число которых, как видно, значительно больше числа статических состояний.

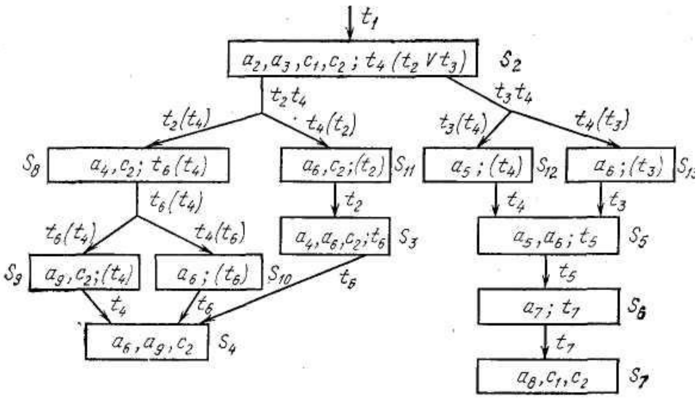

Рис. 3.24.

В таком динамическом графе исходящие из вершин дуги графа помечаются переходами, которые переходят в стадию реализации, входящие дуги помечаются переходом, закончившим реализацию. Переходы, реализация которых продолжается, записываются на этой дуге в скобках. В результате кроме статических состояний в графе появляются неустойчивые состояния; переходы, находящиеся при этом в стадии реализации, также выписываются в скобках в отличие от переходов, которые активизируются в данном неустойчивом состоянии. Следует иметь в виду, что с началом реализации перехода удаляются метки из его входных позиций, поэтому, например, в состоянии S_8 отсутствует метка в позиции a_3 и еще не появилась метка в позиции a_6, так как реализация перехода t_4 не закончена.

Из рис. 3.24 видно, что независимо от начального значения логического условия p_1 и относительной скорости реализации переходов, т. е. длительности соответствующих процедур, процесс, заданный сетью Петри рис. 3.19, попадает в одно из двух тупиковых состояний. Причиной этого является недопустимая структура процесса. Хотя в общем случае трудно сформулировать требования к правильной структуре процесса, некоторые обязательные условия можно указать. Необходимо, чтобы альтернативные ветви процесса имели альтернативное соединение в пределах того параллельного участка, в котором они начались. Если распараллеливание процесса происходит в одной из альтернативных ветвей, то соединение этих параллельных подпроцессов должно быть также в пределах этой ветви.

Нетрудно видеть, что сеть Петри рис. 3.19 не удовлетворяет этим требованиям. Возникновение таких неприятных явлений, как ловушка, зависание, обычно также связано с нарушением указанных условий. Тупики, вызванные неправильной структурой процесса, могут быть устранены только соответствующим ее изменением.

Другой причиной недостижимости конечного состояния может быть возникновение цикла в графе достижимых маркировок. Такая ситуация свойственна не только параллельному, но и последовательному разветвленному процессу при недопустимом взаимодействии процедур процесса с ЛР.

На рис. 3.25, *а* приведен граф достижимых маркировок сети Петри рис. 3.18 для фиксированной начальной маркировки позиций $\bar{d}_1$ и $\bar{d}_2$.

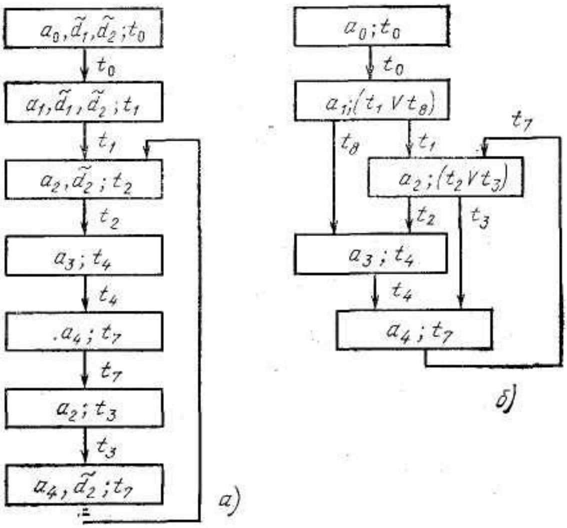

Рис. 3.25.

Как видно, при выполнении процесса возникает бесконечный цикл, так что заключительное состояние, связанное с реализацией перехода $t_6^{к}$, оказывается недостижимым. Полный граф достижимости (рис. 3.25, *б*) показывает, что это состояние не достигается ни при одной из возможных начальных маркировок позиций ЛР. Таким образом, возникновение бесконечного цикла является свойством данного процесса и объясняется недопустимым взаимодействием процедур процесса с логическим ресурсом D_1. В этом случае также требуется преобразование процесса.

Следует иметь в виду, что для более полного и точного анализа КП необходимо использовать всю имеющуюся в распоряжении консультанта информацию, так как в противном случае могут быть сделаны неверные выводы о ходе развития процесса. На рис. 3.26, *а* приведен граф G_N сети Петри рис. 3.20, *а* для истинного начального значения внешнего логического условия p_2. Этот граф содержит цикл, свидетельствующий о том, что если значение p_2 не изменится, процесс не достигнет конечного состояния. Поскольку здесь D_2 является внешним ЛР, неизвестно, произойдет ли такое изменение и будет ли реализован переход $t_6^к$.

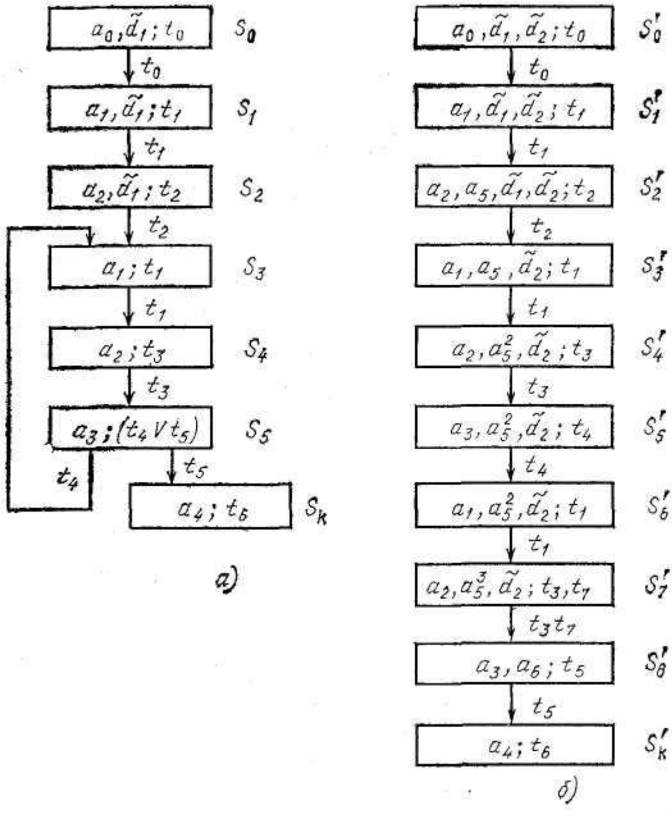

Рис. 3.26.

В сети Петри рис. рис. 3.20, *б* содержится информация о функционировании логического ресурса D_2 и его взаимодействии с КП. Граф достижимости этой сети (рис. 3.26, *б*) показывает, что никаких циклов при выполнении процесса не возникает и конечное состояние достижимо.

3.2.3.4.3. Тупиковые состояния, вызываемые разделением функциональных ресурсов

Значительная часть тупиковых состояний возникает из-за использования одних и тех же ФР процедурами параллельных подпроцессов. Рассмотрим более подробно такую ситуацию и способы ее устранения на примере сети Петри рис. 3.16, которая задает два циклических асинхронных процесса $П_1$ и $П_2$, разделяющих ФР C_1 и C_2. В начальной маркировке сети Петри содержатся метки в позициях c_1 и c_2, так как оба ресурса свободны, и в позициях a_1, a_7, поскольку оба процесса находятся в состоянии готовности. Для указания на асинхронность процессов введены внешние входные позиции b_1 и b_2 для переходов t_1 и t_7 соответственно, так что при появлении метки в позиции b_1 (b_2) начинается выполнение процесса $П_1$ ($П_2$). Если метки появляются одновременно в позициях b_1 и b_2, то оба процесса начинают выполняться параллельно.

Построим граф достижимых маркировок сети Петри и для наглядности изобразим его в виде решетки, горизонтальные пути которой соответствуют функционированию только подсети N_1, т. е. развитию процесса $П_1$, а вертикальные — подсети N_2 (рис. 3.27).

Вершины графа, определяющие состояние сети N, обозначим двойным индексом, первый из которых соответствует состояниям подсети N_1 а второй — подсети N_2.

Начальное состояние S_{00} является состоянием готовности процессов $П_1$ и $П_2$, каждый из которых начинает выполняться при его активизации, что отмечается появлением метки во внешней входной позиции сети Петри. Следует отметить, что маркировка внешних позиций не зависит от самого процесса. Эти позиции могут заполняться метками и освобождаться от них в любой момент, делая процесс активным либо пассивным. Важно только, чтобы метка во внешней входной позиции b_f перехода t_i сохранялась в течение времени, необходимого для реализации этого перехода. В отличие от внутренних входных позиций перехода t_i, метки из которых удаляются после его реализации, метка из позиции b_f не удаляется.

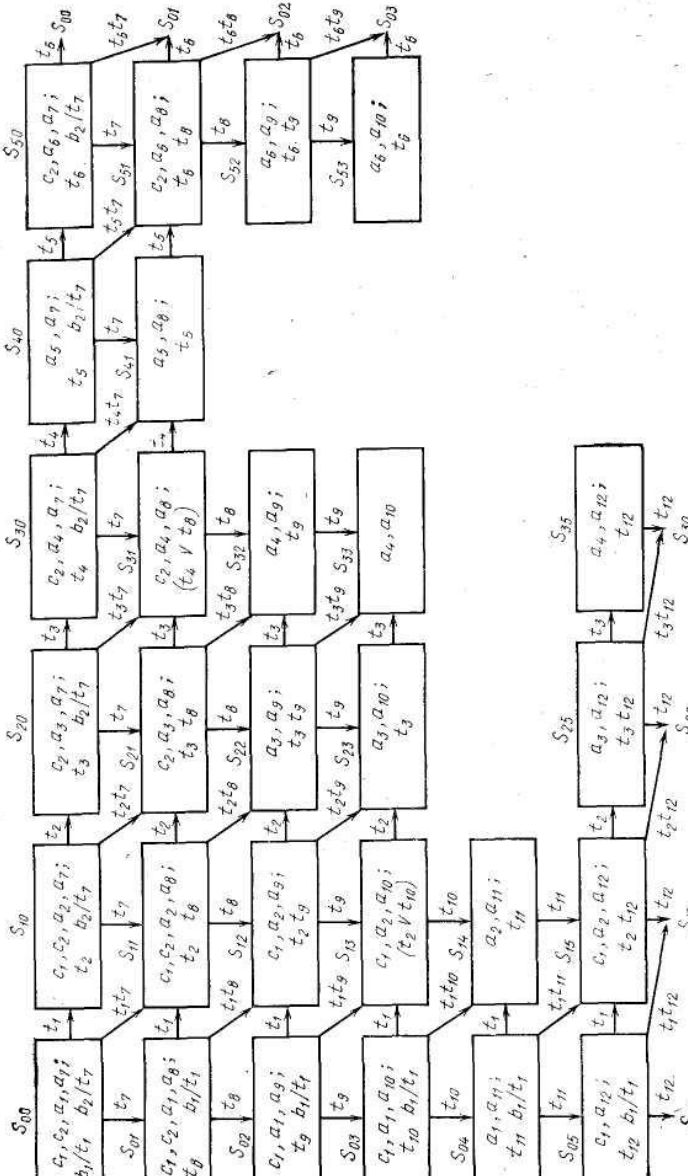

Рис. 3.27.

Активизация и выполнение процесса $П_1$ ($П_2$) вызывают последовательность состояний S_{10}, S_{20}, …, S_{50} (S_{01}, S_{02}, ..., S_{05}). Одновременное выполнение двух процессов соответствует остальным вершинам графа достижимых маркировок. Как видно из рис. 3.27, сеть Петри рис. 3.16 имеет 28 достижимых состояний. Нетрудно заметить, что не все эти состояния являются равноценными. В некоторых состояниях активизировано и может быть реализовано два перехода, что означает параллельное выполнение двух процессов. Другие состояния, например S_{13}, допускают развитие обоих процессов, но одновременно — только одного из них. В других состояниях активизирован только один переход, что говорит о возможности выполнения лишь одного процесса и блокировке другого (S_{41}, S_{14}). Имеется состояние (S_{33}), в котором, несмотря на наличие меток во внутренних позициях сети Петри, ни один из переходов не активизирован, т. е. ни один из процессов не может выполняться. Наличие такого множества различных состояний в графе достижимых маркировок сети Петри приводит к необходимости осуществить их классификацию, введя некоторые определения.

Состоянием блокировки $S_б$ назовем состояние, в котором при наличии метки во внутренней позиции $a_μ$, являющейся входной для перехода t_i, этот переход не активизирован, т. е. число процессов, которые могут развиваться, меньше числа активных процессов. Такая блокировка процесса происходит из-за занятости необходимых ему ресурсов другими процессами, после выполнения которых ресурсы освобождаются и заблокированный процесс может выполняться.

Состоянием взаимной блокировки $S_{в.б}$ назовем состояние, в котором заблокировано несколько активных процессов и они не могут быть разблокированы, несмотря на развитие других процессов. Таким образом, если в состоянии $S_б$ развивающиеся процессы освобождают ресурсы, необходимые для выполнения заблокированного процесса, то в состоянии $S_{в.б}$ все ресурсы, запрашиваемые заблокированным процессом, заняты процессами этого же множества, так что развитие других процессов не выводит систему из состояния взаимной блокировки. Если в множество заблокированных процессов входят все рассматриваемые процессы, система попадает в состояние полной взаимной блокировки $S_{п.в.б}$, когда ни один из процессов не может развиваться. Очевидно, состояния $S_{п.в.б}$ и $S_{в.б}$ одинаково недопустимы; будем называть их тупиковыми ($S_т$).

Состояние сети Петри, из которого все пути графа достижимых маркировок ведут в состояние $S_т$, назовем предтупиковым $S_{п.т}$.

Состояние $S_{т}$ и все предшествующие ему состояния $S_{п.т}$ объединим в одно множество $Q_{з}$, которое назовем множеством запрещенных состояний. Если в графе достижимых маркировок сети Петри имеется ребро, соединяющее вершины S_u и S_v такие, что $S_v \in Q_{з}$, а $S_u \in Q_{з}$, то состояние S_u назовем опасным ($S_{оп}$). Все опасные состояния образуют множество опасных состояний $Q_{оп}$. Остальные состояния являются безопасными.

Состоянием конфликта ($S_{кн}$) назовем состояние, в котором запрещена одновременная реализация хотя бы одной пары активизированных переходов.

Как видно из рис. 3.27 состояние S_{33} является тупиковым, $Q_{з}=\{S_{22}, S_{32}, S_{23}, S_{33}\}$; $Q_{оп}=\{S_{11}, S_{21}, S_{31}, S_{12}, S_{13}\}$.

Состояния S_{13} и S_{31} являются состояниями конфликта, а S_{41}, S_{53}, S_{14}, S_{35} — состояниями блокировки.

Поскольку в графе достижимых маркировок допустимы все пути, то ясно, что при наличии опасных состояний сеть Петри может попасть в тупиковое состояние. Для устранения такой возможности необходимо осуществить преобразование графа, исключив из него все ребра, соединяющие опасные и запрещенные состояния. При этом все опасные состояния станут безопасными и запрещенные состояния будут недостижимы. Такое преобразование графа достижимых маркировок в граф, содержащий только безопасные состояния, должно найти отражение в сети Петри.

Часть пути графа достижимых маркировок, содержащую только опасные состояния, назовем *опасным отрезком пути*. Опасные отрезки, содержащие опасные состояния одного и того же множества $Q_{оп}$, обязательно имеют одну общую величину, которая также соответствует опасному состоянию и может быть названа *корнем опасных отрезков* $S_{к.оп}$ (в крайнем случае это начальная вершина графа). Для того чтобы это состояние стало безопасным, необходимо превратить его в состояние конфликта, запретив одновременную реализацию активизированных переходов, которая переводит сеть в запрещенное состояние. С этой целью введем в сеть Петри дополнительную блокирующую позицию $a_б$, являющуюся общей входной позицией этих переходов. Позиция $a_б$ в начальной маркировке может содержать несколько меток, число которых определяется допустимым числом параллельно выполняемых опасных отрезков. Таким образом, разрешение конфликта в состоянии $S_{к.оп}$ соответствует выбору процессов, выполнение которых будет продолжаться, тогда как остальные процессы блокируются из-за отсутствия меток в позиции $a_б$. Эти процессы необходимо разблокировать сразу же, как только устранится опасность возникновения тупика. Для этого позиция $a_{б}$ должна быть выходной позицией всех переходов, которыми отмечены ребра графа, соответствующие выходу из опасных отрезков. Тогда после реализации таких переходов вновь появляются метки в позиции $a_{б}$, что делает возможным выполнение ранее заблокированных процессов.

Таким образом, предлагаемый способ устранения тупиковых состояний состоит во введении своевременной принудительной блокировки процессов. При этом определяется минимальное число процессов, которые необходимо заблокировать, и минимальное число состояний, в которых эта блокировка должна сохраняться.

Для графа достижимых маркировок рассматриваемого примера состояние S_{11} является корнем двух опасных отрезков S_{21}, S_{31} и S_{12}, S_{13}. Это состояние переводится в состояние конфликта введением в сеть Петри рис. 3.16 позиции $a_{б}$, являющейся входной для переходов t_2 и t_8. Тогда после реализации перехода t_2 (t_8), что соответствует развитию процесса $П_1$ $(П_2)$, удаляется метка из позиции $a_{б}$ и процесс $П_2$ $(П_1)$ блокируется. Как видно из графа рис. 3.27 состояние S_{31} (S_{13}) является последним состоянием выбранного опасного отрезка, поэтому при реализации перехода t_4 (t_{10}) блокировка может быть снята. Для этого позиция $a_{б}$ должна быть выходной позицией переходов t_4 и t_{10} (рис. 3.28).

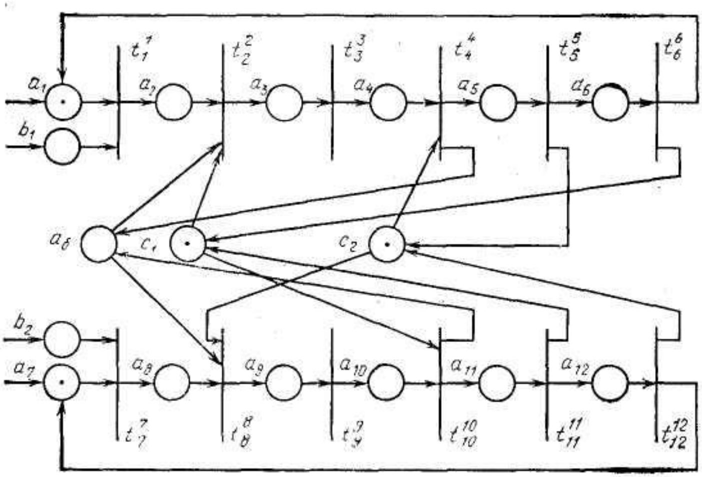

Рис. 3.28.

Теперь состояния S_{12}, S_{13}, S_{21}, S_{31}, которые в исходном графе были опасными, становятся безопасными состояниями блокировки, а состояние S_{11}—состоянием конфликта. Ясно, что при этом запрещенные состояния S_{22}, S_{32}, S_{23}, S_{33} станут недостижимы.

Для сложной системы взаимодействующих процессов, при которой имеется несколько процессов и они неоднократно обращаются к одним и тем же ресурсам, граф достижимых маркировок может содержать не одно, а несколько множеств запрещенных состояний. Тогда для каждого множества определяются множество опасных состояний, множество опасных отрезков путей графа и корень этих опасных отрезков. Затем в сеть Петри вводится блокирующая позиция так, чтобы опасное состояние, соответствующее корню, перевести в безопасное состояние конфликта. В каждом из конфликтующих опасных отрезков определяется последнее опасное состояние и переходы, которые отмечают ребра, переводящие эти состояния в безопасные, становятся входными переходами позиции a_6. В сети Петри, которая содержит необходимое число таких блокирующих позиций, не будут возникать тупиковые состояния, связанные с использованием общих ФР процедурами параллельных подпроцессов.

Таким образом, применение графа достижимых маркировок сети Петри позволяет обнаружить тупиковые состояния, выявить причины их возникновения и принять меры к их устранению. Для этого либо изменяется структура процесса и его взаимодействие с ЛР, либо вводится блокировка, запрещающая одновременное развитие некоторых участков параллельных процессов, которые называют иногда критическими. По графу достижимых маркировок можно также обнаружить непроизводительные циклы и недостижимые состояния процесса. Переходы сети Петри, соответствующие таким состояниям, должны быть исключены, если они являются избыточными, либо сделаны достижимыми путем изменения структуры процесса.

Очевидно, все преобразования структуры КП могут производиться по согласованию с консультантом консультируемой проблемы, так как иногда тупиковые состояния, например, вводятся специально для сигнализации о неисправностях в системе и т. д. Кроме того, после всех преобразований КП должен соответствовать управляемому технологическому процессу.

В результате должен быть получен *правильный КП, т. е. процесс, который не содержит непроизводительных бесконечных циклов, тупиковых состояний и все существенные состояния которого достижимы.*

3.3. Консультационные алгоритмы и языки их описания

3.3.1. Последовательный консультационный алгоритм и его свойства

Алгоритм выполнения консультационного процесса — это точное предписание о порядке выполнения этим консультационным процессом операций в ходе функционирования САК. Хотя понятие алгоритма широко используется в различных областях науки и техники, строгого его определения до сих пор не существует. В ряде работ сформулированы следующие основные свойства, которыми должен обладать любой алгоритм: *детерминированность алгоритма, т. е. наличие точного предписания, не оставляющего места произволу; массовость алгоритма, т. е. приложимость его к изменяющимся в известных пределах исходным данным; результативность алгоритма, т. е. получение результата за конечное число шагов выполнения алгоритма при надлежащих исходных данных*.

Кроме такого содержательного описания понятия алгоритма существуют различные его уточнения, основанные на машине Тьюринга, стекмашине, пушдаун-машине, баллон-машине и др., которые являются абстрактными моделями современных ЭВМ.
Мы не будем останавливаться на этих и других уточнениях понятия алгоритма, которые в основном используются в математике для доказательства несуществования алгоритмов, а рассмотрим некоторые формализованные языки их описания. Очевидно, к таким языкам относится уже рассмотренный язык сетей Петри, поскольку он по своим описательным возможностям приближается к машинам Тьюринга и, следовательно, может применяться для представления любых алгоритмов. Однако проверку некоторых свойств консультационных алгоритмов и их преобразование с целью оптимизации структуры консультационного процесса целесообразно осуществлять с использованием других языков, к числу которых относится язык логических схем алгоритмов и некоторые его модификации.

3.3.1.1. Логические схемы алгоритмов

Логические схемы алгоритмов (ЛСА) были предложены в 1952—1953 гг. А. А. Ляпуновым для целей программирования. Однако они могут использоваться в качестве языка задания алгоритмов выполнения консультационных процессов.

Задача (в том числе и консультационная задача), которую нужно решить, состоит в поиске алгоритма переработки некоторой исходной информации, т. е. в выборе отдельных операций, или *актов* алгоритма, и поиске порядка их выполнения.

С каждым таким актом (операцией) в ЛСА сопоставляется *оператор,* обозначаемый большими латинскими буквами A, B, C ... Различные операторы могут обозначаться разными буквами или одной и той же буквой, но с различными индексами: A_1, A_2, ..., B_1, B_2 ... Если оператор зависит от параметров, то эти параметры могут ставиться в качестве индексов A_i, A_{ij}, A_{ijk} ... или в скобках: $A(_i)$, $A(_{ij})$, $A(_{ijk})$ ... Операторы с различными параметрами выполняют действия над разными частями исходных или промежуточных данных, т. е. над разными частями перерабатываемой информации.

Например, при умножении квадратных матриц необходимо отдельно умножать каждую строку одной матрицы на каждый столбец другой, при этом умножение различных строк и столбцов производится аналогично. Поэтому для умножения различных строк и столбцов достаточно иметь один оператор, параметры которого укажут, какую строку и какой столбец необходимо в данный момент перемножать, т. е. оператор $A(_{ij})$ обозначает, что необходимо i-ю строку умножить на j-й столбец.

Последовательность выполнения операторов в ЛСА определяется порядком их записи. Например, ABC означает, что вначале выполняется оператор A, затем B, а затем C. Порядок выполнения в ЛСА операторов может быть строго фиксированным— *линейный алгоритм* — или зависящим от некоторых условий — *разветвленный алгоритм.* В последнем случае в ЛСА применяют *логические условия* (ЛУ), обозначаемые малыми латинскими буквами p, q, r ... Как и операторы, различные ЛУ обозначаются различными буквами или одной и той же буквой, но с разными индексами.

Логические условия могут зависеть от нескольких переменных. Логические условия, зависящие от значений функции n переменных, будем обозначать через

$$p[f(x_1, x_2, \dots, x_n)].$$

Считается, что логические условия могут принимать только два значения: выполняется проверяемое условие (p_i=1) или нет (p_i=0). В зависимости от значения в данный момент проверяемого ЛУ определяется дальнейший порядок выполнения операторов и ЛУ.

Часто среди логических условий целесообразно выделить такие, которые всегда принимают нулевое (ложное) значение, т. е. *тождественно-ложные логические условия.* Тождественно-ложные логические условия не требуют проверки. Будем обозначать их через ω. Операторы и ЛУ являются основными, а тождественно-ложные логические условия — вспомогательными членами логической схемы алгоритма.

Каждое ЛУ имеет стрелку. Начало *i*-й стрелки (обозначается через; $\uparrow^i$) стоит справа от логического условия, а ее конец (обозначается $\downarrow^i$)— слева от того члена ЛСА, который должен выполняться, если ЛУ принимает нулевое значение.

Логическими схемами алгоритма называют выражения, составленные из следующих друг за другом операторов и ЛУ, а также расставленных определенным образом нумерованных стрелок. Логическая схема алгоритма есть некоторый способ описания алгоритма решения поставленной задачи. Такое описание не зависит от того, какими средствами задача решается.

Описание алгоритма при помощи логических схем есть первый этап формализации алгоритма. Этому этапу предшествует содержательное описание алгоритма.

Логическая схема алгоритма допускает как формальные, так и содержательные равносильные преобразования.

Рассмотрим порядок выполнения алгоритма по логической схеме этого алгоритма. Пусть, например, из операторов *А, В, С* и логических условий p_1 и p_2 составлена следующая логическая схема алгоритма:

$$\downarrow^2 A p_1 \uparrow^1 B \downarrow^1 p_2 \uparrow^2 C. \tag{3.5}$$

Логическая схема определяет порядок выполнения операторов в зависимости от значения входящих в нее ЛУ.

Работа алгоритма начинается с того, что выполняется самый левый член схемы. После того как некоторый член схемы выполнится, определяется, какой член схемы должен выполняться следом за ним. Если это был оператор, то следом за ним должен выполняться тот член схемы, который стоит непосредственно справа от него. Если последний выполнившийся член схемы был логическим условием, то возможны два случая: если проверявшееся условие выполнено, то должен выполняться член, находившийся справа; если оно нарушено, должен выполняться тот член, к которому ведет стрелка, начинающаяся после данного условия.

Работа алгоритма оканчивается либо тогда, когда последний из выполняющихся операторов содержит указание о прекращении работы алгоритма, либо тогда, когда на некотором этапе не оказывается такого члена схемы, который должен был бы выполняться.

Таким образом, распределение значений ЛУ в логической схеме определяет порядок выполнения операторов, входящих в эту схему.

Например, в логической схеме (3.5) порядок выполнения операторов в зависимости от значений ЛУ будет следующим:

1) если p_1=0 и p_2=0, то *AA ... A* ..., т. е. бесконечно будет выполняться оператор *A;*

2) если p_1= 0, а p_2=1, то *AC,* т. е. вначале выполняется оператор *A,* а затем *C* и алгоритм заканчивает свою работу;

3) если p_1= 1, а p_2=0, то будет бесконечная последовательность чередования операторов *A* и *B: ABAB ... AB ...;*

4) если p_1=1 и p_2=1, то последовательно выполняются все три оператора и алгоритм заканчивает свою работу, т. е. *ABC.*

Следует заметить, что в процессе работы алгоритма логические условия могут изменяться или из-за того, что изменяются внешние к данной консультируемой проблеме условия или значения этих логических условий изменяются после выполнения того или иного оператора.

Например, в логической схеме алгоритма

$$A\downarrow^{1}Bp\uparrow^{2}C\omega\uparrow^{1}\downarrow^{2}D \tag{3.6}$$

будут поочередно выполняться операторы *B* и *C* до тех пор, пока не изменится значение логического условия *p* с 1 на 0:

$$\underbrace{ABCBC \ldots BC \ldots BC}_{p=1}\ \underbrace{BD}_{p=0}.$$

В качестве примера составления ЛСА рассмотрим алгоритм Евклида, решающий задачу нахождения наибольшего общего делителя для двух заданных натуральных чисел *a* и *b*. Решение для любых *a* и *b* можно получить путем построения убывающей последовательности чисел, из которых первое является большим из двух данных, второе — меньшим, третье получается как остаток от деления первого на второе, четвертое — как остаток от деления второго на третье и т. д., пока не будет деления без остатка. Делитель в последнем делении и будет искомым результатом.

Поскольку деление может быть сведено к повторному вычитанию, можно предложить следующий список предписаний:

1) сравнить числа *a* и *b* $(a=b,\ a<b,\ a>b)$ и перейти к следующему предписанию;

2) если числа равны, то каждое из них дает искомый результат. Процесс вычисления остановить. Если нет, то перейти к следующему предписанию;

3) если $a<b$, то поменять числа местами и перейти к следующему предписанию;

4) вычесть второе из первого и в качестве исходных взять два числа: вычитаемое и остаток. Перейти к предписанию 1.

Этот перечень предписаний есть словесная формулировка алгоритма, по которой можно составить логическую схему алгоритма, введя следующие операторы и ЛУ:

A — оператор, указывающий, какую пару чисел выбираем;

B — оператор вычитания второго числа y из первого x;

C — оператор перестановки чисел x и y;

D — оператор выбора вычитаемого и остатка в качестве первого и второго чисел;

O — оператор остановки и выдачи результата;

p — логическое условие, определяющее равенство двух чисел x и y; p=1, если $x=y$; p = 0, если $x \neq y$;

q— логическое условие, проверяющее равенство $x>y$;

$$q=1, \text{ если } x>y; \; q=0, \text{ если } x \not> y.$$

Для введенного таким образом множества операторов и ЛУ логическая схема алгоритма будет иметь вид

$$A\bar{p}\uparrow^{1}\bar{q}\uparrow^{2}\uparrow^{3}C\downarrow^{2}BD\bar{p}\uparrow^{1}\bar{q}\uparrow^{2}\omega\uparrow^{3}\downarrow^{1}0. \tag{3.7}$$

Здесь $\overline{p}$ — отрицание p, т. е. если p=1, то $\overline{p}$ =0 и если p=0, то $\overline{p}$ =1, аналогично и для q.

В ЛСА часто выделяют оператор A_0, символизирующий начало выполнения алгоритма, — *оператор начала* и оператор $A_к$, символизирующий окончание работы алгоритма, — *оператор конца.* Последний оператор часто заменяют точкой.

Кроме ЛСА на практике используются и другие способы описания структуры алгоритма. Наиболее распространенными из них являются *матричные схемы алгоритма* (МСА) и *формулы перехода,* обычно употребляемые в качестве промежуточных языков при преобразовании ЛСА.

3.3.1.2. Матричные схемы алгоритмов и их связь с логическими схемами. Понятие о граф-схемах

Матричная схема алгоритма — это квадратная матрица (табл. 3.6), каждая строка и каждый столбец которой сопоставлены с оператором.

Таблица 3.6

$$\mathfrak{A} = \begin{array}{c} \\ A_0 \\ A_1 \\ \vdots \\ A_{k-1} \end{array} \begin{array}{c} \begin{array}{ccc} A_1 & \cdots & A_к \end{array} \\ \begin{bmatrix} \alpha_{0,1} & \cdots & \alpha_{0к} \\ \alpha_{1,1} & \cdots & \alpha_{1к} \\ \vdots & & \vdots \\ \alpha_{k-1,1} & \cdots & \alpha_{k-1к} \end{bmatrix} \end{array}$$

Элементом a_{ij} матрицы является логическая функция логических условий. При этом оператор A_j, сопоставленный с j-м столбцом матрицы, выполняется после оператора A_i, сопоставленного с i-й строкой, если логическая функция

$$\alpha_{ij} = \alpha_{ij}\,(p_1 \ldots, p_m) = 1.$$

Логические функции МСА обладают следующими двумя свойствами, определяющими условия *непротиворечивости* и *полноты* алгоритма соответственно:

1) $\alpha_{ij}\alpha_{il} = 0,\ i \neq l;$
2) $\bigvee\limits_{j=1}^{k} \alpha_{ij} = 1,$

иными словами, произведение двух различных функций одной и той же строки МСА всегда равно 0. Это первое условие нужно для того, чтобы после оператора A_i могло выполняться не более одного оператора. Из второго условия следует, что после оператора A_i всегда должен выполняться хотя бы один оператор. Таким образом, после оператора A_i всегда выполняется только один оператор.

От ЛСА легко перейти к МСА. Рассмотрим такой переход на примере ЛСА вида (3.7). Вначале заполним строку МСА, сопоставленную с оператором A. Для этого найдем все функции α_{A_i}. Если $\bar{p}$ =1 и $\bar{q}$ =1, то после A необходимо выполнить оператор C. Следовательно, $\alpha_{AC} = \bar{p}\ \bar{q}$. Если $\bar{p}$ =1, а $\bar{q}$ =0, то после A необходимо выполнить оператор B. Таким образом, $\alpha_{AB} = \bar{p}\,q$. Аналогично при $\bar{p}$ = 0 получим $\alpha_{AO} = p$. Элемент α_{AD} равен нулю, так как ни при каких наборах значений ЛУ после оператора A не может непосредственно выполниться оператор D.

После оператора C всегда выполняется оператор B, т. е. α_{CB} =1. Аналогично, образовывая и другие логические функции, получим МСА, соответствующую ЛСА (3.7) (табл. 3.7).

Таблица 3.7

$$\begin{array}{c c}
 & \begin{array}{cccc} C & B & D & O \end{array} \\
\begin{array}{c} A \\ C \\ B \\ D \end{array} & \begin{bmatrix} \bar{p}\bar{q} & \bar{p}q & & p \\ & 1 & & \\ & & 1 & \\ \bar{p}\bar{q} & \bar{p}q & & p \end{bmatrix}.
\end{array}$$

Существует ряд способов перехода и от МСА к ЛСА. Рассмотрим наиболее простой, основанный на использовании формул перехода. Формула перехода записывается следующим образом.

Выбирается оператор (например, C) в МСА (табл. 3.7). Если после оператора C непосредственно следует только один оператор (в нашем случае оператор B), то этот факт записывается в виде $C\cdot\quad B$. Если после оператора (например, A в МСА табл. 3.7) может выполняться один из нескольких операторов в зависимости от значений логических условий, то в формуле перехода после полустрелки записывается первый в строке ненулевой элемент α_{ij}, который отмечается оператором A_j. Затем к такому члену с помощью знака дизъюнкции присоединяется следующий в этой строке ненулевой элемент α_{il}, отмеченный оператором A_l и т. д.

Так, в нашем случае для оператора A получим следующую формулу перехода:

$$A \rightarrow \bar{p}\,\bar{q}C \vee \bar{p}qB \vee pO.$$

Образовав такие формулы перехода для всех операторов, получим систему формул перехода. Например, по МСА (табл. 3.7) получим следующую систему формул перехода:

$$\begin{cases} A \rightarrow \bar{p}\,\bar{q}C \vee \bar{p}qB \vee pO. \\ C \rightarrow B; \\ B \rightarrow D; \\ D \rightarrow \bar{p}\,\bar{q}C \vee \bar{p}qB \vee pO. \end{cases}$$

От системы формул перехода можно перейти к ЛСА. Для этого необходимо прежде выписать оператор начала A_0. В нашем случае это оператор A. После него выписываются ЛУ (снабженные стрелками), соответствующие первому слева члену формулы перехода A, т. е. получим

$$A\bar{p}\uparrow\bar{q}\uparrow.$$

Затем выписывается оператор, отмечающий заданную функцию, т. е.

$$A\bar{p}\uparrow\bar{q}\uparrow C.$$

Когда выписан оператор, необходимо обратиться к формуле этого оператора и повторить эту процедуру:

$$A\bar{p}\uparrow\bar{q}\uparrow CB.$$

Затем получим последовательно:

$$A\bar{p}\uparrow\bar{q}\uparrow CBD;\quad Ap\uparrow\bar{q}\uparrow\ \downarrow^{1}CBD\bar{p}\uparrow\bar{q}\uparrow\omega\uparrow^{1}.$$

В последнем выражении введено тождественно-ложное ЛУ ω со стрелкой. Это сделано для того, чтобы не повторять в выражении дважды оператор *C*.

Когда последовательно уже нельзя выписать ни одного члена ЛСА (как это имеет место в нашем случае), необходимо обратиться к первому слева в последнем выражении ЛУ; пронумеровать стрелку и выписать члены, следующие за оператором *A* при $\bar{p}$ =0, и т. д.

В результате находим

$$A\bar{p}\uparrow^{2}q\uparrow\ \downarrow^{1}CBD\bar{p}\uparrow\bar{q}\uparrow\omega\uparrow^{1}\downarrow^{2}O;$$

$$A\bar{p}\uparrow^{2}\bar{q}\uparrow^{3}\downarrow^{1}C\downarrow^{3}BD\bar{p}\uparrow\bar{q}\uparrow\omega\uparrow^{1}\downarrow^{2}O;$$

$$A\bar{p}\uparrow^{2}\bar{q}\uparrow^{3}\downarrow^{1}C\downarrow^{3}BD\bar{p}\uparrow^{2}\bar{q}\uparrow\omega\uparrow^{1}\downarrow^{2}O;$$

$$A\bar{p}\uparrow^{2}\bar{q}\uparrow^{3}\downarrow^{1}C\downarrow^{3}BD\bar{p}\uparrow^{2}\bar{q}\uparrow^{3}\omega\uparrow^{1}\downarrow^{2}O. \tag{3.8}$$

В последнем выражении (3.8) выписаны все операторы МСА (табл. 3.7) и все стрелки пронумерованы. Это выражение и является ЛСА. При этом легко заметить, что ЛСА (3.8) с точностью до обозначения стрелок повторяет ЛСА (3.7).

Таким образом, формулы переходов использованы в качестве промежуточного языка при переходе от МСА к ЛСА. Правда, легко заметить, что переход от МСА к ЛСА можно осуществить и без формул перехода.

Однако, как далее будет видно, формулы перехода эффективно применяются при переходе от МСА к ЛСА, если при этом ставится задача получения минимальной ЛСА, т. е. ЛСА с минимальным числом членов.

В ряде случаев для задания алгоритма выполнения консультационного процесса могут быть использованы *граф-схемы алгоритмов*, которые дают более наглядное представление об алгоритме.

Граф-схема, соответствующая ЛСА (3.4), имеет вид схемы, изображенной на рис. 3.29.

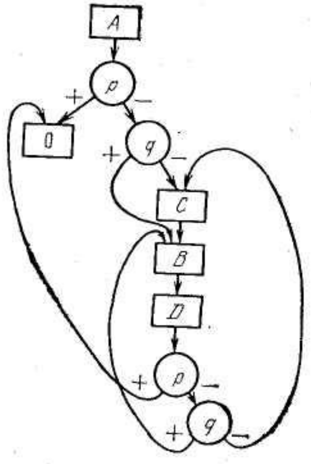

Рис. 3.29.

Из рисунка и ЛСА (3.8) легко видеть их взаимную связь, причем на граф-схеме операторы указаны прямоугольниками, а логические условия — кружочками. Начальный оператор (оператор *A)* не имеет входных стрелок, конечный (оператор *O*) — выходных. Единичное значение ЛУ на граф-схеме соответствует стрелке, отмеченной знаком +, а нулевое — стрелке со знаком —.

3.3.2. Преобразование логических схем алгоритмов

Процесс составления консультационного алгоритма является в настоящее время одним из самых неформализованных этапов построения консультационного процесса и формирования рекомендаций. Несмотря на попытки автоматизировать этот процесс на основе диалога ЭВМ с консультантом, получаемая при этом ЛСА, задающая алгоритм выполнения консультационного процесса, может содержать избыточное число членов. В то же время минимальность ЛСА является одним из критериев оптимальности структуры консультационного процесса и формируемых рекомендаций. В связи с этим возникает задача преобразования ЛСА с целью сокращения числа ее членов.

В ряде работ для упрощения ЛСА используются различные модели, позволяющие свести задачу минимизации ЛСА к другим задачам.

3.3.2.1. Минимизация числа логических условий

Рассмотрим метод, основанный на применении формул перехода и предполагающий отсутствие повторяющихся операторов в ЛСА. По имеющейся ЛСА или МСА составляется система формул перехода, затем в каждой формуле перехода находятся переменные, которые входят во все члены формулы. Заметим, что всегда можно найти хотя бы одну такую переменную. Если это переменная p_i , то формула перехода приведена по переменной p_i.

Например, возьмем следующую систему формул перехода из п. 3.3.1:

$$\begin{cases} A \to \bar{p}\,\bar{q}C \vee \bar{p}qB \vee pO; \\ C \to B; \\ B \to D; \\ D \to \bar{p}\,\bar{q}C \vee \bar{p}qB \vee pO, \end{cases}$$

полученную по МСА (табл. 3.7), равносильной ЛСА (3.7). Легко видеть, что в этой системе формула перехода для оператора A приведена по переменной p.

Среди переменных, по которым приведены формулы перехода, выбираются общие для всех или большинства формул. В оставшихся формулах выбираются другие общие переменные и т. д.

В нашем случае выберем переменную p, так как по ней приведены формулы переходов для операторов A и D; в остальных формулах переменных нет. Одну из выбранных переменных выносят за скобки в тех формулах перехода, которые приведены по этой переменной.

Для рассматриваемого примера получим

$$\begin{cases} A \to \bar{p}\,\bar{q}C \vee \bar{p}qB \vee pO; \\ C \to B; \\ B \to D; \\ D \to \bar{p}\,\bar{q}C \vee \bar{p}qB \vee pO, \end{cases}$$

Если в скобках получены выражения типа $(\bar{q}\,C \vee qB)$, т. е. такие, в которых уже нельзя вынести за скобки ни одной переменной, то такие скобки называют *элементарными.* Все формулы перехода преобразовывают так, чтобы процесс вынесения за скобки привел к элементарным скобкам.

В результате этого получим систему скобочных формул перехода. В ней выделяют общие выражения. Например, в полученной выше системе скобочных формул имеется общее выражение
$\bar{p}\ (\bar{q}\,C \vee qB) \vee pO$ в формулах перехода для операторов *A* и *D.*

Из всех вхождений в систему одного и того же выражения оставляют только одно (пусть в первой из таких формул перехода и первой слева в одной формуле перехода). Остальные заменяют тождественно-ложными логическими условиями $\omega\uparrow^{i}$, конец стрелки от которых помещают перед оставшимся выражением.

Таким образом, в рассматриваемом случае получим следующую преобразованную систему скобочных формул перехода:

$$\begin{cases} A \to \downarrow^{1} \bar{p}\,(\bar{q}C \vee qB) \vee pO; \\ C \to B; \\ B \to D; \\ D \to \omega \uparrow^{1}. \end{cases}$$

После того как из системы скобочных формул перехода исключены все повторяющиеся выражения, можно переходить к ЛСА, для чего необходимо выполнить следующие действия. Выписывается оператор начала, после которого—логическое условие, первое слева в формуле перехода этого оператора, затем второе условие и т. д. до тех пор, пока не будет выписан следующий оператор.

Например, из формулы перехода для оператора *A* выпишем следующее выражение:

$$A \downarrow^{1} \bar{p} \uparrow \bar{q} \uparrow C.$$

При этом, если перед выписываемым оператором или ЛУ стоит стрелка, она также выписывается с сохранением номера. Затем, как и ранее, обращаемся к формуле перехода оператора *C,* т. е.

$$A \downarrow^{1} \bar{p} \uparrow \bar{q} \uparrow CB.$$

Далее последовательно получим:

$$A \downarrow^{1} \bar{p} \uparrow \bar{q} \uparrow CBD; \quad A \downarrow^{1} \bar{p} \uparrow \bar{q} \uparrow CBD\omega \uparrow^{1}.$$

Если этот процесс обрывается, необходимо пронумеровать первую слева безындексную стрелку в последнем выражении, обратиться к соответствующей формуле перехода и аналогично рассмотренному выше продолжить построение ЛСА. Такиим образом, получим следующую последовательность выражений, иллюстрирующую процесс построения ЛСА:

$$A\downarrow^{1}\breve{p}\uparrow^{2}\bar{q}\uparrow CBD\omega\uparrow^{1}\downarrow^{2}O;$$

$$A\downarrow^{1}\bar{p}\uparrow^{2}\bar{q}\uparrow{}^{3}C\downarrow^{3}BD\omega\uparrow^{1}\downarrow^{2}O. \qquad (3.9)$$

Выражение (3.9) является ЛСА. При этом полученная таким образом ЛСА равносильна исходной ЛСА, по которой была построена МСА, т. е. при подстановке вместо ЛУ какого-либо набора их значений как в исходную, так и в получаемую этим способом ЛСА получаем по ним одну и ту же последовательность выполнения операторов. Следовательно, ЛСА (3.9) равносильна ЛСА (3.7), хотя в них содержится различное число членов, причем ЛСА (3.9) содержит меньшее число членов, чем ЛСА (3.7).

Таким образом можно осуществлять минимизацию ЛСА. Для этого от исходной ЛСА необходимо перейти к МСА, затем к формулам перехода, преобразовать их в скобочные, от которых перейти к минимальной ЛСА. Правда, необходимо отметить, что для построения минимальной ЛСА в ряде случаев необходимо осуществлять перебор вариантов вынесения переменных за скобки, объединение одинаковых выражений и т. п.

Заметим, что при минимизации ЛСА можно и не переходить к МСА, а систему формул перехода получать непосредственно по исходной ЛСА.

Рассмотрим пример упрощения ЛСА (3.10):

$$\mathfrak{A}=p_1\uparrow^{1}p_2\uparrow^{2}\downarrow^{5}A_1p_3\uparrow^{3}p_2\uparrow^{4}p_1\uparrow^{3}\omega\uparrow^{5}\downarrow^{1}p_2\uparrow^{5}{}_{к}$$

$$\omega\uparrow^{6}\downarrow^{2}A_2\downarrow^{6}A_3p_2\uparrow^{6}\omega\uparrow^{3}\downarrow^{4}p_1\uparrow^{6}\omega\uparrow^{2}\downarrow^{3}A_{к}. \qquad (3.10)$$

По ЛСА (3.10) составляем МСА (табл. 3.8).

Таблица 3.8

$$\mathfrak{A}=\begin{array}{c} \\ A_0 \\ A_1 \\ A_2 \\ A_3 \end{array}\begin{array}{c} \begin{array}{cccc} A_1 & A_2 & A_3 & A_к \end{array} \\ \left[\begin{array}{cccc} p_1p_2\vee\bar{p}_1\bar{p}_2 & p_1\bar{p}_2 & \bar{p}_1p_2 & \\ p_1p_2p_3 & p_1\bar{p}_2p_3 & \bar{p}_1\bar{p}_2p_3 & \bar{p}_1p_2p_3\vee\bar{p}_3 \\ & & 1 & \\ & & \bar{p}_2 & p_2 \end{array}\right] \end{array}$$

По МСА (табл. 3.8) получим систему уравнений перехода и преобразуем ее в систему скобочных формул перехода:

$$A_0 \to p_1 p_2 A_1 \vee \bar{p}_1 \bar{p}_2 A_1 \vee p_1 \bar{p}_2 A_2 \vee \bar{p}_1 p_2 A_3 = p_1 (p_2 A_1 \vee \bar{p}_2 A_2) \vee \bar{p}_1 (\bar{p}_2 A_1 \vee p_2 A_3);$$

$$A_1 \longrightarrow p_1 p_2 p_3 A_1 \vee p_1 \bar{p}_2 p_3 A_2 \vee \bar{p}_1 \bar{p}_2 p_3 A_3 \vee \bar{p}_1 p_2 p_3 A_\text{к} \vee \bar{p}_3 A_\text{к} =$$

$$= p_3 (p_1 p_2 A_1 \vee p_1 \bar{p}_2 A_2 \vee \bar{p}_1 \bar{p}_2 A_3 \vee \bar{p}_1 p_2 A_\text{к}) \vee \bar{p}_3 A_\text{к} =$$

$$= p_3 (p_1 (p_2 A_1 \vee \bar{p}_2 A_2) \vee \bar{p}_1 (\bar{p}_2 A_3 \vee p_2 A_\text{к}) \vee \bar{p}_3 A_\text{к});$$

$$A_2 \longrightarrow A_3;$$

$$A_3 \to \bar{p}_2 A_3 \vee p_2 A_\text{к}.$$

выявив в формулах перехода этой системы общие выражения, преобразуем ее в следующую систему:

$$A_0 \to p_1 \downarrow^\alpha (p_2 A_1 \vee \bar{p}_2 A_2) \vee \bar{p}_1 (\bar{p}_2 A_1 \vee p_2 A_3);$$

$$A_1 \longrightarrow p_3 (p_1 \omega \uparrow^\alpha \vee \bar{p}_1 \downarrow^\beta (\bar{p}_2 A_3 \vee p_2 A_\text{к})) \vee \bar{p}_3 A_\text{к};$$

$$A_2 \to A_3;$$

$$A_3 \to \omega \uparrow^\beta.$$

По последней системе составим следующую ЛСА:

$$\mathfrak{A} = p_1 \uparrow^3 \downarrow^5 p_2 \uparrow^1 \downarrow^4 A_1 p_3 \uparrow^7 p_1 \uparrow^6 \omega \uparrow^5 \downarrow^1 A_2 \downarrow^2 A_3 \downarrow^6 \bar{p}_2 \uparrow^7 \omega \uparrow^2 \downarrow^3 \bar{p}_2 \uparrow^2 \omega \uparrow^4 \downarrow^7 A_\text{к}, \quad (3.11)$$

Легко видеть, что ЛСА (3.11) содержит десять основных членов вместо двенадцати в равносильной ей ЛСА (3.10).

3.3.2.2. Учет распределения сдвигов при минимизации ЛСА

В процессе выполнения алгоритма консультационного процесса отдельные операторы могут изменять значения ЛУ. Например, имеется *ОФБ,* (сопоставленный с оператором A_i ЛСА), осуществляющий прибавление единицы к числу l. Логический $ФБ_j$ проверяет значение числа l и при $l \geq n$ вырабатывает значение логического условия p_j =l; при $l < n$ логическое условие p_j=0, т. е. $ЛФБ_j$ реализует логическое условие p_j $(l \geq n)$ ЛСА. Следовательно, оператор A_i может изменить значение логического условия p_j $(l \geq n)$.

Известно, что логическое условие p_j входит в *распределение сдвигов оператора* A_i т. е. A_i —{ p_j }, если оператор A_i может изменить значение p_j.

Таким образом, с каждым оператором A_i ЛСА может быть сопоставлено некоторое множество логических условий { p_{i1}, ..., $p_{i\zeta}$} из множества {p_1, ..., p_n}, т. е. A_i — p_{i1}, ..., $p_{i\zeta}$. Такое соответствие и называется распределением сдвигов.

Если с каждым оператором ЛСА сопоставлено полное множество ЛУ, т. е. $\zeta = n$, то *распределение сдвигов универсальное;* если с каждым оператором ЛСА сопоставлено пустое подмножество ЛУ, то *распределение сдвигов пустое*.

Заметим, что, если из условий работы САК нельзя или трудно выявить для оператора A_i логические условия, значения которых он может изменить, необходимо принимать универсальное распределение сдвигов.

Выше изложен способ минимизации ЛСА без учета влияния изменения значений ЛУ операторами ЛСА, т. е. этот способ применим в том случае, когда имеется универсальное распределение сдвигов. Учет конкретного распределения сдвигов позволяет в ряде случаев произвести дальнейшее упрощение ЛСА за счет сокращения числа вхождений ЛУ или числа проверок их значений в процессе выполнения ЛСА.

Рассмотрим несколько примеров, иллюстрирующих данный способ упрощения.

Пусть для ЛСА (3.12) задано следующее распределение сдвигов:

$$A_0\text{—}\{p_1,\ p_2,\ p_3\};\ A_1\text{—}\{p_2,\ p_3\};\ A_3\text{—}\{\text{—}\};$$

$$A_{\text{к}}\text{—}\{p_1,\ p_2,\ p_3\};$$

$$\mathfrak{A}=p_1\uparrow^1 A_1 p_1\uparrow^1 p_2\uparrow^1 A_2 p_1\uparrow^1 p_2\uparrow^1 p_3\uparrow^1 A_3\downarrow^1 A_{\text{к}}. \tag{3.12}$$

Заметим, что для оператора A_3 задано пустое распределение сдвигов (черточка в скобках), а для операторов A_0 и $A_{\text{к}}$— универсальное.

Для упрощения ЛСА с учетом распределения сдвигов удобно использовать МСА, поэтому от ЛСА (3.12) перейдем к МСА (табл. 3.9).

Таблица 3.9

$$\mathfrak{A}=\begin{array}{c} \\ A_0\\ A_1\\ A_2\\ A_3 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} A_1 & A_2 & A_3 & A_{\text{к}} \end{array}\\ \left[\begin{array}{cccc} p_1 & & & \bar p_1\\ & p_1 p_2 & & p_1\bar p_2\vee\bar p_1\\ & & p_1 p_2 p_3 & p_1 p_2\bar p_3\vee p_1\bar p_2\vee\bar p_1\\ & & & 1 \end{array}\right] \end{array}$$

Из МСА (табл. 3.9) видно, что оператор A_1 может выполниться лишь после оператора A_0 при p_1=l. Вместе с тем логическое условие p_1 не входит в распределение сдвигов оператора A_1. Значит, оператор A_1 не может изменить значение логического условия p_1 поэтому после выполнения A_1 логическое условие по-прежнему будет иметь единичное значение. В связи с этим после выполнения оператора A_1 не нужно проверять p_1 и можно в строке Л[МСА (табл. 3.9) переменную p_1 заменить ее значением, т. с. получить следующую МСА (табл. 3.10).

Таблица 3.10

$$
\mathfrak{A} = \begin{array}{c} \\ A_0 \\ A_1 \\ A_2 \\ A_3 \end{array}
\begin{array}{c}
\begin{array}{ccccc} A_1 & A_2 & A_3 & & A_к \end{array} \\
\left[\begin{array}{ccccc}
p_1 & & & & \bar{p}_1 \\
 & p_2 & & & \bar{p}_2 \\
 & & p_1 p_2 p_3 & & p_1 p_2 \bar{p}_3 \vee p_1 \bar{p}_2 \vee \bar{p}_1 \\
 & & & & 1
\end{array}\right]
\end{array}
$$

Из этой МСА видно, что оператор A_3 может быть выполнен только после A_2, если p_1 = l, p_2=1 и p_3=1. Однако ни p_1, ни p_2 в распределение сдвигов оператора A_2 не входят. Следовательно, значения логических условий p_1 и p_2 будут такими, какими они были до выполнения оператора A_2.   .

Из табл. 3.10 видно, что для выполнения оператора A_2 необходимо иметь значение p_2= 1. Поэтому, как и ранее, в строке A_2 МСА (табл. 3.10) переменную p_2 можно заменить единицей, а ее отрицание — нулем. Вместе с тем из табл. 3.10 не следует, что оператор A_2, выполнится лишь при каком-то одном значении p_1. В связи с этим необходимо рассмотреть более подробно условия выполнения оператора A_2 и его предшественников. Оператор A_2 выполняется только после оператора A_1, который, как и A_2, не изменяет значения p_1, поэтому рассмотрим условия выполнения оператора A_1. Из МСА видно, что оператор A_1 может выполняться при p_1=l. Так как ни A_1, ни A_2 не изменяют значение p_1, то после выполнения оператора A_2 логическое условие p_1 по-прежнему будет иметь единичное значение. Следовательно, в строке A_2 МСА переменную p_1 можно заменить единицей, а ее отрицание — нулем. В результате получим МСА (табл. 3.11).

Таблица 3.11

$$
\mathfrak{A} = \begin{array}{c} \\ A_0 \\ A_1 \\ A_2 \\ A_3 \end{array}
\begin{array}{c}
\begin{array}{cccc} A_1 & A_2 & A_3 & A_к \end{array} \\
\left[\begin{array}{cccc}
p_1 & & & \bar{p}_1 \\
 & p_2 & & \bar{p}_2 \\
 & & p_3 & \bar{p}_3 \\
 & & & 1
\end{array}\right]
\end{array}
$$

Заметим, что условие выполнения A_2 при p_1=l следует и из МСА (табл. 3.9), в которой еще не произведено упрощение с учетом распределения сдвигов, тогда как из табл. 3.10 этого не видно. Поэтому желательно не заменять переменные их значениями до окончания упрощения ЛСА.

По полученной МСА (табл. 3.11) легко получить систему формул перехода, а затем и упрощенную, ЛСА (3.13), имеющую только семь членов вместо десяти в ЛСА (3.12), составленной без учета распределения сдвигов:

$$A_0 \to p_1A_1 \vee \bar{p}_1A_к;$$
$$A_1 \to p_2A_2 \vee \bar{p}_2A_к;$$
$$A_2 \to p_3A_3 \vee \bar{p}_3A_к;$$
$$A_3 \to A_к$$
$$\mathfrak{A} = p^1\uparrow^1 A_1 p_2 \uparrow^1 A_2 p_3 \uparrow^1 A_3 \downarrow^1 A_к. \quad (3.13)$$

Процесс выявления ЛУ, которые могут быть заменены их значениями, можно формализовать. Для этого по первоначальной МСА (в рассматриваемом примере по табл. 3.9) образуют функцию f_{A_j} в виде дизъюнкции всех элементов столбца A_j:

$$f_{A_j} = \bigvee_{i=0}^{k-1} \alpha_{ij}. \quad (3.14)$$

В нашем случае

$$f_{A_1} = p_1; \quad f_{A_2} = p_1p_2; \quad f_{A_3} = p_1p_2p_3;$$

$$f_{A_1} = p_1; \quad f_{A_2} = p_1\bar{p}_2 \vee \bar{p}_1 \vee p_1p_2\bar{p}_3 \vee p_1\bar{p}_2 \vee \bar{p}_1 \vee 1 = 1$$

Среди всех этих функций выявляют так называемые *сокращенные особенные функции* вида

$$f(p_1, \ldots, p_n) = p_{i_1}^{\sigma_1}, \ldots, p_{i_m}^{\sigma_m} f',$$

где

$$m \leqslant n, \; p_{i_\zeta}^{\sigma_\zeta} = \begin{cases} p_{i_\zeta}, & \text{если } \sigma_\zeta = 1; \\ \bar{p}_{i_\zeta}, & \text{если } \sigma_\zeta = 0; \end{cases}$$

f' — функция, не зависящая от переменных $p_{i_1}, \ldots, p_{i_m}$

В нашем случае сокращенными особенными функциями будут f_{A_1}, f_{A_2} и f_{A_3}. Из этих функций получим $\varphi_{A_i} = p_{i_1}, \ldots, p_{i_m}$.

Из последнего выражения исключим логические условия, входящие в распределение сдвагов оператора A_i, и получим функции

$$\varphi^*_{A_i} \geqslant \varphi_{A_i}; \quad \varphi^*_{A_1} = p_1; \quad \varphi^*_{A_2} = p_1p_2; \quad \varphi^*_{A_3} = p_1p_2p_3.$$

Из этих функций видно, что в строке A_1 матрицы (3.9) переменную p_1 можно заменить на единицу, а переменную $\bar{p}_1$ — на нуль; в строке A_2

переменные p_1 и p_2 — на единицу, а переменные $\overline{p}_1$ и $\overline{p}_2$ — на нуль. В результате такой замены получим МСА (табл. 3.11).

Если в какой-либо строке МСА стоит единица, изложенный способ может не привести к сокращению числа ЛУ, тогда как такое сокращение возможно. Для этого все единицы в матрице необходимо заменить *максимумами логических функций*.

Пусть задана МСА (табл. 3.12) и следующее распределение сдвигов: A_0— $\{p_1, p_2\}$; A_1—$\{p_2\}$; A_2 — $\{p_2\}$; A_3—$\{p_2\}$; $A_к$—$\{p_1, p_2\}$.

Таблица 3.12

$$\mathfrak{A} = \begin{array}{c} \\ A_0 \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & A_3 & A_к \end{array} \\ \left[\begin{array}{cccc} p_1p_2 & & \overline{p}_1 & p_1\overline{p}_2 \\ & 1 & & \\ & & \overline{p}_1 & p_1 \\ & p_1 & & \overline{p}_1 \end{array}\right] \end{array}$$

Образуем вначале функции

$$f_{A_1} = p_1p_2;\ f_{A_2} = 1 \vee p_1 = 1;\ f_{A_3} = \overline{p}_1;\ f_{A_к} = p_1\overline{p}_2 \vee p_1 \vee \overline{p}_1 = 1,$$

а затем $\varphi'_{A_1} = p_1p_2$; $\varphi_{A_3} = \overline{p}_1$ и функции

$$\varphi^*_{A_1} = p_1p_2;\ \varphi^*_{A_3} = \overline{p}_1.$$

В строке A_3 переменную p_1 можно заменить на нуль, а переменную $\overline{p}_1$ — на единицу. В результате получим МСА (табл. 3.13).

Таблица 3.13

$$\mathfrak{A} = \begin{array}{c} \\ A_0 \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & A_3 & A_к \end{array} \\ \left[\begin{array}{cccc} p_1p_2 & & \overline{p}_1 & p_1\overline{p}_2 \\ & 1 & & \\ & & \overline{p}_1 & p_1 \\ & & & 1 \end{array}\right] \end{array}$$

Однако если умножить единичный элемент $\alpha_{1,2}$ из табл. 3.12 на $\varphi^*_{A_1}$, то найдем f_{A_2} =l $(p_1p_2 \vee p_1)=p_1$ Это означает, что перед выполнением оператора A_2 логическое условие p_1 всегда имеет единичное значение, хотя для выполнения оператора A_2 после оператора A_1 его проверять нет необходимости.

Образовав теперь $\varphi_{A_2}=p_1$, видим, что в строке A_2 (табл. 3.12) переменную p_1 можно заменить на единицу, а $\overline{p}_1$—на нуль. В результате получим следующую МСА (табл. 3.14):

Таблица 3.14

$$\mathfrak{A} = \begin{array}{c} \\ A_0 \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & A_3 & A_к \end{array} \\ \left[\begin{array}{cccc} p_1p_2 & & \bar{p}_1 & p_1\bar{p}_2 \\ & 1 & & \\ & & & 1 \\ & & & 1 \end{array}\right] \end{array}$$

Как видно, последняя МСА проще МСА табл. 3.13, полученной без умножения единичных элементов МСА на функции $\varphi^*_{A_i}$.

По МСА табл. 3.13 получим ЛСА

$$\mathfrak{A} = p_1 \uparrow^1 p_2 \uparrow^2 A_1 A_2 \bar{p}_1 \uparrow^2 \downarrow^1 A_3 \downarrow^2 A_к, \qquad (3.15)$$

а по МСА табл. 3.14 следующую ЛСА:

$$\mathfrak{A} = p_1 \uparrow^1 p_2 \uparrow^2 A_1 A_2 \downarrow^1 A_3 \downarrow^2 A_к. \qquad (3.16)$$

Как видно, ЛСА (3.16) проще ЛСА (3.15):

3.3.3. Объединение ЛСА

При задании условий работы САК в виде совокупности частных ЛСА, очевидно, в каждую из таких ЛСА могут входить одни и те же операторы и ЛУ. Поэтому когда вместо каждого символа частной ЛСА будут подставлены соответствующие выражения из операторов и ЛУ, ЛСА, описывающая общий алгоритм реализации консультационного процесса, может оказаться не минимальной. При этом в такой ЛСА повторяются не только ЛУ, но и операторы. Однако описанными выше способами такое повторение операторов не может быть исключено.

Для минимизации ЛСА с повторяющимися операторами и ЛУ может быть использован метод объединения ЛСА. Вначале этот метод рассмотрим для случая попарного объединения ЛСА. Произведем объединение двух частных ЛСА (3.17) и (3.18), описывающих два режима работы САК:

$$\mathfrak{A}_{\mathrm{I}} = \downarrow^2 F_i p_1 \uparrow^1 p_3 \uparrow^2 \downarrow^4 F_j p_4 \uparrow^3 p_5 \uparrow^4 p_2 \uparrow^4 A_1 A_2 \omega \uparrow^2 \downarrow^3 A_3 \omega \uparrow^2 \downarrow^1; \qquad (3.17)$$

$$\mathfrak{A}_{\mathrm{II}} = \downarrow^1 F_j p_4 \uparrow^2 p_6 \uparrow^1 \downarrow^4 F_i p_1 \uparrow^3 p_7 \uparrow^4 p_2 \uparrow^4 A_1 A_2 \omega \uparrow^1 \downarrow^3 A_3 \omega \uparrow^1 \downarrow^2. \qquad (3.18)$$

Общий алгоритм функционирования САК имеет вид

$$\mathfrak{A} = \downarrow^2 r \uparrow^1 \mathfrak{A}_{\mathrm{I}} \omega \uparrow^2 \downarrow^1 \mathfrak{A}_{\mathrm{II}} \omega \uparrow^2. \qquad (3.19)$$

В зависимости от значения параметра *r*, задаваемого извне, можно настраивать САК на выполнение A_1 или A_{11}. Если подставить в ЛСА (3.19) выражения для A_1, A_{11}, то получим ЛСА с 21 членом (как и ранее, здесь не учитываем число тождественно-ложных условий).

Процесс построения объединенной ЛСА, в которой операторы не повторяются, состоит в следующем. По МСА двух частных ЛСА составляется МСА объединенной ЛСА, каждый элемент которой

$$\alpha_{ij} = \bigvee_{l=1,\,2} \alpha^{l}_{ij}\beta^{l}_{i}, \tag{3.20}$$

где β^l_i — определяющая функция, равная функции $f\ (r_1\ \ldots\ \ldots,\ r_n)$, принимающей единичное значение на наборе значений переменных $r_1\ \ldots\ \ldots,\ r_n$, который сопоставлен с ЛСА A_l. Если оператор A_i не входит в ЛСА A_l, то в функцию β^l_i соответствующий набор значений $r_1\ \ldots\ \ldots,\ r_m$ войдет в качестве условного.

Если число наборов значений переменных $r_1\ \ldots\ \ldots,\ r_n$ больше числа объединяемых ЛСА, то в определяющую функцию могут входить конъюнкции, которые принимают единичные значения на неиспользуемых наборах.

В нашем случае для всех операторов ЛСА A_I $\beta_i^{\mathrm{I}}=r$, а для всех операторов ЛСА A_{II} $\beta_i^{\mathrm{II}}=\overline{r}$, так как в A_I и A_{II} входят все операторы.

Составим предварительно МСА, равносильные ЛСА (3.17) и (3.18), в виде табл. 3.15 и 3.16:

Таблица 3.15

$$\mathfrak{A} = \begin{array}{c} \\ A_0 \\ A_1 \\ A_2 \\ A_3 \\ F_i \\ F_j \end{array} \begin{array}{c} \begin{array}{cccccc} A_1 & A_2 & A_3 & F_i & F_j & A_\kappa \end{array} \\ \left[\begin{array}{cccccc} & & & 1 & & \\ & 1 & & & & \\ & & & 1 & & \\ & & & 1 & & \\ & & & p_1\overline{p}_3 & p_1p_3 & \overline{p}_1 \\ p_2p_4p_5 & & \overline{p}_4 & & \overline{p}_2p_4p_5 \vee p_4\overline{p}_5 & \end{array}\right] \end{array}$$

Таблица 3.16

$$\mathfrak{A}_{II} = \begin{array}{c} \\ A_0 \\ A_1 \\ A_2 \\ A_3 \\ F_i \\ F_j \end{array} \begin{array}{c} \begin{array}{cccccc} A_1 & A_2 & A_3 & F_i & F_j & A_\kappa \end{array} \\ \left[\begin{array}{cccccc} & & & & 1 & \\ & 1 & & & & \\ & & & & 1 & \\ & & & & 1 & \\ p_1p_2p_7 & & \overline{p}_1 & p_1\overline{p}_2p_7 \vee p_1\overline{p}_7 & & \\ & & & p_4p_6 & p_4\overline{p}_6 & \overline{p}_4 \end{array}\right] \end{array}$$

После этого по указанному выше правилу получим следующую МСА, равносильную объединенной ЛСА (табл. 3.17):

Таблица 3.17

$$\mathfrak{A} =$$

$$
= \begin{array}{c} \\ A_0 \\ A_1 \\ A_2 \\ A_3 \\ F_i \\ F_j \end{array}
\begin{array}{c}
\begin{array}{ccccc} A_1 & A_2 & A_3 & F_i & F_j \end{array} \\
\left[\begin{array}{ccccc}
 & & & r & \bar{r} \\
 & r \vee \bar{r} = 1 & & & \\
 & & & r & \bar{r} \\
 & & & r & \bar{r} \\
p_1p_2p_7\bar{r} \vee \bar{p}_1 r & & \bar{p}_1\bar{r} & p_1\bar{p}_3 r \vee p_1\bar{p}_2p_7\bar{r} \vee p_1\bar{p}_7\bar{r} & p_1p_3r \\
p_2p_4p_5r \vee \bar{p}_4\bar{r} & & \bar{p}_4 r & p_4p_6\bar{r} & \bar{p}_2p_4p_5r \vee p_4\bar{p}_5r \vee p_4\bar{p}_6\bar{r}
\end{array}\right]
\end{array}
$$

От МСА (табл. 3.17) перейдем к системе формул перехода, после чего преобразуем ее в систему скобочных формул перехода, где объединим общие выражения:

$$
\left\{\begin{array}{c}
A_0 \rightarrow \downarrow^{\alpha} rF_i \vee \bar{r}F_j; \\
A_1 \rightarrow A_2; \\
A_2 \rightarrow rF_i \vee \bar{r}F_j = \omega \uparrow^{\alpha}; \\
A_3 \rightarrow rF_i \vee \bar{r}F_j = \omega \uparrow^{\alpha}; \\
F_i \rightarrow p_1p_2p_7\bar{r}A_1 \vee \bar{p}_1 rA_1 \vee \bar{p}_1\bar{r}A_3 \vee p_1\bar{p}_3rF_i \vee \\
\vee\, p_1\bar{p}_2p_7\bar{r}F_i \vee p_1\bar{p}_7\bar{r}F_i \vee p_1p_3rF_j = p_1(r(p_3F_j \vee \bar{p}_3F_i) \vee \\
\vee\, \bar{r}(p_7(p_2A_1 \vee \bar{p}_2F_i) \vee \bar{p}_7F_i)) \vee \bar{p}_1(rA_1 \vee \bar{r}A_3) = \\
= p_1(r(p_3F_j \vee \bar{p}_3F_i) \vee \bar{r}(p_7 \downarrow^{\beta} (p_2A_1 \vee \bar{p}_2F_i) \vee p_7F_i)) \vee \\
\vee\, \bar{p}_1(rA_1 \vee \bar{r}A_3); \\
F_j \rightarrow p_2p_4p_5rA_1 \vee \bar{p}_4\bar{r}A_1 \vee \bar{p}_4rA_3 \vee p_4p_6\bar{r}F_i \vee \\
\vee\, \bar{p}_2p_4p_5rF_j \vee p_4\bar{p}_5rF_j \vee p_4\bar{p}_6\bar{r}F_j = \\
= r(p_4(p_5(p_2A_1 \vee \bar{p}_2F_j) \vee \bar{p}_5F_j) \vee \bar{p}_4A_3) \vee \bar{r}(p_4(p_6F_i \vee \\
\vee\, \bar{p}_6F_j) \vee \bar{p}_4A_1) = r(p_4(p_5\omega \uparrow^{\beta} \vee \bar{p}_5F_j) \vee \\
\vee\, \bar{p}_4A_3) \vee \bar{r}(p_4(p_6F_i \vee \bar{p}_6F_j) \vee \bar{p}A_1).
\end{array}\right.
$$

Теперь нетрудно получить и объединенную ЛСА:

$$\mathfrak{A} = \downarrow^1 r \uparrow^2 \downarrow^6 F_i p_1 \uparrow^3 r \uparrow^4 p_3 \uparrow^6 \downarrow^2 F_j r \uparrow^7 p_4 \uparrow^8 p_5 \uparrow^2 \omega \uparrow^9 \downarrow^3 r \uparrow^8 \downarrow^5$$
$$A_1 A_2 \omega \uparrow^1 \downarrow^4 p_7 \uparrow^6 \downarrow^9 p_2 \uparrow^6 \omega \uparrow^5 \downarrow^7 p_4 \uparrow^5 p_6 \uparrow^2 \omega \uparrow^6 \downarrow^8 A_3 \omega \uparrow^1, \quad (3.21)$$

в которой имеется 17 членов вместо 21 в ЛСА (3.19).

Если в ЛСА (3.21) подставить значение r=1, то можно легко убедиться, что получим ЛСА, равносильную ЛСА (3.17), а при подстановке r =0 получим ЛСА, равносильную ЛСА (3.18).

Очевидно, при объединении L ЛСА вместо одной переменной потребуется n переменных r_1 ..., r_n, $2^n \geq L$, так как каждой из L ЛСА сопоставляется своя, отличная от других конъюнкция

$$R_i = r_1^{\sigma_1^i}, \ldots, r_n^{\sigma_n^i},$$

называемая *определяющей конъюнкцией.*

Таким образом, объединенной ЛСА $\mathcal{A}$ называют ЛСА, удовлетворяющую следующим условиям:

1) если оператор A_i входит хотя бы в одну из частных ЛСА $\mathcal{A}_j$, то он обязательно входит в ЛСА $\mathcal{A}$, причем только один раз (если в каждой из частных ЛСА нет повторений одинаковых операторов);

2) при подстановке набора значений переменных r_1 ..., r_n, на котором R_j=1, ЛСА $\mathcal{A}$ превращается в ЛСА, равносильную частной ЛСА $\mathcal{A}_j$.

Предполагая наличие только неповторяющихся операторов в каждой из частных ЛСА, получим, что каждый оператор в объединенную ЛСА входит только один раз. Однако в различных частных ЛСА после одного и того же оператора могут выполняться различные члены ЛСА. Поэтому каждый раз при выполнении оператора объединенной ЛСА необходимо знать, в какой из частных ЛСА он в данный момент выполняется.

Проверка значений переменных r_1 ..., r_n, осуществляемая после выполнения оператора, входящего в различные частные ЛСА, позволяет выбрать тот член объединенной ЛСА, который должен выполняться в частной ЛСА, соответствующей данному набору значений переменных r_1 ..., r_n .

Нетрудно понять, что если оператор входит только в одну из L ЛСА, то

$$\beta_i^l = R_l \vee \frac{R_{j_1}}{0} \vee \ldots \vee \frac{R_{i_2 n}}{0} = 1.$$

При объединении L ЛСА определяющая функция может быть недоопределенной, поэтому из (3.20) следует, что элементы матрицы объединенной ЛСА могут быть недоопределены.

Получаемая при объединении L ЛСА матрица с такими недоопределенными элементами является недетерминированной МСА $\hat{A}$.

Легко понять, что недетерминированная МСА переводится в МСА при подстановке вместо переменных $r_1 ..., r_n$ в β_i^l набора значений $\sigma_1^l, ..., \sigma_n^l$, на котором

$$R_l = r_1^{\sigma_1^l}, ..., r_n^{\sigma_n^l} = 1.$$

Если с каждой из L объединяемых ЛСА сопоставлена лишь одна определяющая конъюнкция, а $L<2^n$, то 2^n—L конъюнкций переменных $r_1 ..., r_n$ останутся неиспользуемыми. Для того чтобы объединенная ЛСА обладала свойством детерминированности, эти конъюнкции должны быть переведены в используемые. Например, их можно сопоставить с одной из ЛСА или потребовать, чтобы на соответствующих наборах значений переменных $r_1 ..., r_n$ выполнилась бы «пустая» ЛСА вида $A = A_k$. Однако такое доопределение приводит в большинстве случаев к неоптимальной ЛСА. Оказывается более целесообразным составить неиспользуемые ЛСА, которые могут быть и неравносильны ни одной из используемых частных ЛСА. Построение таких неиспользуемых ЛСА целесообразно производить не сразу, а в процессе получения объединенной ЛСА, т. е. здесь задача в каком-то смысле аналогична задаче синтеза консультационных процессов или схемы из функциональных элементов по не полностью определенным булевым функциям.

Таким образом, как и при объединении двух ЛСА, в данном случае по частным ЛСА составляются МСА. Затем выявляются определяющие функции β_i^l, которые в данном случае могут быть и недоопределенными. После этого в соответствии с (3.20) построим недетерминированную МСА. По недетерминированной МСА получим недоопределенные формулы перехода.

В процессе вынесения за скобки переменных, как и в случае учета неиспользуемых наборов значений логических условий, недоопределенные формулы перехода переводим в определенные скобочные формулы перехода, причем доопределение формул перехода следует делать так, чтобы обеспечить максимальное удовлетворение требований к порядку вынесения переменных за скобки и выявлению общих выражений, при выполнении которых может быть построена ЛСА с минимальным (или близким к минимальному) числом членов. После этого, как и ранее, переходим к объединенной ЛСА.

Рассмотрим пример объединения трех следующих ЛСА:

$$\mathfrak{A}_1 = \downarrow^2 p_1 \uparrow^1 A_1 p_2 \uparrow^2 A_2 \downarrow^3 A_3 p_3 \uparrow^3 A_4 \downarrow^1 A_к; \tag{3.22}$$

$$\mathfrak{A}_2 = p_2 \uparrow^1 A_2 \downarrow^2 A_3 p_3 \uparrow^2 A_4 p_1 \uparrow^3 \downarrow^1 A_1 \downarrow^3 A_к; \tag{3.23}$$

$$\mathfrak{A}_3 = A_1 p_2 \uparrow^1 A_3 A_4 \downarrow^1 A_к. \tag{3.24}$$

Требуется построить объединенную ЛСА; так как *L*=3, число дополнительных переменных равно двум (дополнительные логические переменные r_1 и r_2). Очевидно, при этом одна определяющая конъюнкция будет неиспользуемой.

Если не осуществлять объединение заданных ЛСА, а получить общий алгоритм функционирования в соответствии с ЛСА, например, вида

$$\mathfrak{A} = r_1 \uparrow^1 \mathfrak{A}_1 \omega \uparrow^2 \downarrow^1 r_2 \uparrow^3 \mathfrak{A}_2 \omega \uparrow^2 \downarrow^3 \mathfrak{A}_3 \downarrow^2 A_к, \tag{3.25}$$

то общее число членов будет равно 21.

Для объединения ЛСА (3.21) —(3.24) перейдем к МСА (табл. 3.18—3.20).

Таблица 3.18

$$\mathfrak{A}_1 = \begin{array}{c} \\ A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \end{array} \begin{array}{c} \begin{array}{ccccc} A_1 & A_2 & A_3 & A_4 & A_к \end{array} \\ \begin{bmatrix} p_1 & & & & \bar{p}_1 \\ p_1\bar{p}_2 & p_2 & & & \bar{p}_1\bar{p}_2 \\ & & 1 & & \\ & & \bar{p}_3 & p_3 & \\ & & & & 1 \end{bmatrix} \end{array}$$

Таблица 3.19

$$\mathfrak{A}_2 = \begin{array}{c} \\ A_0 \\ A_1 \\ A_2 \\ A_3 \\ A_4 \end{array} \begin{array}{c} \begin{array}{ccccc} A_1 & A_2 & A_3 & A_4 & A_к \end{array} \\ \begin{bmatrix} \bar{p}_2 & p_2 & & & \\ & & & & 1 \\ & & 1 & & \\ & & \bar{p}_3 & p_3 & \\ p_1 & & & & \bar{p}_1 \end{bmatrix} \end{array}$$

Таблица 3.20

$$\mathfrak{A}_3 = \begin{array}{c} \\ A_0 \\ A_1 \\ A_3 \\ A_4 \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_3 & A_4 & A_к \end{array} \\ \begin{bmatrix} 1 & & & \\ & p_2 & & \bar{p}_2 \\ & & 1 & \\ & & & 1 \end{bmatrix} \end{array}$$

Теперь выберем определяющие конъюнкции R_1—R_3. Соседние определяющие конъюнкции должны быть приписаны тем парам ЛСА, для которых число совпадающих элементов МСА является максимальным. При этом упростится наибольшее число элементов объединенной МСА.

В соответствии с этим условием приписывания определяющих конъюнкций подсчитываем число одинаковых элементов различных пар МСА. При этом будем подсчитывать число совпадающих элементов между каждой из всех четырех МСА.

Так как элементы неиспользуемой МСА не определены, то при сравнении с ней число совпадающих элементов берется равным числу элементов используемой МСА, не равных нулю.

Кроме того, заметим, что если из сравнения двух используемых МСА выяснится, что оператор A_i входит лишь в МСА $\mathcal{A}_j$, а в МСА $\mathcal{A}_l$ не содержится, то в число совпадающих элементов МСА $\mathcal{A}_j$ и $\mathcal{A}_l$ входят все ненулевые элементы строки A_i МСА $\mathcal{A}_j$.

Для наглядности связи между МСА изобразим в виде графа, каждому ребру которого припишем цифру, указывающую число одинаковых элементов у МСА, приписанных вершинам, соединенным этим ребром. Для нашего примера такой отмеченный граф изображен на рис. 3.30, где $\mathcal{A}_{\text{Н}}$ — вершина, соответствующая неиспользуемой МСА.

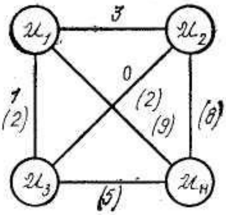

Рис. 3.30.

Тем парам МСА, которым соответствует наибольшее число совпадающих элементов, должны быть приписаны соседние определяющие конъюнкции. Надо заметить, что такой способ выявления определяющих конъюнкций не гарантирует построения объединенной ЛСА с минимальным числом членов. Однако он позволяет получить вполне приемлемые решения. Кроме того, надо иметь в виду, что наибольшее упрощение может быть получено в том случае, когда соседние определяющие конъюнкции приписываются тем парам МСА, у которых одинаковыми являются достаточно сложные элементы. Поэтому целесообразно в первую очередь приписывать таким парам МСА соседние определяющие конъюнкции, если даже у них число одинаковых элементов меньше, чем у других пар МСА.

В рассматриваемом примере таких сложных одинаковых элементов нет, поэтому выявление определяющих конъюнкций осуществлено в соответствии, с графом, показанным на рис. 3.30, т. е.

$$R_1=r_1r_2;\ R_2=r_1\overline{r}_2;\ R_3=\overline{r}_1\, r_2;\ R_{\text{Н}}=\overline{r}_1\ \overline{r}_2,$$

где $R_{\text{Н}}$ —неиспользуемая определяющая конъюнкция. Теперь получим определяющие функции β_i^l:

$$\beta_0^1=\beta_1^1=\beta_3^1=\beta_4^1=r_1r_2\vee\frac{\overline{r}_1\overline{r}_2}{0}=r_1r_2;$$

$$\beta_2^1=r_1r_2\vee\frac{\overline{r}_1\overline{r}_2}{0}\vee\frac{\overline{r}_1r_2}{0}=\frac{r_1}{1}\,r_2;$$

$$\beta_0^2=\beta_1^2=\beta_3^2=\beta_4^2=r_1\overline{r}_2\vee\frac{\overline{r}_1\overline{r}_2}{0}=\frac{r_1}{1}\,\overline{r}_2;$$

$$\beta_2^2=r_1r_2\vee\frac{\overline{r}_1\overline{r}_2}{0}\vee\frac{\overline{r}_1r_2}{0}=\frac{r_1}{1}\,r_2;$$

$$\beta_0^3=\beta_1^3=\beta_3^3=\beta_4^3=\overline{r}_1r_2\vee\frac{\overline{r}_1\overline{r}_2}{0}=\overline{r}_1\,\frac{r_2}{1}.$$

Строим недетерминированную МСА (табл. 3.21**):**

Таблица 3.21

$$
\begin{array}{c|cccccc|}
 & A_1 & A_2 & A_3 & A_4 & A_{\text{к}} \\
\hline
A_0 & r_1r_2p_1 \vee \frac{r_1}{1}\bar{r}_2\bar{p}_2 \vee \bar{r}_1\frac{r_2}{1} & \frac{r_1}{1}\bar{r}_2 p_2 & & & r_1r_2\bar{p}_1 \\
A_1 & r_1r_2p_1\bar{p}_2 & r_1r_2p_2 & \bar{r}_1\frac{r_2}{1}p_2 & & r_1r_2\bar{p}_1\bar{p}_2 \vee \frac{r_1}{1}\bar{r}_2 \vee \bar{r}_1\frac{r_2}{1}\bar{p}_2 \\
\hat{\mathfrak{A}}=A_2 & & & \frac{r_1}{1}r_2 \vee \frac{r_1}{1}\bar{r}_2 = 1 & & \\
A_3 & & & r_1r_2\bar{p}_3 \vee \frac{r_1}{1}\times \bar{r}_2\bar{p}_3 = r_1\bar{p}_3 & r_1r_2p_3 \vee \frac{r_1}{1}\times \bar{r}_2\bar{p}_3 \vee \bar{r}_1 \times \frac{r_2}{1} = r_1p_3 \vee \bar{r}_1 & \\
A_4 & \frac{r_1}{1}\bar{r}_2 p_1 & & & & r_1r_2 \vee \frac{r_1}{1}\bar{r}_2\bar{p}_1 \vee \bar{r}_1\frac{r_2}{1} = r_2 \vee \frac{r_1}{1}\bar{r}_2\bar{p}_1 \\
\hline
\end{array}
$$

От недетерминированной МСА переходим к недоопределенным формулам перехода, которые затем преобразовываем:

$$A_0 \to r_1 r_2 p_1 A_1 \vee \frac{r_1}{1} \bar{r}_2 \bar{p}_2 A_1 \vee \bar{r}_1 \frac{r_2}{1} A_1 \vee \frac{r_1}{1} \bar{r}_2 p_2 A_2 \vee r_1 r_2 \bar{p}_1 A_{\text{к}} =$$

$$= r_1 \left(r_2 \downarrow^1 (p_1 A_1 \vee \bar{p}_1 A_{\text{к}}) \vee \bar{r}_2 (p_2 A_2 \vee \bar{p}_2 A_1)\right) \vee \bar{r}_1 A_1;$$

$$A_1 \to r_1 r_2 p_1 \bar{p}_2 A_1 \vee r_1 r_2 p_2 A_2 \vee \bar{r}_1 \frac{r_2}{1} p_2 A_3 \vee r_1 r_2 \bar{p}_1 \bar{p}_2 A_{\text{к}} \vee \frac{r_1}{1} \bar{r}_2 A_{\text{к}} \vee$$

$$\vee \bar{r}_1 \frac{r_2}{1} \bar{p}_2 A_{\text{к}} = r_1 \left(r_2 (p_2 A_1 \vee \bar{p}_2 (p_1 A_2 \vee \bar{p}_1 A_{\text{к}}) \vee \bar{r}_2 A_{\text{к}})\right) \vee \bar{r}_1 (p_2 A_3 \vee \bar{p}_2 A_{\text{к}}) =$$

$$= r_1 \left(r_2 (p_2 A_2 \vee \bar{p}_2 \omega \uparrow^1) \vee \bar{r}_2 A_{\text{к}}\right) \vee \bar{r}_1 (p_2 A_3 \vee \bar{p}_2 A_{\text{к}});$$

$$A_2 \to A_3;$$

$$A_3 \to r_1 \bar{p}_3 A_3 \vee r_1 p_3 A_4 \vee \bar{r}_1 A_4 = r_1 (\bar{p}_3 A_3 \vee p_3 A_4) \vee \bar{r}_1 A_4;$$

$$A_4 \to \frac{r_1}{1} \bar{r}_2 p_1 A_1 \vee r_2 A_{\text{к}} \vee \frac{r_1}{1} \bar{r}_2 \bar{p}_1 A_{\text{к}} = r_2 A_{\text{к}} \vee \bar{r}_2 (p_1 A_1 \vee \bar{p}_1 A_{\text{к}}) =$$

$$= r_2 A_{\text{к}} \vee \bar{r}_2 \omega \uparrow^1.$$

После этого составляем объединенную ЛСА:

$$\mathfrak{A} = r_1 \uparrow^1 r_2 \uparrow^2 \downarrow^6 p_1 \uparrow^7 \downarrow^1 A_1 r_1 \uparrow^4 r_2 \uparrow^7 p_2 \uparrow^6 \downarrow^3 A_2 \downarrow^5 A_3 r_1 \uparrow^8 p_3 \uparrow^5 \downarrow^8 A_4 r_2 \uparrow^6 \omega \uparrow^7 \downarrow^2 p_2 \uparrow^1 \omega \uparrow^3$$
$$\downarrow^4 p_2 \uparrow^7 \omega \uparrow^5 \downarrow^7 A_{\text{к}}, \tag{3.26}$$

в которой содержится 16 членов вместо 21 в ЛСА (3.25).

В рассмотренном примере при определенных значениях дополнительных ЛУ выполняется одна из частных ЛСА, после чего процесс заканчивается.

Значения дополнительных логических условий r_1 и r_2 при этом должны быть заданы или каким-либо образом выбраны. Однако возможны случаи, когда частные ЛСА выполняются одна за другой и после выполнения всех ЛСА процесс повторяется. Тогда общий алгоритм функционирования САК запишем в виде

$$\mathfrak{A} = \downarrow^1 \mathfrak{A}_1 \mathfrak{A}_2 \ldots \mathfrak{A}_L \omega \uparrow^1, \tag{3.27}$$

при этом изменение значений дополнительных логических условий должно производиться в процессе выполнения общей ЛСА, а для формирования соответствующих их значений могут быть введены новые операторы

$$F_{\mathfrak{A}_1}, F_{\mathfrak{A}_2}, \ldots, F_{\mathfrak{A}_L},$$

каждый из которых является одновременно оператором начала соответствующей ЛСА.

Таким образом, в каждой из частных ЛСА вместо оператора A_0 будет оператор F_{A_i} , который вместе с тем будет оператором конца ЛСА A_{i-1} Теперь

$$\mathfrak{A} = \downarrow^1 F_{\mathfrak{A}_1} \mathfrak{A}_1 F_{\mathfrak{A}_2} \mathfrak{A}_2 \ldots F_{\mathfrak{A}_L} \mathfrak{A}_L \omega \uparrow^1 . \tag{3.28}$$

В ЛСА (3.28) порядок выполнения частных ЛСА единственный, однако может быть задан и некоторый алгоритм над частными ЛСА, так что допускается несколько различных последовательностей выполнений частных ЛСА. Тогда в общей ЛСА имеются внешние ЛУ, значения которых определяют эти последовательности. При объединении ЛСА эти условия входят в МСА объединенной ЛСА.

Хотя объединение частных ЛСА не всегда приводит к сокращению числа членов, но в большинстве практических случаев объединенная ЛСА содержит значительно меньшее число членов, чем общая ЛСА, в которой частные ЛСА не объединены.

3.3.4. Описание параллельных консультационных алгоритмов

Стремление к повышению производительности САК и ее многоблочная структура часто приводят к необходимости распараллеливания консультационного алгоритма. При этом язык задания алгоритма функционирования САК должен обладать средствами для описания параллелизма, которых, очевидно, не имеет язык ЛСА.

Предложено расширение языка ЛСА — язык ЛСАП, который позволяет описывать алгоритмы функционирования с параллельными участками. Запись на языке ЛСАП может иметь, например, следующий вид:

$$\mathfrak{A} = A_0 p_1 \uparrow^1 A_1 \downarrow^1 \left[\begin{array}{l} \to A_2 A_3 \left[\begin{array}{l} \to A_7 p_3 \uparrow^3 A_8 \downarrow^3 \\ \to A_9 p_4 \uparrow^4 A_{10} \downarrow^4 \end{array}\right. \\ \left.\begin{array}{l} \to p_2 \uparrow^2 A_4 \downarrow^2 \\ \to A_5 A_6 \end{array}\right] \to p_5 \uparrow^5 A_{11} \downarrow^5 \end{array}\right] \to A_{12} A_k . \tag{3.29}$$

Такая запись означает, что после выполнения оператора A_1 или при ложном значении логического условия p_1 начинается одновременное выполнение трех участков ЛСА. Затем после выполнения операторов A_4 и A_6 два участка соединяются, а после выполнения оператора A_3 алгоритм опять распараллеливается и т. д.

Этот язык удобен своей наглядностью, однако он недостаточно формализован, что ограничивает его использование для выполнения различных преобразований алгоритма функционирования. Кроме того, являясь фактически сочетанием логических и граф-схем алгоритмов, язык ЛСАП может привести к громоздкому описанию сложных алгоритмов.

Теперь рассмотрим другое расширение языка ЛСА — *язык параллельных логических схем алгоритмов,* в котором с помощью введения в язык ЛСА специальных операторов и дополнительных символов сохранена линейная запись алгоритма с параллельными участками.

Для расширения возможностей языка ЛСА введем *оператор распараллеливания R*. Оператор *R* подобно оператору A_0 (начало алгоритма) является фиктивным, он лишь символизирует переход к выполнению параллельных частей алгоритма. Поскольку после оператора *R* начинается выполнение параллельных участков алгоритма, этот оператор в отличие от других операторов ЛСА имеет несколько последователей, которые являются начальными членами параллельных участков. Для перехода к параллельным участкам введем специальные стрелки $\underline{\uparrow}_j^{\,i}\,\overline{\downarrow}^{\,i}$, которые отличаются от имеющихся в языке ЛСА *стрелок разветвления* $\uparrow^{i}\downarrow^{i}$ — *стрелки распараллеливания.* Нижний индекс *j* стрелки распараллеливания определяет число параллельных участков, следующих за данным оператором распараллеливания.

Справа от оператора распараллеливания *R* будем записывать тождественно-ложное логическое условие ω со стрелкой распараллеливания $\underline{\uparrow}_j^{\,i}\,(\omega\,\underline{\uparrow}_j^{\,i})$. Концы *i*-й стрелки $\overline{\downarrow}^{\,i}$ записываются перед первыми членами параллельных участков, которые начинаются после данного оператора распараллеливания. Очевидно, что число таких стрелок равно *j*. Заметим, что нижний индекс стрелки распараллеливания не определяет последовательность выполнения отдельных шагов алгоритма, а играет лишь вспомогательную роль и введен для проверки правильности записи алгоритма. Справа от каждой стрелки $\overline{\downarrow}^{\,i}$ последовательно, как в языке ЛСА, записываются члены каждого параллельного участка.

Таким образом, с помощью оператора и стрелок распараллеливания можно осуществить переход к параллельным участкам алгоритма.

Для соединения параллельных участков введем *оператор соединения S* и *стрелки соединения* $\underline{\uparrow}^{\,k}\,\overline{\downarrow}^{\,k}_{m}$. После последнего члена каждого параллельного участка запишем тождественно-ложное условие ω со стрелкой соединения $\underline{\uparrow}^{\,k}(\omega\,\underline{\uparrow}^{\,k})$.

Конец *k*-й стрелки соединения $\overset{\frown}{\downarrow}{}^{k}_{m}$ записывается перед оператором соединения *S*, в котором соединяются данные параллельные участки. Нижний индекс стрелки соединения *m* играет вспомогательную роль, указывая число параллельных участков, которые соединяются в данном операторе соединения, т. е. число стрелок $\underset{\smile}{\uparrow}{}^{k}$.

Оператор *S*, как и оператор *R*, является фиктивным, символизирующим конец выполнения параллельных участков. Оператор соединения *S* может быть выполнен только после того, как закончится выполнение всех соединяющихся в нем параллельных участков. Справа от оператора *S* записывается первый член последовательного участка алгоритма, который должен выполняться после окончания предшествующих параллельных участков.

Таким образом, использование операторов и стрелок соединения позволяет осуществлять переход от параллельных участков алгоритма к последовательным.

Логическую схему алгоритма, содержащую кроме обычных операторов, логических условий и стрелок разветвления также операторы и стрелки распараллеливания и соединения, будем называть *параллельной логической схемой алгоритма* (ПЛСА).

Будем рассматривать такие ПЛСА, в которых каждый параллельный участок является замкнутым, т. е. имеет только один вход и один выход. Каждый такой участок можно считать отдельным подалгоритмом, введя в него фиктивные операторы, символизирующие начало и окончание выполнения этого подалгоритма, подобно операторам A_0 и $A_\text{к}$ в ЛСА. Для *i*-го подалгоритма обозначим эти операторы B_0^i и $B_\text{к}^i$ соответственно (нумерация самих подалгоритмов не имеет значения). При этом приведенная ранее ЛСАП (3.29) может быть представлена в виде следующей ПЛСА:

$$\begin{gathered}
\mathfrak{A} = A_0 B_0^0 p_1 \uparrow^1 A_1 \downarrow^1 B^0_\text{к} R_1 \omega \underset{\smile}{\uparrow}{}_3^6 \overset{\frown}{\downarrow}{}^6 B_0^1 A_2 A_3 B_k^1 R_2 \omega \underset{\smile}{\uparrow}{}_2^7 \overset{\frown}{\downarrow}{}^7 B_0^4 A_7 \\
p_3 \uparrow^3 A_8 \downarrow^3 B_k^4 \omega \underset{\smile}{\uparrow}{}^8 \overset{\frown}{\downarrow}{}^7 B_0^5 A_9 p_4 \uparrow^4 A_{10} \downarrow^4 B_\text{к}^5 \omega \underset{\smile}{\uparrow}{}^8 \overset{\frown}{\downarrow}{}^6 B_0^2 \\
p_2 \uparrow^2 A_4 \downarrow^2 B_k^2 \omega \underset{\smile}{\uparrow}{}^9 \overset{\frown}{\downarrow}{}^6 B_0^3 A_5 A_6 B_\text{к}^3 \omega \underset{\smile}{\uparrow}{}^9 \overset{\frown}{\downarrow}{}_2^9 S_1 B_0^6 p_5 \\
\uparrow^5 A_{11} \downarrow^5 B_\text{к}^6 \omega \underset{\smile}{\uparrow}{}^8 \overset{\frown}{\downarrow}{}_3^8 S_2 A_{12}.
\end{gathered} \tag{3.30}$$

Здесь фиктивные операторы B_0 и $B_\text{к}$ введены во все подалгоритмы, так что каждый из них не содержит ни операторов, ни стрелок распараллеливания и соединения и описывается своей ЛСА. При этом ПЛСА может быть записана в более компактной форме с использованием обозначений соответствующих подалгоритмов:

$$\mathfrak{A} = \mathfrak{A}_0 R_1 \omega \underline{\uparrow}_3^{6} \underline{\uparrow}^{6} \mathfrak{A}_1 R_2 \omega \underline{\uparrow}_2^{7} \overline{\downarrow}^{7} \mathfrak{A}_4 \omega \underset{\frown}{\uparrow}^{8} \overline{\downarrow}^{7} \mathfrak{A}_5$$
$$\omega \underset{\frown}{\uparrow}^{8} \overline{\downarrow}^{6} \mathfrak{A}_2 \omega \underset{\frown}{\uparrow}^{9} \overline{\downarrow}^{6} \mathfrak{A}_3 \omega \underset{\frown}{\uparrow}^{9} \overset{\smile}{\downarrow}_2^{9} S_1 \mathfrak{A}_6 \omega \underset{\frown}{\uparrow}^{8} \overset{\smile}{\downarrow}_3^{8} S_2 A_{12}. \tag{3.31}$$

Для каждого из подалгоритмов можно составить МСА и проверить такие важные свойства, как полноту и непротиворечивость. По аналогии с МСА может быть введена и параллельная матричная схема алгоритма (ПМСА). Параллельная МСА также является квадратной матрицей, строки и столбцы которой соответствуют операторам ПЛСА. Все элементы ПМСА имеют тот же смысл, что и в МСА, за исключением строк операторов распараллеливания и столбцов операторов соединения. Строка оператора *R* содержит несколько единичных элементов, число которых равно нижнему индексу стрелки распараллеливания этого оператора в ПЛСА. Очевидно, что формула перехода оператора распараллеливания отличается от формул перехода других операторов. Иначе чем для других столбцов ПМСА необходимо трактовать наличие нескольких единичных элементов в столбце оператора соединения. Они определяют всех предшественников, которые должны быть выполнены для перехода к этому оператору соединения. Для других операторов необходимо выполнение лишь одного из возможных предшественников.

Такие же особенности, вытекающие из свойств параллельного алгоритма, характерны и для параллельной граф-схемы алгоритма (ПГСА). Вершина графа, соответствующая оператору *R,* имеет несколько последователей в отличие от всех других операторов. В вершине, соответствующей оператору *S*, параллельно соединяется несколько путей графа в отличие от альтернативного соединения в других его вершинах. На рис. 3.31 приведен пример ПГСА для ПЛСА (3.31), подалгоритм A_4 представлен в терминах операторов и ЛУ в соответствии с ПЛСА (3.30).

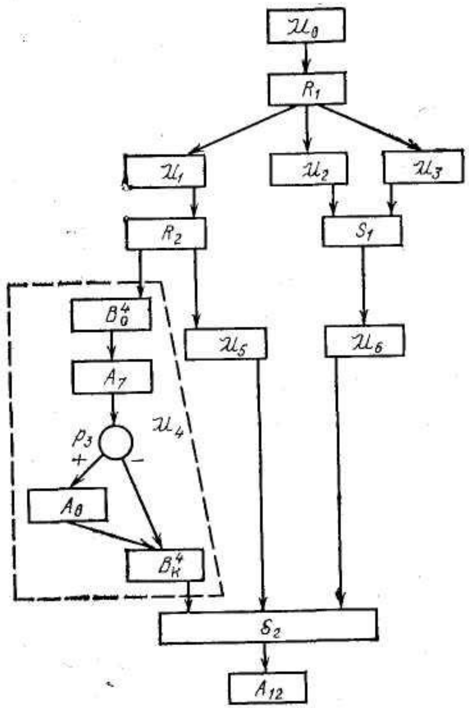

Рис. 3.31.

Здесь в операторе $B_к^4$ альтернативное соединение ветвей алгоритма, тогда как в S_1 и S_2 —параллельное.

Ясно, что для сложного параллельного консультационного алгоритма ПМСА и ПГСА оказываются очень громоздкими, что затрудняет их использование для проверки и преобразования алгоритма.

3.3.5. Переход от правильного консултационного процесса к консультационному алгоритму

В предыдущих разделах показано, что ряд задач по преобразованию и упрощению консультационного алгоритма удобно решать с использованием языка ЛСА. В то же время консультационный процесс, реализуемый в САК, с учетом функциональных и ЛР, необходимых для выполнения процедур процесса, задается на языке сетей Петри. Очевидно, для использования различных языков на разных этапах формирования рекомендаций решения консультационных задач необходимо установить связь между ЛСА, ПЛСА и сетями Петри.

Осуществим вначале переход от ЛСА к сети Петри, описывающей тот же последовательный процесс. Заметим, что возможны различные интерпретации ЛСА в терминах языка сетей Петри. Рассмотрим одну из них, основанную на использовании МСА.

Первой строке A_0 МСА сопоставим переход t_0^0 сети Петри, входной позицией которого является позиция a_0. Появление метки в позиции a_0, вызывающее реализацию перехода t_0^0, соответствует началу выполнения алгоритма. После реализации перехода t_0^0 появляется метка в его выходной позиции, которая является входной для перехода t_1^j, если в строке оператора A_0 МСА заполнена только одна клетка $\alpha_{0,j}$ (очевидно, она содержит 1). Если оператор A_0 имеет несколько последователей, например A_l, A_s, A_q, т. е. в МСА заполнены клетки $\alpha_{0,l}$, $\alpha_{0,s}$, $\alpha_{0,q}$, выходная позиция перехода t_0^0 является позицией альтернативного разветвления. Она будет общей входной позицией переходов t_1^l, t_2^s, t_3^q, которые помечаются конъюнкциями, записанными в соответствующих клетках МСА. Обратившись затем к строке оператора A_l и определив его последователей, можно таким же образом построить фрагмент сети Петри для перехода t_1^l. Процесс построения сети Петри по МСА заканчивается после того, как будут рассмотрены все строки матрицы и введены все переходы $t_j^к$, число которых должно быть равно числу заполненных клеток столбца $A_к$ МСА. Ясно, что эти переходы не имеют выходных позиций. Если для ЛСА (МСА) было задано распределение сдвигов ее операторов, эта информация должна быть сохранена и для сети Петри введением в нее позиций ЛР. На рис. 3.32 показан граф сети Петри, построенный по МСА (табл. 3.12) в соответствии с приведенными правилами (без учета распределения сдвигов) .

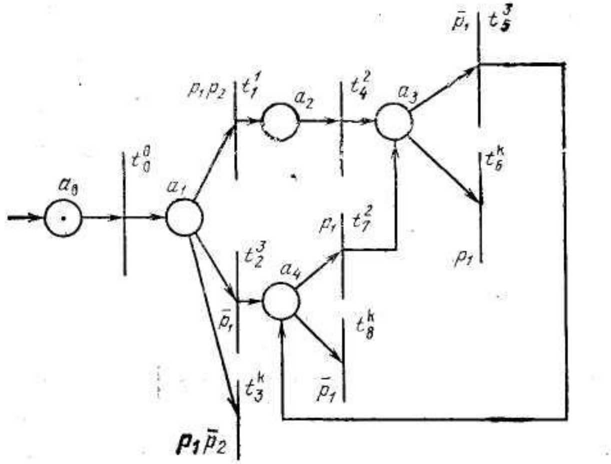

Рис. 3.32.

Как видно, число переходов получаемой сети Петри на единицу больше числа заполненных клеток МСА, а число позиций равно числу различных операторов, так что позиция a_l получает метку после реализации перехода t_j^{l-1}. Переходы-последователи позиции a_l определяются формулой перехода оператора A_{l-1}, т. е. строкой A_{l-1} МСА. Если оператор A_{l-1} имеет несколько предшественников, позиция a_l является позицией альтернативного соединения.

Аналогичным образом можно осуществить переход от ПЛСА к сети Петри, сопоставив с каждым оператором R переход распараллеливания, число выходных позиций которого равно числу единиц в строке ПМСА этого оператора, а с оператором S переход соединения с числом входных позиций, равным числу единиц в столбце ПМСА этого оператора. Для каждого подалгоритма ПЛСА сеть Петри строится так же, как для ЛСА. Очевидно, при этом сеть Петри может содержать избыточное число переходов и позиций из-за большого числа введенных в ПЛСА фиктивных операторов начала и окончания подалгоритмов. Поскольку эти операторы были введены специально, чтобы иметь одинаковые правила перехода от ЛСА и ПЛСА к сети Петри, соответствующие им переходы и позиции могут быть затем исключены из сети.

Рассмотрим способ перехода от сети Петри, описывающей правильный КП, к ПЛСА или ЛСА, если процесс является последовательным. Сравнение языков сетей Петри и ПЛСА показывает, что существенным отличием сетей Петри является их способность описывать конфликтные состояния. Благодаря этому сетью Петри можно задать гораздо больше последовательностей выполнения процедур процесса, чем ПЛСА, которая описывает параллелизм, но не обладает средствами для задания конфликтов. Поэтому при переходе от сети Петри к ПЛСА необходимо либо разрешить конфликты путем проверок значений некоторых дополнительно введенных переменных, либо удалить позиции конфликтов, к которым относятся позиции ресурсов и блокирующие позиции. Во втором случае для сохранения всех ограничений на возможность одновременного выполнения процедур процесса, которые задаются этими позициями, приходится осуществлять преобразование сети Петри, выбирая лишь допустимую последовательность реализации переходов. С этой целью можно использовать граф достижимых маркировок сети Петри и совместить преобразование процесса с устранением тупиковых состояний.

На рис. 3.33 приведен граф статических состояний сети Петри рис. 3.15.

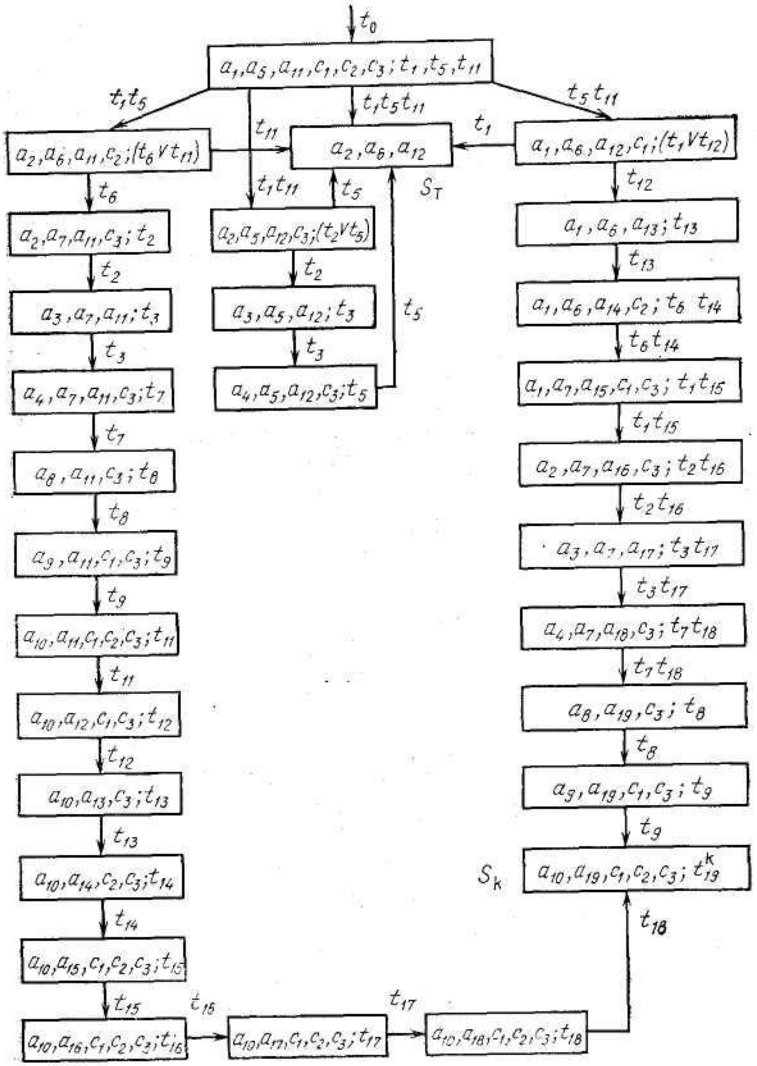

Рис. 3.33.

Как видно, этот граф содержит тупиковое состояние $S_Т$, в которое процесс попадает при одновременном развитии трех параллельных подпроцессов. Для того чтобы исключить возможность возникновения

тупикового состояния и получить правильный процесс, необходимо ввести блокирующую позицию, которая должна быть входной для переходов t_1 и t_{11}. При этом достижимыми станут только два пути графа, каждый из которых ведет в конечное состояние S_K. Однако эти пути не равноценны. Один из них соответствует практически последовательному выполнению процедур процесса, допуская одновременное выполнение лишь процедур A_5 и A_2 (переходы t_1^5 и t_5^2). В то же время второй путь позволяет процессу достигнуть конечного состояния за значительно меньшее время, так как предусматривает возможность одновременного выполнения многих процедур.
Поскольку при переходе к описанию процесса на языке ПЛСА мы вынуждены ограничить множество допустимых последовательностей выполнения процедур, целесообразно выбрать именно ту последовательность, которая соответствует кратчайшему пути графа маркировок и, следовательно, обеспечивает максимально возможный при заданном распределении ресурсов параллелизм. Учитывая выбранную таким образом последовательность, а также содержащуюся в исходной сети Петри информацию о зависимости процедур по ресурсам, можно осуществить ее преобразование с целью исключения всех блокирующих позиций, если они уже были введены для получения правильного КП, и позиций ресурсов.

На рис. 3.34 показана сеть Петри, полученная в результате такого преобразования сети рис. 3.15 с учетом кратчайшего пути графа достижимых маркировок (рис. 3.33).

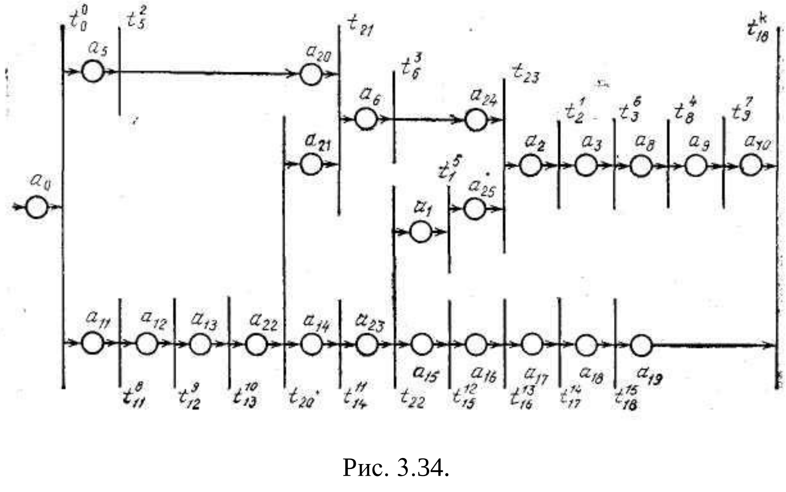

Рис. 3.34.

Как видно, в граф сети Петри введены дополнительные переходы распараллеливания и соединения и новые позиции, которые запрещают одновременное выполнение процедур, использующих одни и те же ФР.

От полученной таким образом сети Петри можно перейти к описанию этого процесса на языке ПЛСА. Для этого с каждым переходом t_j^i, т. е. с процедурой A_i процесса, сопоставим оператор A_i ПЛСА, с каждым переходом распараллеливания— оператор R, а с переходом соединения —оператор S. Если в сети Петри не были выделены переходы, соответствующие началу и окончанию процесса, операторы A_0 и $A_к$ могут быть введены в ПЛСА дополнительно.
Для сети Петри рис. 3.34 получим следующую ПЛСА:

$$\mathfrak{A} = A_0R_1\omega\underline{\uparrow}_2^{1}\overline{\downarrow}^{1}A_2\omega\underset{\frown}{\uparrow}^{2}\overline{\downarrow}^{1}A_8A_9A_{10}R_2\omega\underline{\uparrow}_2^{3}\overline{\downarrow}^{3}\omega\underset{\frown}{\uparrow}^{2}\overline{\downarrow}^{3}A_{11}R_3$$
$$\omega\underline{\uparrow}_2^{4}\overline{\downarrow}^{4}A_5\omega\underset{\frown}{\uparrow}^{5}\overline{\downarrow}^{4}A_{12}A_{13}A_{14}A_{15}\omega\underset{\frown}{\uparrow}^{6}\overset{\smile}{\downarrow}_2^{2}S_1A_3\omega\underset{\frown}{\uparrow}^{5}\overset{\smile}{\downarrow}_2^{5}S_2A_1A_6A_4A_7$$
$$\omega\underset{\frown}{\uparrow}^{6}\overset{\smile}{\downarrow}_2^{6}S_3A_к.$$

Каждый параллельный участок ПЛСА можно оформить в виде отдельного подалгоритма, введя операторы B_0 и $B_к$.

Рассмотрим особенности перехода от сети Петри разветвленного процесса к ПЛСА. Если этот процесс является последовательным, то он описывается ЛСА, содержащей толька операторы и ЛУ. Переходу t_j^i сопоставляется оператор ЛСА A_i. Если выходной позицией перехода t_j^i является позиция альтернативного разветвления, то в ЛСА после A_i вводятся ЛУ, наборы значений которых разрешают конфликт переходов— последователей этой позиции. При этом не различаются внутренние и внешние ЛУ, а следующие за ними стрелки разветвления проставляются в соответствии с заданной сетью Петри последовательностью выполнения процедур. Информация о влиянии процедур на значения ЛУ задается в виде распределения сдвигов операторов ЛСА. В результате исключаются все позиции d_s и $\overline{d}_s$, причем последние заменяются логическим условием p_s.

Сеть Петри рис. 3.20 может быть описана следующей ЛСА:

$$\mathfrak{A} = A_0\downarrow^{2}A_1p_1\uparrow^{1}A_2\omega\uparrow^{2}\downarrow^{1}A_3p_2\uparrow^{3}A_4\omega\uparrow^{2}\downarrow^{3}A_5A_к$$

и распределением сдвигов: A_2—$\{p_1\}$, A_1—$\{p_2\}$ (остальные операторы имеют пустое распределение сдвигов). Ясно, что при этом вновь происходит потеря информации, содержащейся в исходной сети Петри. Из рис. 3.20 видно, какие значения принимают логические условия p_1 и p_2 после выполнения процедур A_2 и A_1. Распределение сдвигов задает лишь возможность изменения операторами ЛСА значений ЛУ.

Аналогичным образом можно перейти к ПЛСА от сети Петри, задающей разветвленный параллельный процесс. Очевидно, получаемая ПЛСА эквивалентна первоначальной сети Петри в том смысле, что она описывает некоторое подмножество всех возможных последовательностей выполнения процедур КП и может рассматриваться как одна из его допустимых реализаций.

Следует подчеркнуть, что прежде чем переходить к ПЛСА от сети Петри, необходимо перевести задаваемый ею процесс в правильный, так как существующие методы анализа ПЛСА не позволяют обнаруживать и устранять тупиковые состояния. Например, по сети Петри рис. 3.19 можно составить следующую ПЛСА:

$$\mathfrak{A} = A_0 R_1 \omega \underline{\uparrow}_2^{1} \overline{\downarrow}^{1} p_1 \uparrow^{2} A_1 A_5 \omega \underset{\frown}{\uparrow}^{3} \downarrow^{2} A_2 \omega \underset{\frown}{\uparrow}^{4} \overline{\downarrow}^{1} A_3 \omega \underset{\frown}{\uparrow}^{4} \overset{\smile}{\downarrow}_2^{4} S_1$$

$$A_4 \omega \underset{\frown}{\uparrow}^{3} \overset{\smile}{\downarrow}_2^{3} S_2 A_к.$$

Поскольку ФР C_2 используется процедурами разных альтернативных ветвей, позиция c_2 не накладывает дополнительных ограничений на последовательность выполнения процедур. По этой ПЛСА уже трудно выявить наличие тупикового состояния в данном процессе, тогда как это сразу выявляется по графу достижимости (см. рис. 3.22).

В то же время проверку полноты и непротиворечивости удобно делать по ПЛСА, составляя МСА для отдельных подалгоритмов. Для объединения частных алгоритмов, их преобразования и упрощения с учетом распределения сдвигов также используются ЛСА и МСА. Таким образом, *при формировании рекомендаций с использованием консультационных процессов целесообразно применять различные языки, имея средства перехода от одного языка к другому*.

Необходимо иметь в виду, что граф статических состояний сети Петри дает неполную информацию о возможных последовательностях выполнения процедур процесса, поэтому рассмотрение полного графа с учетом длительности реализации переходов может привести к получению другой ПЛСА. Заметим, что для процессов, не зависящих от скорости, результаты анализа по графу статических состояний и по полному графу совпадают, однако в общем случае необходимо использовать полный граф, несмотря на его сложность.

Проиллюстрируем это на примере сети Петри рис. 3.35, описывающей параллельный разветвленный процесс с начальным значением внутреннего логического условия p_2 равным нулю.

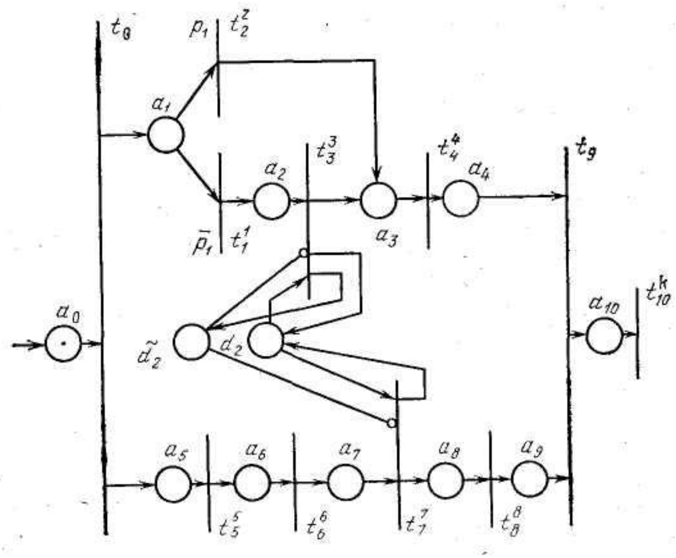

Рис. 3.35.

Все процедуры процесса используют собственные ФР, которые поэтому в описание процесса не включены. На рис. 3.36 приведена часть полного графа достижимых маркировок, соответствующая развитию процесса при ложном значении внешнего логического условия p_1.

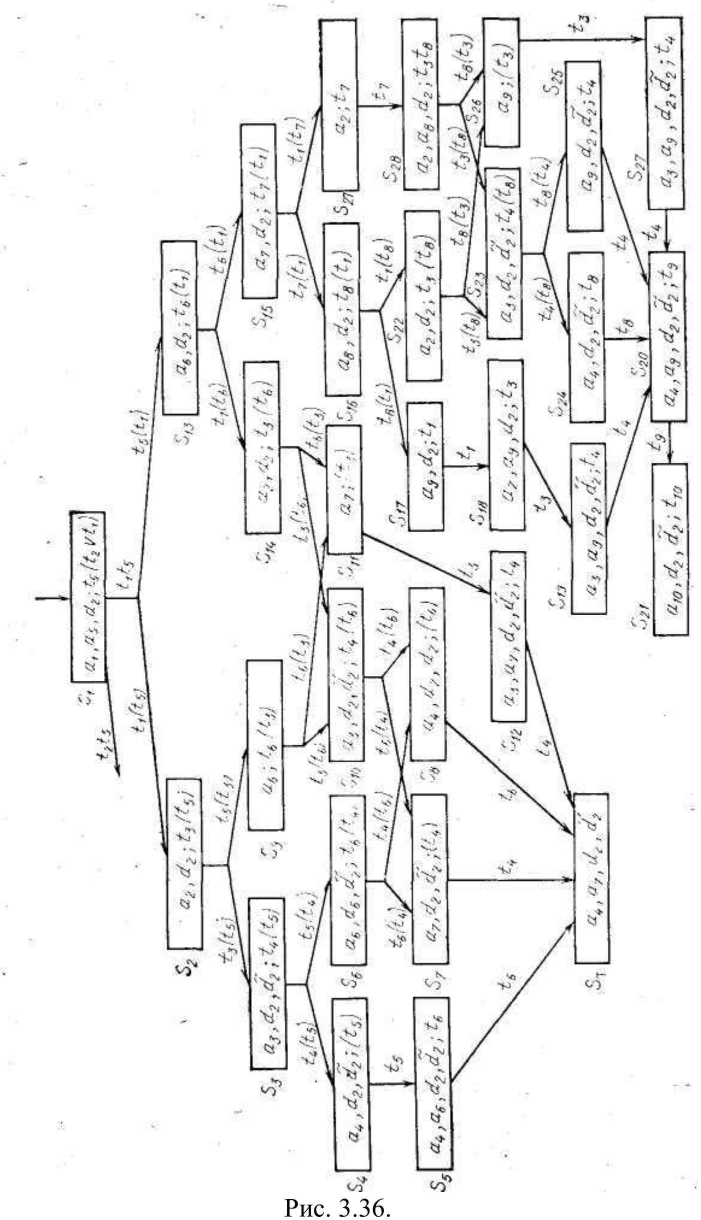

Рис. 3.36.

Как видно, в зависимости от длительности реализации отдельных переходов процесс либо успешно завершается, переходя в заключительное состояние S_{21}, либо попадает в тупиковое состояние $S_{\text{Т}}$. Для предотвращения развития процесса, приводящего в тупик, нужно обеспечить переход его в состояние S_{15}, после которого все пути, имеющие одинаковую длину, ведут в конечное состояние. Преобразованная в соответствии с этим требованием сеть Петри приведена на рис. 3.37, по которой получена следующая ПЛСА:

$$\mathfrak{A} = A_0 R_1 \omega \underline{\uparrow}_2^{\,1} \overline{\downarrow}^{1} \bar{p}_1 \uparrow^{2} \omega \underset{\frown}{\uparrow}^{3} \overline{\downarrow}^{1} A_5 A_6 A_7 R_2 \omega \underline{\uparrow}_2^{\,4} \overline{\downarrow}^{4} \omega \underset{\frown}{\uparrow}^{3} \overset{\smile}{\downarrow}_2^{\,3} S_1$$

$$A_3 \downarrow^{2} A_4 \omega \underset{\frown}{\uparrow}^{5} \overline{\downarrow}^{4} A_8 \omega \underset{\frown}{\uparrow}^{5} \overset{\smile}{\downarrow}_2^{\,5} S_2 A_{\text{к}}.$$

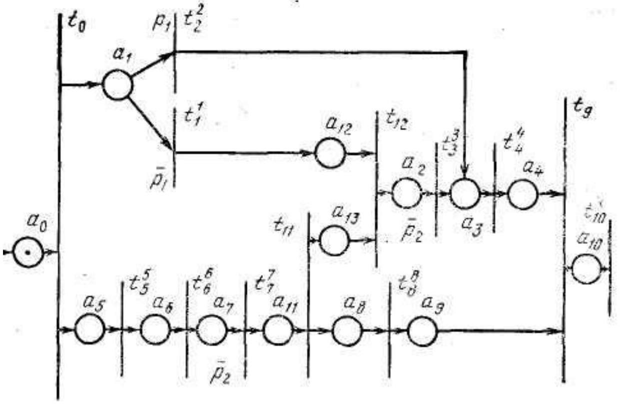

Рис. 3.37.

Если использовать граф статических состояний сети Петри рис. 3.35, то можно сделать неверный вывод о том, что при p_1=0 процесс всегда попадает в тупиковое состояние. В действительности, как видно из рис. 3.36, это происходит только при определенном соотношении длительностей реализации переходов t_5, t_6 и t_1. Когда для описания процесса используется временная сеть Петри и с каждым переходом t_j сопоставляется длительность его реализации Δ_j, можно проверить, выполняются ли необходимые временные соотношения. Так, в нашем примере, если $\Delta_1>\Delta_5+\Delta_6$, то тупиковое состояние недостижимо и никаких преобразований исходной сети Петри не требуется. Поскольку в нашем случае длительность реализации перехода не известна, осуществляется преобразование сети Петри с целью принудительного выполнения условия, устраняющего тупиковое состояние.

Таким образом, для *перехода от сети Петри к ПЛСА* необходимо *построить полный граф достижимых маркировок, выбрать в нем кратчайший путь, приводящий процесс в конечное состояние, и в соответствии с ним осуществить преобразование сети Петри, исключив все позиции конфликтов.* По полученной сети Петри *составляют ПЛСА*, а по исходной — *распределение сдвигов*. После этого проверяют ПЛСА на полноту и непротиворечивость с целью получения ПЛСА, задающей правильный консультационный алгоритм, которая затем используется для построения (формирования) рекомендаций.

Конечно, не исключается возможность синтеза рекомендаций непосредственно по сети Петри. Это наиболее целесообразно при рассмотрении множества асинхронных взаимодействующих процессов, когда возникающие тупики устраняются, например, с применением механизма семафоров и предполагается программная реализация консультационного процесса.

3.4. Реализация консультационного алгоритма

3.4.1. Принципы реализации параллельного консультационного алгоритма

При реализации консультационного алгоритма в зависимости от его сложности, используемой элементной базы, особенностей консультируемой проблемы и т. п. могут быть использованы различные принципы реализации консультационного алгоритма, из которых рассмотрим четыре основных. При этом для наглядности принципы реализации параллельного консультационного алгоритма рассмотрим на примере алгоритма, представленного в виде ПГСА (рис. 3.38).

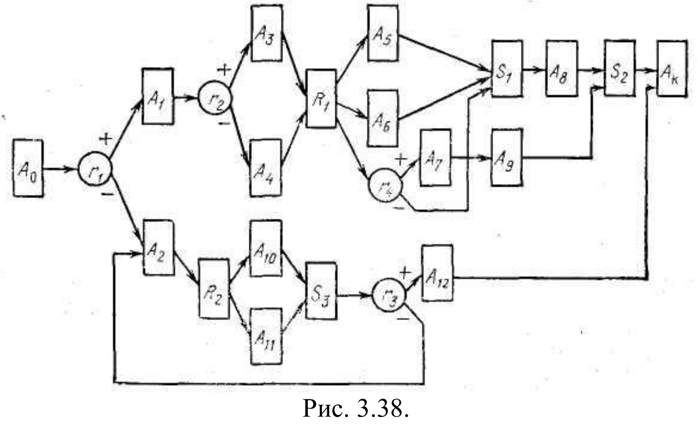

Рис. 3.38.

Заметим, что ПГСА вместо отдельных команд управления (операторов A_i и логических условий r_j), представляющих собой элементарные частные консультационные алгоритмы, могут быть представлены и достаточно сложные частные консультационные алгоритмы $\mathcal{A}_i$ и $\mathcal{A}_j$ (консультационные подалгоритмы), определяющие выполнение консультационного алгоритма на отдельных этапах или режимах его работы. При этом аналогами оператора A_i и логического условия r_j будут подалгоритм $\mathcal{A}_i$ с одним выходом (выходным портом) и подалгоритм $\mathcal{A}_j$ с двумя выходами (выходными портами). Число входов (входных портов) как в подалгоритме $\mathcal{A}_i$ с одним выходным портом, так и в подалгоритме $\mathcal{A}_j$ с двумя выходными портами может быть любым. В ПГСА, описывающей порядок выполнения подалгоритмов (т. е. структуру ПГСА), по-прежнему будем использовать операторы A_0, A_k, R_l и S_m.

Первый принцип. Наиболее наглядным и простым для понимания принципом реализации консультационного алгоритма является такой, при котором каждый оператор A_i и логическое условие, r_j (частные консультационные алгоритмы $\mathcal{A}_i$ и $\mathcal{A}_j$) реализуются отдельными программами САК. При этом будет получена совокупность операций, структура которой будет повторять структуру ПГСА. Таким образом, в нашем случае в совокупности операций, не считая входные и выходные операции, соответствующие операторам A_0 и A_k, будет использовано 19 операций, включая операции, предназначенные для распараллеливания и соединения параллельных подалгоритмов.
Сложность каждой операции такой совокупности зависит от функций, которые сопоставляются с соответствующим оператором и ЛУ. В случае применения в ПГСА частных консультационных алгоритмов каждая из операций совокупности может оказаться достаточно сложной, из-за чего, в свою очередь, потребуется ее декомпозиция на совокупность более простых операций, т. е. образуется двухуровневая иерархия совокупностей операций.

Таким образом, одноуровневая или двухуровневая (может быть и многоуровневая) совокупностей операций представляет собой *распределенный принцип построения консультационного процесса*.

Второй принцип. Альтернативным распределенному принципу реализации консультационного алгоритма может служить *централизованный принцип*, когда весь параллельный алгоритм реализуется одним КП, в каждом такте работы которого одновременно выполняется один или несколько операторов и (или) логических условий.

В том случае, когда в ПГСА вместо операторов и ЛУ указаны частные консультационные алгоритмы, вместо такта говорят о цикле (периоде) работы консультационного процесса, в течение которого для реализации соответствующего частного консультационного алгоритма может потребоваться несколько, может быть и довольно большое число тактов работы консультационного процесса.

При данном принципе реализация последовательного консультационного алгоритма не вызывает никаких трудностей. Однако при реализации параллельного консультационного процесса алгоритма в одной САК возникает ряд трудностей, связанных с выполнением параллельных ветвей алгоритма (параллельных подалгоритмов) в одном такте (цикле, периоде) работы САК.

Рассмотрим четыре возможных варианта сочетаний параллельно выполняемых членов (частных консультационных алгоритмов) ПГСА:

1. В параллельных ветвях ПГСА имеются только операторы (параллельные подалгоритмы с одним выходным портом). Для простоты, здесь далее будем рассматривать ПГСА с двумя параллельными ветвями.

Пусть в одной ветви имеется n операторов A_{i1}, ..., A_{in}, а в другой — m операторов A_{j1},..., A_{jm}, $m \neq n$, и для определенности $m<n$ (рис. 3.39).

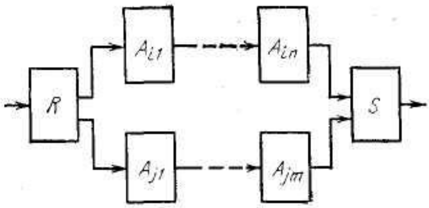

Рис. 3.39.

Тогда могут быть образованы m объединенных операторов A_{ij1},..., A_{ijm} и $n—m$ операторов A_{im+1},..., A_{in}, каждый из которых может реализоваться в отдельном такте работы САК.

2. В одной ветви ПГСА имеются только операторы, а во второй — хотя бы одно ЛУ. При этом ЛУ может находиться в начале, середине или конце ветви.

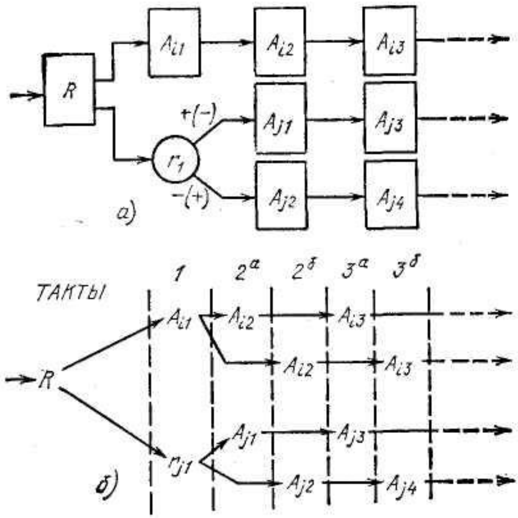

Рис. 3.40.

Рассмотрим эти три случая:

а) логическое условие находится в начале ветви (рис. 3.40, *а*). Очевидно, в одном и том же такте (например, такте 1) можно объединить оператор A_{i1} с логическими условиями r_{j1} (рис. 3.40, *б*). В связи с тем что после ЛУ r_{j1} происходит разветвление ПГСА, операторы A_{j1} и A_{j2}, находящиеся в альтернативных ветвях, не могут быть выполнены одновременно в одном и том же такте. Пусть оператор A_{j1} выполняется в такте $2^а$ при r_{j1}= 1 (стрелка со знаком + на рис. 3.40, *а*), а оператор A_{j2}— в такте $2^б$, следующим за тактом 1 при r_{ji}=0 (стрелка со знаком минус). Вместе с тем оператор A_{i2} выполняется независимо от значения ЛУ r_{j1} находящегося в параллельной ветви, т. е. оператор A_{i2} должен выполняться как при r_{j1} = 1, так и при r_{j1}=0. Поэтому оператор A_{i2} должен быть совмещен и с оператором A_{j1} (такт $2^а$) и с оператором A_{j2} (такт $2^б$). Аналогично последующие операторы A_{i3}, ..., A_{in} также должны совмещаться с операторами как одной, так и другой альтернативной ветвью второй параллельной ветви консультационного алгоритма;

б) логическое условие находится в середине ветви. До этого ЛУ совмещение операторов различных параллельных ветвей происходит так же, как это было описано в варианте 1, а начиная с этого логического условия — как в случае для варианта $2^а$;

в) логическое условие находится в конце ветви.

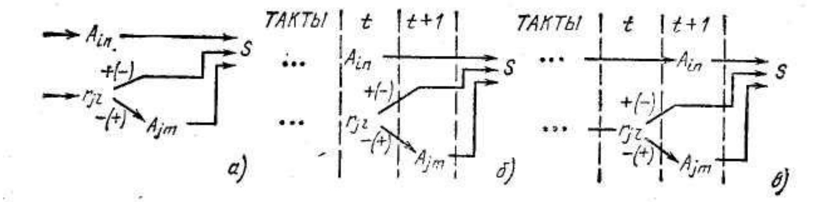

Рис. 3.41.

В этом случае (рис. 3.41, *а*) после логического условия r_{jl} в одной из альтернативных ветвей после r_{jl} =0 (знак —) или r_{jl} =1 (знак +) и только в одной из них перед оператором соединения *S* должен быть хотя бы один оператор (в нашем случае оператор A_{jm}). Тогда могут быть два варианта совмещения: в такте *t* совмещаются оператор A_{in} и логическое условие r_{jl} (рис. 3.41, *б*), а в такте *t* +1 выполняется только оператор A_{jm}, или в такте *t* выполняется только логическое условие r_{jl}, а в такте *t* +1 совмещаются операторы A_{in} и A_{jm}.

Заметим, что операторы *S* и *R*, как и операторы A_0 и $A_к$, совмещаться между собой и с другими операторами, как правило, не могут.

3. В каждой из параллельных ветвей имеются логические условия. Если они находятся в разных местах параллельных ветвей ПГСА и совмещение этих ЛУ невозможно, то условия совмещения операторов и логических условий параллельных ветвей в этом случае будут такими же, как и во втором варианте.

Рассмотрим особенности совмещения ЛУ, входящих в разные параллельные ветви. Как и в предыдущем варианте сочетания параллельных ветвей (вариант 2), рассмотрим три случая (ЛУ находятся в начале, середине и конце параллельных ветвей ПГСА):

а) логические условия находятся в начале параллельных ветвей ПГСА (рис. 3.42, *а*).

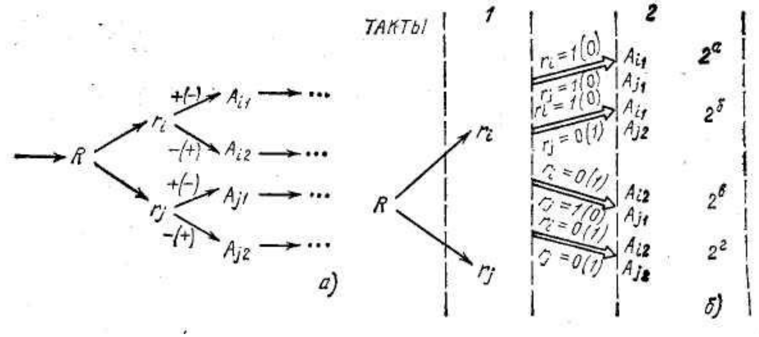

Рис. 3.42.

Если их совместить в одном такте 1 (рис. 3.42, *б*), то при двух ЛУ возможно четыре сочетания значений ЛУ, а следовательно, четыре возможных разновидности такта 2*;* такт 2ª, когда должны одновременно выполняться операторы A_{i1} и A_{j1}*;* такт 2ᵇ — операторы A_{i1} и A_{j2}*;* такт 2ᵛ — операторы A_{i2} и A_{j1}, такт 2ᵍ — операторы A_{i2} и A_{j2}. После такта 2^{ξ} (ξ = а, б, в, г) в последующих тактах совмещаются те операторы, которые являются последователями операторов, совмещенных в такте 2^{ξ}, (ξ =а, б, в, г);

б) в том случае, когда логические условия находятся в середине параллельных ветвей ПГСА, то, как и в случае для варианта 2б, совмещение операторов различных параллельных ветвей происходит аналогично тому, как это было описано в варианте 1а, начиная с этих ЛУ — как в случае для варианта 3а;

в) логические условия находятся в конце ветвей ПГСА. Как и в случае для варианта 2в в одной из альтернативных ветвей в каждой параллельной ветви за r_i, r_j должен следовать хотя бы один оператор (рис. 3.43, *а*).

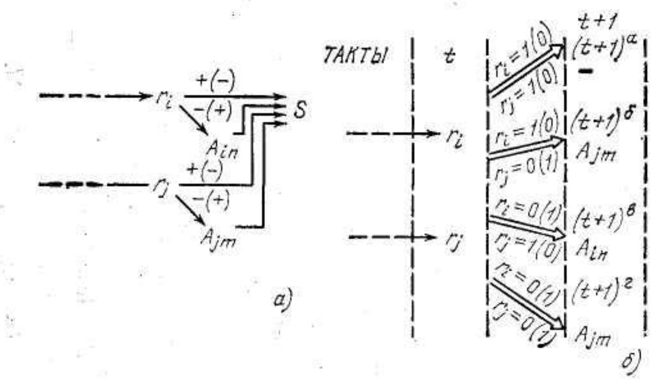

Рис. 3.43.

Тогда очевидно, что при двух ЛУ могут быть четыре сочетания их значений, а поэтому четыре разновидности такта t+1, как это указано на рис. 3.43, *б*.

4. Хотя бы в одной из параллельных ветвей ПГСА имеется цикл. Например, в одной из двух параллельных ветвей имеется цикл, как это показано на рис. 3.44, *а*.

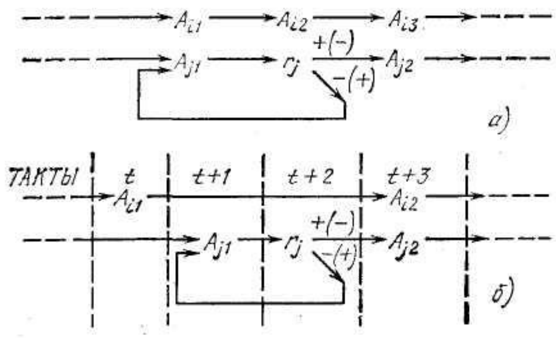

Рис. 3.44.

В связи с тем, что заранее не известно, сколько тактов работы УА займет цикл, операторы и ЛУ, входящие в цикл, не могут быть совмещены с операторами и ЛУ параллельной ветви. Поэтому для членов (операторов, ЛУ) цикла должны быть отведены отдельные такты (рис. 3.44, *б*), число повторений которых, например, в нашем случае зависит от того, когда ЛУ сменит свое значение с 0 (или 1) на 1 (или 0).

Второй принцип реализации консультационного алгоритма рассмотрим на примере.

Пример 1. Пусть консультационный алгоритм задан в виде ПГСА, изображенной на рис. 3.38. Требуется распределить операторы и ЛУ по тактам работы САК при втором принципе реализации консультационного алгоритма. При этом для последовательно выполняемых членов ПГСА, т. е. операторов и ЛУ, входящих в одну и ту же ветвь ПГСА, будем отводить отдельный такт работы САК.

При этих условиях в первых двух тактах последовательно выполняются оператор A_0 и логическое условие r_1 (рис. 3.45,*а*).

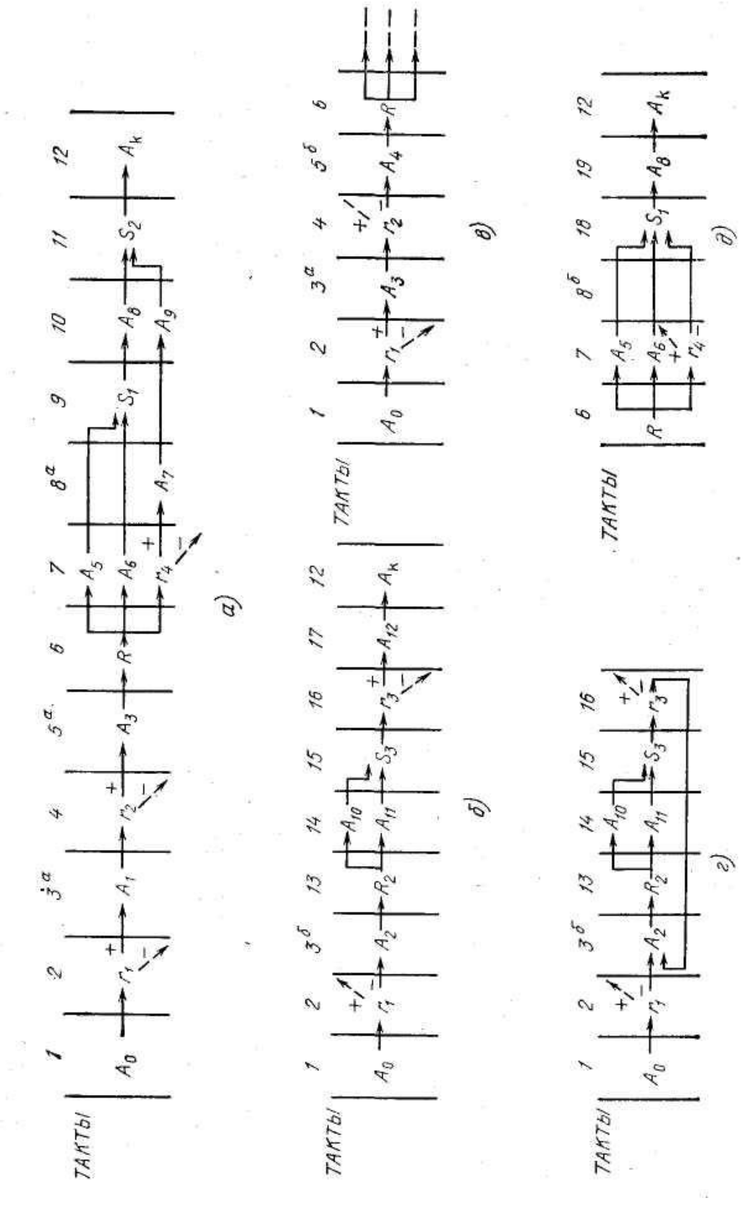

Рис. 3.45.

После ЛУ r_1 происходит разветвление алгоритма: выполняется ветвь ПГСА, начиная с оператора A_1 (такт $3^а$), или ветвь ПГСА, начиная с оператора A_2 (такт. $3^б$, рис. 3.45,*б*). Аналогично алгоритм разветвляется на две альтернативные ветви после ЛУ r_2 и r_3. При этом если r_2=1, то будет выполнен оператор A_3 (такт $5^а$, рис. 3.45,*а*), а при r_2=0 — оператор A_4 (такт $5^б$, рис. 3.45,*в*. При r_3=1 реализуется оператор A_1 (такт 17, рис. 3.45,*б*), а при r_3=0 — оператор A_2 (такт $3^б$, рис. 3.45,*г*).

Таким образом, из-за разветвления алгоритма на альтернативные ветви при проверке логических условий r_1, r_2 и r_3 может быть получено пять вариантов последовательности выполнения операторов и ЛУ консультационного алгоритма. При этом начиная с такта 6 процессы выполнения операторов и ЛУ при r_1 = 1, r_2=1 (рис. 3.45,*а*) и при r_1=1, r_2=0 (рис. 3.45,*б*) совпадают. Поэтому на рис. 3.45,*в* после такта *6* при r_4=1 такты *7*, $8^а$, *9, 10, 11, 12* (рис. 3.45,*а*), а также при r_4 = 0 такты *7*, $8^б$, *18, 19, 12* (рис. 3.45,*д*) не указаны.

В связи с тем что при r_4=0 соединение параллельных подалгоритмов осуществляется оператором S_1, а не оператором S_2, при r_4=0 оператору S_1 отведен новый такт *18,* где он реализует несколько отличные функции от функций такта *9* — соединение не двух, а трех параллельных ветвей консультационного алгоритма.

В том случае, когда вместо операторов в ПГСА используются подалгоритмы, их объединение осуществляется на уровне операторов и логических условий, входящих в эти подалгоритмы по приведенным выше правилам.

Третий принцип. Промежуточным принципом реализации консультационного алгоритма между рассмотренными выше может служить псевдопараллельная реализация в виде *режима разделения времени.*

Режим (система) разделения времени возник в начале шестидесятых годов для организации на одной ЭВМ обслуживания многих пользователей. Система разделения времени дала возможность программисту постоянно контролировать ход решения задачи на ЭВМ и вмешиваться в процесс ее решения без заметного по сравнению с пакетной обработкой снижения использования ЭВМ. В связи с этим режим разделения времени позволил эффективно применять ЭВМ в качестве консультационных машин, работающих в масштабе реального времени, когда создаваемыми ЭВМ задержками при выполнении консультационных процессов можно пренебречь.

Режим разделения времени может использоваться не только для обслуживания многих пользователей на однопроцессорной ЭВМ или

управлять многими блоками объекта управления в масштабе реального времени, но и реализовать параллельный консультационный алгоритм.
При реализации параллельного консультационного алгоритма в режиме разделения времени процесс выполнения операторов и ЛУ параллельных ветвей ПГСА разбивается на *периоды.* При этом в каждом периоде для каждой параллельной ветви отводится один *квант времени,* в течение которого САК реализует команды, соответствующие операторам или ЛУ, только одной ветви ПГСА. Так, вначале в первом кванте времени выполняются команды первой параллельной ветви. Затем во втором кванте времени — команды первой ветви прерываются и начинаются команды второй параллельной ветви и т. д. до тех пор, пока не закончится данный период. Во втором периоде данный процесс повторится, т. е. в первом кванте времени возобновится выполнение команд первой параллельной ветви и т. д.

В простейшем случае, когда в параллельных ветвях имеется по одной команде (оператору или ЛУ), то период соответствует такту работы САК, а в качестве кванта времени может рассматриваться *микротакт.* При этом режим разделения времени вырождается в обычный процесс последовательного (по-тактного) выполнения команд параллельных ветвей консультационного алгоритма.
Принцип псевдопараллелыюй реализации параллельного консультационного алгоритма в режиме разделения времени для такого простейшего случая рассмотрим на примере.

Пример 2. Пусть задан параллельный консультационный алгоритм, представленный на рис. 3.38 в виде ПГСА. Временная диаграмма работы программной САК, реализующей данный консультационный алгоритм, представлена на рис. 3.46.

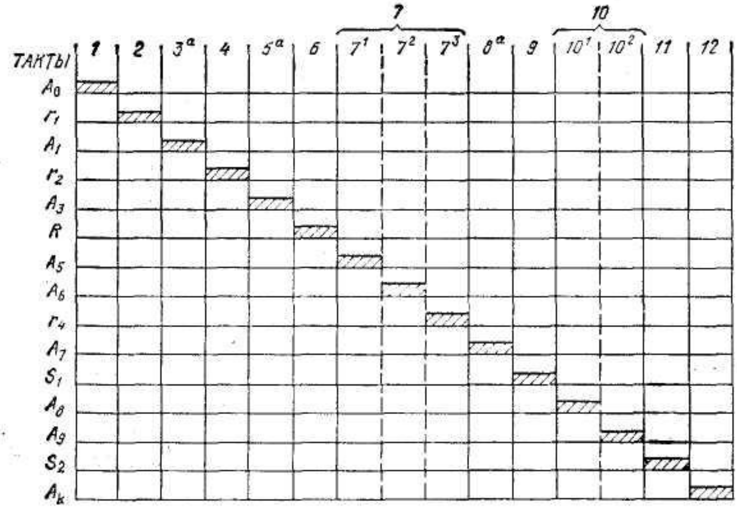

Рис. 3.46.

При этом, как и ранее, предполагается, что в одном такте работы САК выполняется один оператор или одно ЛУ одной ветви ПГСА, а различные последовательности операторов при различных наборах значений логических условий r_1, r_2, r_3 и r_4 представим, как и в примере 4.1, различными временными диаграммами. Для примера на рис. 3.46 изображена временная диаграмма выполнения операторов и ЛУ для одной ветви ПГСА при r_1=1, r_2=l и r_4=1 соответствующей ветви, изображенной на рис. 3.45,*а*.

Из рис. 3.46 видно, что такт *7*, в котором параллельно должны быть выполнены операторы A_5, A_6 и логические условия r_4, разделен на три кванта времени (микротакта), каждый из которых отводится для одной из трех параллельных ветвей. Аналогично такт *10* разделен на два микротакта — в одном (микротакте 10^1) для оператора A_8, а в другом (микротакте 10^2) — для оператора A_9, принадлежащего второй параллельной ветви.

В том случае, когда в такте нет операторов или ЛУ, принадлежащих параллельным ветвям, его деление на микротакты не происходит и вся длительность такта отводится для соответствующего члена ПГСА.

Наиболее рельефно эффективность системы разделения времени при реализации параллельного консультационного алгоритма проявляется в том случае, когда в параллельных ветвях имеется достаточно большое число операторов и ЛУ.

Пример 3. Пусть в режиме разделения времени необходимо организовать выполнение двух подалгоритмов A_1 и A_2 консультационного алгоритма, включающих соответственно 5 и 11 членов (команд) ПГСА K_i^1, i=1, ..., 5; K_i^2, j=1, ..., 11. При этом для подалгоритма A_1 отводятся нечетные кванты времени в периоде, а для подалгоритма A_2 — четные.

Процесс выполнения команд этих параллельных подалгоритмов в режиме разделения времени виден из рис. 3.47.

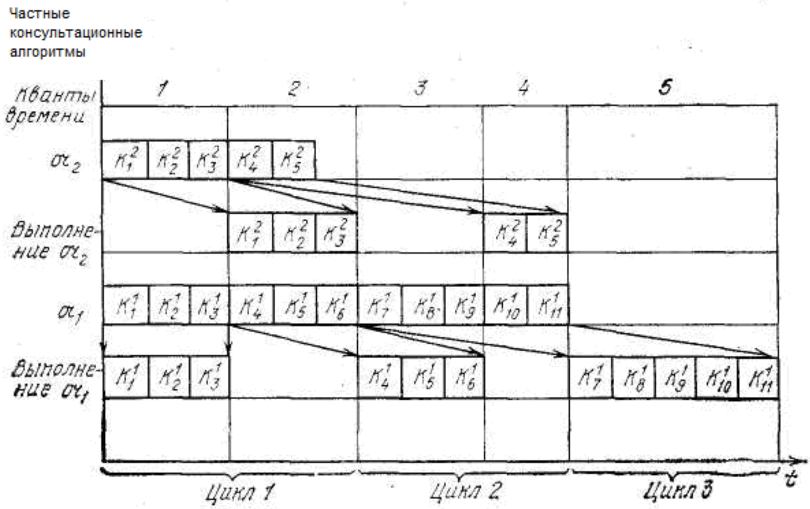

Рис. 3.47.

При этом легко заметить, что когда и в A_1 и в A_2 имеются невыполненные команды, в периоде по очереди отводятся кванты времени и для A_1 (кванты *1, 3)*, и для A_2 (кванты *2, 4).* Когда же заканчиваются команды подалгоритма A_2, деление периода на кванты уже не осуществляется — САК начинает последовательно выполнять команды только подалгоритма A_1. При этом квант времени совпадает с периодом, а сам период не ограничен во времени. Фактически все время работы САК отводится для подалгоритма A_1 до тех пор, пока он не прекратится или не встретится новый подалгоритм, который должен выполняться с ним параллельно.

Четвертый принцип. В заключение следует отметить, что система разделения времени в значительной степени опирается на использование *режима многопрограммного управления,* которое реализуется в децентрализованной САК с активными функциональными блоками (ФБ), среди которых могут встречаться *медленно действующие ФБ* (МФБ).

При применении режима многопрограммного управления, которое может рассматриваться в качестве четвертого принципа реализации параллельного консультационного алгоритма, будем говорить, что САК является *многопрограммной.*

В связи с тем что многопрограммное управление получает широкое распространение в современных САК, рассмотрим его более подробно в следующем разделе.

3.4.2. Многопрограммное управление

Программный принцип построения САК, как уже отмечалось, наиболее эффективен в том случае, когда в САК может быть реализовано несколько различных консультационных алгоритмов. Обычно выбирается набор однотипных консультационных задач, для решения которых требуются одинаковые ФБ, и для каждой задачи составляется своя программа (алгоритм). Все эти программы хранятся в центральном блоке управоения (ЦБУ) САК (см. рис. 3.48) и в зависимости от поступившего на ЦБУ сигнала *R* (это в частном случае может быть код операции) в САК выполняется определенный алгоритм.

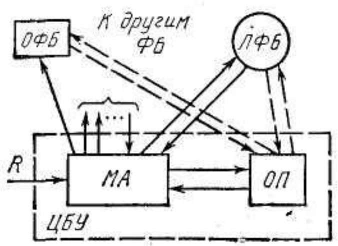

Рис. 3.48.

САК может перейти к другому алгоритму лишь после того, как будет закончен предыдущий. При этом ЦБУ непрерывно следит за работой функциональных блоков, принимающих участие в решении данной задачи. Подав включающий сигнал на ФБ, ЦБУ независимо от длительности работы этого блока находится в состоянии ожидания ответного сигнала от этого блока (при асинхронном режиме) или от тактового генератора (при синхронном режиме). Таким образом, хотя САК, как правило, может решать несколько различных задач или управлять несколькими процессами, одновременно САК решает только одну задачу, т. е. работает только по одному из алгоритмов, хранящихся в ЦБУ. Такой принцип можно было бы назвать однопрограммным.

Однако характерной особенностью современных САК является наличие в их составе устройств, имеющих разное быстродействие. Часть этих устройств (например, ЦБУ) может обладать весьма высоким быстродействием. Быстродействие же другой части устройств в значительной степени ограничено связанными с их работой процессами механических перемещений в САК. Так, в САК медленно действующими являются операции ввода и вывода информации.

Во всех этих случаях при сохранении однопрограммной работы САК повышение быстродействия отдельных ее узлов (например, ЦБУ) не приводит к соответствующему увеличению производительности САК из-за простаивания этих быстродействующих узлов в ожидании срабатывания более медленных.

Одним из средств повышения производительности САК является совмещение во времени отдельных шагов консультационного алгоритма. При этом из ЦБУ подается одновременно несколько сигналов на включение параллельно работающих различных ФБ САК, что приводит к сокращению общего времени выполнения алгоритма. Но и в этом случае ЦБУ управляет одновременно выполнением только одного консультационного процесса и каждый ФБ САК все время находится под контролем ЦБУ. При этом даже при асинхронном режиме работы САК, обеспечивающем наибольшее быстродействие, ЦБУ непроизводительно простаивает в ожидании ответа об окончании работы медленно действующего функционального блока (МФБ).

Между тем для работы активных ФБ не требуется постоянное воздействие со стороны ЦБУ — достаточно подать лишь кратковременный сигнал на начало работы ФБ. Производительность ЦБУ можно значительно повысить, если на время работы МФБ отключить ЦБУ от управления начатым процессом и использовать его для решения другой задачи. При этом такие МФБ должны быть в значительной степени автономными, чтобы они могли выполнить достаточно сложную обработку информации, не получая от ЦБУ дополнительных данных и сигналов. В ряде случаев такие МФБ могут быть настолько сложными, что сами оказываются САК. Таким образом, возникает некоторая иерархия САК. Однако для САК верхнего уровня все САК нижнего уровня являются просто ФБ, выполняющими определенные акты алгоритма и находящимися в подчинении его ЦБУ.

Подав сигнал на включение МФБ, ЦБУ переходит к решению другой задачи, вырабатывая последовательность управляющих сигналов в соответствии с алгоритмом ее решения, доходит до сигнала, включающего МФБ в этой программе, переходит к решению третьей задачи и т. д. (рис. 3.49).

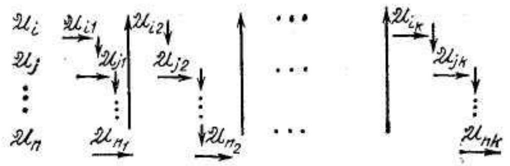

Рис. 3.49.

Таким образом создается возможность выполнять параллельно несколько ветвей одной программы, или выполнять параллельно несколько различных программ, или решать одну и ту же задачу, но для различных исходных данных одновременно. Такой принцип работы САК находит применение во многих САК и полунил название многопрограммного управления.

Очевидно, для того чтобы вернуться к решению прерванной задачи, ЦБУ должен иметь информацию об этапе, на котором она была прервана, об окончании работы того МФБ, который был включен, и др. Вся эта информация хранится в ячейках оперативной памяти.

Выбор емкости памяти и способ ее распределения могут быть выполнены по-разному и представляют самостоятельную задачу, близкую к задачам, решаемым в теории массового обслуживания. Можно считать, что имеется какое-то число *N* обслуживаемых объектов (это могут быть абоненты, исходные данные для решения задачи и т. д.), от которых в САК поступают заявки на определенный вид обслуживания. Начав обслуживание по определенному алгоритму и дойдя до выполнения первого медленно действующего оператора, ЦБУ переходит к обслуживанию другого объекта и т. д. При этом за каждым из объектов можно закрепить свою ячейку памяти, в отдельных разрядах которой будут отмечаться номер алгоритма, по которому в данный момент обслуживается этот объект, этап алгоритма, выполняемый в данный момент, а также номера промежуточных устройств, занятых на этом этапе обслуживания.

Однако если число обслуживаемых объектов велико, то при таком способе распределения памяти САК объем ее может оказаться очень большим. Не от всех объектов могут поступать одновременно заявки на их обслуживание и не все поступившие заявки САК может обслуживать одновременно. Поэтому число ячеек памяти целесообразно выбирать не по общему числу объектов, а по числу объектов, которое САК может обслуживать одновременно. Тогда в каждой ячейке памяти нужно будет записывать еще и номер объекта, за которым она в данный момент закреплена.

Возможен и другой подход. Поскольку САК может обслуживать один и тот же объект в соответствии с различными алгоритмами, то можно для каждого из алгоритмов, хранящихся в ЦБУ, отвести свою ячейку.

Вообще говоря, число одновременно выполняемых операций не может превышать число функциональных блоков в САК, а число алгоритмов, обслуживание которых начал ЦБУ, может быть значительно большим и ограничивается только емкостью памяти. Однако с целью экономии емкости оперативной памяти (ОП) целесообразно число ячеек выбирать не по общему числу алгоритмов, а по числу алгоритмов, в соответствии с которыми одновременно может вестись обслуживание поступающих заявок. Тогда в отдельные разряды ячеек памяти будут записываться номер заявки, номер алгоритма, по которому она обслуживается, выполняемый этап алгоритма и необходимые номера промежуточных устройств, занятых в этом процессе. Так как заявки могут одновременно обслуживаться по одному и тому же алгоритму, различные ячейки памяти могут соответствовать одному алгоритму. В зависимости от решаемых задач и требований, предъявляемых к САК, может быть выбран тот или иной способ распределения памяти.

Отказавшись от жесткого закрепления ячеек памяти за номерами поступающих заявок или за номерами алгоритмов, приходим к общему случаю, когда каждая свободная ячейка памяти может быть занята любой заявкой и отведена для любого из алгоритмов. При этом хотя и сократится число ячеек по сравнению с описанными крайними случаями, разрядность каждой ячейки увеличится, так как нужно будет запоминать номер заявки, которой она занята, и номер алгоритма, по которому эта заявка в данный момент обслуживается. Число ячеек памяти, которое необходимо при этом выбрать, будет определяться числом заявок и алгоритмов их обслуживания, а также свойствами САК.

Наряду с выбором емкости оперативной памяти ЦБУ возникает задача выбора способа организации его работы при многопрограммном управлении. Начав обслуживание одной заявки по требуемому алгоритму и дойдя до первой медленно действующей операции этого алгоритма, ЦБУ включает соответствующий МФБ и переходит к обслуживанию другой заявки. Возникает естественный вопрос—через какое время ЦБУ должен продолжить обслуживание первой заявки. Это время будет зависеть от того, по какому принципу (дисциплине) организована работа ЦБУ. Рассмотрим два из возможных принципов.

При первой дисциплине обслуживания ЦБУ последовательно просматривает состояние всех ячеек оперативной памяти и выполняет для каждой заявки все операции алгоритма (по которому она обслуживается), не требующие включения МФБ. После включения МФБ ЦБУ переходит к считыванию информации со следующей ячейки ОП и т. д. Таким образом, ЦБУ может продолжить обслуживание начатой заявки лишь после того, как будут просмотрены все остальные ячейки памяти и выполнены определенные операции по обслуживанию заявок, для которых они отведены. Если число ячеек ОП будет очень большим, то ЦБУ сможет начать обслуживание большого числа заявок, однако время выполнения каждой заявки увеличится по сравнению с однопрограммным методом управления, так как МФБ успеют закончить свою работу намного раньше, чем ЦБУ обратится вновь к данной ячейке памяти. Если число ячеек слишком мало, то ЦБУ возьмет на обслуживание небольшое число заявок и будет столь часто обращаться к каждой ячейке памяти, что МФБ не будут успевать заканчивать работу за время одного цикла опроса ЦБУ ячеек памяти. Таким образом, при выборе емкости ОП необходимо учитывать быстродействие ЦБУ, конкретные алгоритмы, которые им реализуются, число и длительность работы МФБ.

Если ЦБУ работает по второй дисциплине обслуживания, он продолжает обслуживание начатой заявки сразу же после поступления сигнала о том, что включенный МФБ закончил свою работу. Для этого ЦБУ, обратившись к очередной ячейке памяти, выполнив все быстродействующие операции и включив нужный МФБ, прежде чем перейти к опросу следующей ячейки памяти, должен опросить все включенные МФБ. Если какой-либо из них закончил работу, ЦБУ продолжает обслуживание той заявки, выполнением которой был занят этот МФБ. При этом для каждого МФБ в специальном регистре должен храниться номер той ячейки памяти, в которой записан номер этого МФБ, чтобы ЦБУ мог продолжить обслуживание заявки по нужному алгоритму и записать в данную ячейку номер следующего включенного МФБ.

Таким образом, каждый раз номер следующей команды, вырабатываемой ЦБУ, будет зависеть от того, какой МФБ подал сигнал об окончании своей работы. Разумеется, если таких МФБ несколько, должна быть организована определенная очередность их опроса.

При такой дисциплине обслуживания ЦБУ может приступить к обслуживанию новой заявки при условии, если ни в одной из уже обслуживаемых заявок не закончилось выполнение длительных операций, т. е. нет сигнала ни от одного из включенных МФБ. При этом число одновременно обслуживаемых заявок может уменьшиться, но время выполнения отдельных заявок может сократиться по сравнению с первой дисциплиной обслуживания. Такой принцип работы ЦБУ нашел применение в некоторых САК.

Следует отметить, что как при первой, так и при второй дисциплинах обслуживания, может возникнуть ситуация, когда при обслуживании i -й заявки потребуется включить j-й МФБ, который уже занят для выполнения другой заявки и еще не закончил свою работу. При этом i-я заявка либо может быть поставлена на ожидание, либо источнику этой заявки может быть дан отказ.

Ранее отмечалось, что при многопрограммном управлении параллельно могут выполняться не только различные алгоритмы, но и разные ветви одного и того же параллельного консультационного алгоритма. Очевидно, при этом возникает задача выявления таких ветвей алгоритма, которые могут выполняться одновременно (распараллеливание алгоритма).

Принцип многопрограммного управления может быть использован не только при наличии в выполняемых алгоритмах длительных операций, но и в тех случаях, когда заявки, которые обслуживает САК, неравноценны по своей значимости и задается некоторая система приоритетов. Тогда при поступлении заявки высокой категории ЦБУ переходит к ее обслуживанию и прерывает обслуживание заявки более низкой категории, записав в ячейку памяти этап, на котором она была прервана. Обслуживание этой заявки будет продолжено после того, как будут выполнены все заявки более высоких категорий. При этом ЦБУ будет работать так же, как и при наличии МФБ, только обслуживание заявки здесь будет прерываться не естественно, в момент включения МФБ, а принудительно, в любой момент, когда поступит заявка более высокой категории.

3.4.3. Преобразование консультационного алгоритма при его реализации в многопрограммной САК

В связи с тем что в многопрограммной САК кроме ЦБУ имеются активные МФБ, в которых параллельно с ЦБУ могут выполняться отдельные или группы операций алгоритма, требуется так преобразовать консультационный алгоритм, чтобы обеспечить ЦБУ в соответствии с управляющей информацией, хранящейся в его ОП, возможность последовательного выполнения различных консультационных алгоритмов или подалгоритмов параллельного консультационного алгоритма.

Консультационный алгоритм представим в виде ЛСА. В дальнейшем нам удобно будет считать, что в ЛСА нет членов, которые могут выполняться одновременно. Очевидно, этого всегда можно добиться, объединив одновременно выполняющие члены в один оператор или в одно логическое условие.

Тогда при обычном программном принципе работы САК, если началось выполнение i-го члена ЛСА (т. е. осуществлена подача сигнала на включение i-го ФБ), нельзя начать выполнение (i+l)-го члена ЛСА до тех пор, пока не будет получена информация об окончании работы его предшественника. При этом управляющий сигнал, вырабатываемый ЦБУ и включающий $ФБ_i$, и осведомительный сигнал, воздействующий на ЦБУ от $ФБ_i$, описываются одним членом ЛСА (A_i или p_i).

Если говорим, что выполнен оператор A_i, то, значит, от ЦБУ на блок $ОФБ_i$ поступил сигнал z_{A_i}, этот блок закончил свою работу и от него на ЦБУ поступил сигнал A_i (при синхронном режиме этот сигнал заменяется сигналом от тактового генератора).

При многопрограммном управлении ЦБУ, подав включающий сигнал на МФБ, отключается от него и осведомительный сигнал получает уже от другого ФБ, принадлежащего даже другому алгоритму. При этом управляющий и осведомительный сигналы МФБ, которые обычно описываются одним членом ЛСА, теперь должны быть разделены. И в ЛСА вместо ее членов, соответствующих МФБ, должны быть введены новые члены, характеризующие отдельно управляющие и осведомительные сигналы. Предварительно в каждой ЛСА, реализуемой в ЦБУ, должны быть выделены члены, соответствующие МФБ. Разделение операций на быстрые и медленные является условным, однако во многих практических задачах относительно некоторых операций можно сразу сказать, что для их выполнения потребуется гораздо больше времени, чем для выполнения других операций.

Рассмотрим пример ЛСА, отметив ее члены, соответствующие МФБ, индексом «м»:

$$\mathfrak{A}_i = \downarrow^3 p_1 \uparrow^1 A_1 A_{2м} \downarrow^1 p_{2м} \uparrow^2 A_3 \downarrow^2 A_{4м} p_3 \uparrow^3.$$

Управляющий сигнал, поступающий из ЦБУ на i-й *операторный* МФБ (ОМФБ), будем обозначать C_i, а сигнал на включение i-го *логического* МФБ (ЛМФБ) —S_i. Сигналы C_i и S_i, включают соответствующие МФБ и записывают в определенные разряды ячеек памяти сигналы о том, что эти блоки включены.

Сигнал о выполнении оператора $A_{iм}$, поступающий от i-го ОМФБ в ЦБУ, будем обозначать K_i. Заметим, что i-й логический блок ЛМФБ выдает два сигнала: G_i, если в результате проверки выяснено, что $p_{iM}=1$, и H_i, если $p_{iM}=0$.

Сигналы K_i, G_i и H_i не могут воздействовать непосредственно на ЦБУ, так как в момент их появления он может работать в соответствии с другим алгоритмом. Эти сигналы записывают в определенные разряды ячейки памяти информацию об окончании работы МФБ. Для того чтобы эта информация могла быть использована ЦБУ, необходимо осуществить проверку этих разрядов ячейки памяти. Поэтому в ЛСА кроме операторов C_i и S_i вводятся логические условия g_1, g_2, ..., g_k, где $k=l+2m$, если l — число ОМФБ и m — число ЛМФБ. Для сигналов от ЛМФБ отводятся два разряда ячейки памяти, при этом можно условиться, что для сигнала G_i (p_i=1) отводится всегда младший из этих двух разрядов. Тогда для работы по выписанной ранее ЛСА A_i ячейка памяти будет иметь вид, показанный на рис. 3.50,*а*.

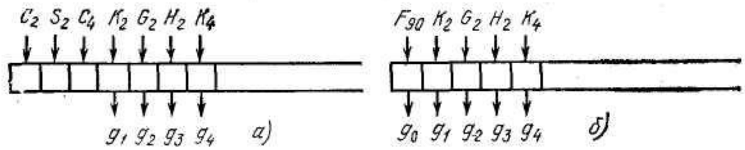

Рис. 3.50.

Однако эта информация является избыточной. Действительно, если включен второй ОМФБ (т. е. подан сигнал C_2), то может прийти только сигнал K_2, после чего логическое условие g_1 примет истинное значение. И, наоборот, если g_1=1, это значит, что был включен второй ОМФБ. Для того чтобы значения переменных g_1, ..., g_k действительно характеризовали этап, на котором было прервано выполнение алгоритма, необходимо, чтобы в каждый момент не более чем одна из этих переменных принимала единичное значение. Поэтому каждый раз после считывания единичного значения переменной g_i разряд ячейки, в который она была записана, должен быть очищен. При этом сигналы о включении МФБ можно не записывать в ячейку памяти. Однако тогда потребуется еще один дополнительный сигнал, свидетельствующий о том, что данная ячейка памяти уже занята и начато выполнение определенного алгоритма. В противном случае, когда в ячейке памяти значения всех переменных g_1, ..., g_k равны нулю, нельзя будет определить, начата ли работа по данному алгоритму, но включенный МФБ еще не закончил работу, либо данная ячейка памяти еще свободна. Для формирования такого сигнала введем в ЛСА оператор

F_{g_0}, который будет осуществлять запись сигнала о занятии данной ячейки в определенный ее разряд.

Содержимое этого разряда будем определять путем проверки логического условия g_0. Если g_0=l, значит, ячейка занята и начато обслуживание поступившей заявки по требуемому алгоритму. Оператор F_{g_0} может выполняться вслед за первым членом ЛСА, соответствующим МФБ. Очевидно, после того как алгоритм полностью выполнен, все разряды ячейки памяти должны быть освобождены. Для этого в ЛСА введен оператор $F^*_{g_0}$.

Тогда ячейка памяти для алгоритма A_i принимает вид ячейки, изображенной на рис. 3.50,*б*, а ЛСА запишем в виде

$$\mathfrak{A}_i = g_0 \uparrow^1 \downarrow^6 \bar{g}_1 \uparrow^2 \bar{g}_2 \uparrow^3 \bar{g}_3 \uparrow^4 \bar{g}_4 \uparrow^5 \omega \uparrow^6 \downarrow^1 p_1 \uparrow^2 A_1 C_2$$
$$F_{g_0} \omega \uparrow^6 \downarrow^2 S_2 \omega \uparrow^6 \downarrow^3 A_3 \downarrow^4 C_4 \omega \uparrow^6 \downarrow^5 p_3 \uparrow^1 F^*_{g_0}.$$

Конечно, в ячейке памяти могут быть и другие разряды, необходимые для записи некоторых дополнительных данных. Число таких разрядов зависит от конкретно выполняемого алгоритма и здесь не указывается.

Такая запись означает, что ЦБУ по-прежнему одновременно обслуживает только одну заявку, так как после включения МФБ внутреннее состояние МА не меняется и он находится в состоянии ожидания. Здесь только разделены сигналы включения МФБ и сигналы об окончании их работы. При многопрограммном управлении после выполнения оператора, включающего МФБ, ЦБУ переходит к обслуживанию следующей заявки, а выполнение начатого алгоритма заканчивается, т. е.

$$\mathfrak{A}_i = g_0 \uparrow^1 \bar{g}_1 \uparrow^2 \bar{g}_2 \uparrow^3 g_3 \uparrow^4 \bar{g}_4 \uparrow^5 \omega \uparrow^6 \downarrow^1 p_1 \uparrow^2 A_1$$
$$C_2 F_{g_0} \omega \uparrow^6 \downarrow^2 S_2 \omega \uparrow^6 \downarrow^3 A_3 \downarrow^4 C_4 \omega \uparrow^6 \downarrow^5 p_3 \uparrow^1 F^*_{g_0} \downarrow^6.$$

Рассмотрим теперь работу ЦБУ при многопрограммном управлении. Пусть имеется N источников, от которых на САК поступают заявки на обслуживание. Каждая заявка может быть обслужена по одному из следующих трех алгоритмов:

$$\mathfrak{A}_1 = \downarrow^3 p_1 \uparrow^1 A_1 A_{2\text{м}} \downarrow^1 p_{2\text{м}} \uparrow^2 A_3 \downarrow^2 A_{4\text{м}} p_3 \uparrow^3;$$
$$\mathfrak{A}_2 = A_6 \downarrow^2 A_{5\text{м}} p_1 \uparrow^1 A_3 p_{2\text{м}} \uparrow^2 \downarrow^1 A_1;$$
$$\mathfrak{A}_3 = A_6 A_{2\text{м}} p_3 \uparrow^1 p_{2\text{м}} \uparrow^2 A_3 \downarrow^2 A_{4\text{м}} \downarrow^1.$$

Заметим, что эти алгоритмы можно рассматривать как частные алгоритмы одного параллельного консультационного алгоритма.
Для обеспечения одновременного выполнения нескольких заявок выберем L ячеек памяти и организуем работу ЦБУ следующим образом. Из ЦБУ поступают сигналы на последовательный опрос ячеек временной памяти. Для этого вводится оператор F_j, формирующий номер ячейки (j=1, 2, ... L). После того как найдена первая свободная ячейка (g_0=0), выдается сигнал на определение наличия заявок. Пусть имеется специальный блок F_1, который выдает номер одного из вызывающих в данный момент источников и номер алгоритма, по которому эта заявка должна быть обслужена. Все эти данные одновременно с сигналом $F^*_{g_0}$ записываются в выбранную ячейку памяти. Затем начинается выполнение требуемого алгоритма до первого его члена, соответствующего МФБ. Подав сигнал на включение этого МФБ, ЦБУ выдает команду F_i и переходит к считыванию информации со следующей ячейки памяти. Если ячейка памяти уже занята (g_0=l), то анализируется состояние разрядов, в которых записываются значения переменных g_1, ..., g_k. В зависимости от этих значений либо выполняются определенные акты алгоритма, номер которого записан в данной ячейке, либо вновь выполняется команда F_j и начинается опрос следующей ячейки памяти.

Для каждого МФБ введем разряды ячейки памяти. Поскольку в различных алгоритмах имеются члены, соответствующие разным МФБ, некоторые разряды будут не заполнены в зависимости от того, для выполнения какого алгоритма занята в данный момент эта ячейка. (Возможно и другое распределение этих разрядов ячейки памяти, о котором скажем ниже). Для определения номера алгоритма, который записан в данной ячейке, введем логические условия r_1, ..., r_n. Для нашего примера, когда имеются три различных алгоритма, достаточно двух логических условий r_1 и r_2, т. е. для записи номера алгоритма будем использовать двоичный код. Пусть $R_1=r_1r_2$; $R_2=\overline{r_1}\ \overline{r_2}$; $R_3=r_1\overline{r_2}$.

На рис. 3.51 приведена структура ячеек памяти, занятых обслуживанием алгоритмов A_1, A_2 и A_3.

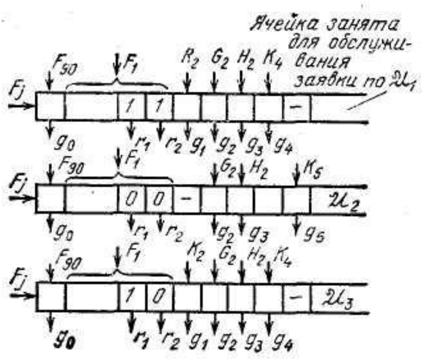

Рис. 3.51.

Ранее отмечали, что возможна такая ситуация, когда для обслуживания вновь поступившей заявки потребуется включить такой ФБ, который уже занят обслуживанием другой заявки. Поэтому в общем случае перед включением ФБ нужно вводить проверку его свободности и в зависимости от результатов этой проверки либо включить ФБ, если он свободен, либо ставить заявку на ожидание или давать отказ источнику заявки, если ФБ занят. Однако предположим, что среди одинаковых ФБ всегда найдется хотя бы один свободный, и не будем вводить специальных ЛУ для проверки состояния ФБ.

Тогда работу ЦБУ при многопрограммном управлении можно описать следующей ЛСА:

$$\mathfrak{A} = \downarrow^{4} F_j \overline{g}_0 \uparrow^{1} F_1 F_{g_0} r_1 \uparrow^{2} r_2 \uparrow^{3} \downarrow^{11} p_1 \uparrow^{7} A_1 C_2 \omega \uparrow^{4} \downarrow^{3} A_6$$

$$C_2 \omega \uparrow^{4} \downarrow^{2} A_6 \downarrow^{17} C_5 \omega \uparrow^{4} \downarrow^{1} r_1 \uparrow^{5} r_2 \uparrow^{6} \overline{g}_1 \uparrow^{7} \overline{g}_2 \uparrow^{8} \overline{g}_3 \uparrow^{9} \overline{g}_4 \uparrow^{10}$$

$$\omega \uparrow^{4} S_2 \omega \uparrow^{4} \downarrow^{8} A_3 \downarrow^{9} C_4 \omega \uparrow^{4} \downarrow^{10} p_3 \uparrow^{11} F^{\bullet}_{g_0}$$

$$\omega \uparrow^{4} \downarrow^{6} \overline{g}_1 \uparrow^{12} \overline{g}_2 \uparrow^{13} \overline{g}_3 \uparrow^{14} \overline{g}_4 \uparrow^{15} \omega \uparrow^{4} \downarrow^{12} p_3 \uparrow^{15} S_2 \omega \uparrow^{4} \downarrow^{13}$$

$$A_3 \downarrow^{14} C_4 \omega \uparrow^{4} \downarrow^{15} F^{*}_{g_0} \omega \uparrow^{4} \downarrow^{5} \overline{g}_2 \uparrow^{16} \overline{g}_3 \uparrow^{17} g_5 \uparrow^{4} p_1 \uparrow^{16} A_3 S_2$$

$$\omega \uparrow^{4} \downarrow^{16} A_1 F^{\bullet}_{g_0} \omega \uparrow^{4}.$$

Как видно, в этой ЛСА много повторяющихся операторов и логических условий, поэтому ее целесообразно упростить, объединив одинаковые члены ЛСА, принадлежащие различным ЛСА. Метод объединения ЛСА изложен в 3.3.3.

Таким образом, все особенности, связанные с использованием многопрограммного принципа управления, проявляются в основном на этапе составления общего консультационного алгоритма. После выбора режима работы ЦБУ и способа распределения памяти в этот алгоритм приходится вводить новые операторы и ЛУ, которых не было ни в одном из алгоритмов обслуживания заявок.

3.4.4. Программно-аппаратурная реализация консультационного алгоритма

Консультационный алгоритм САК представляет собой *алгоритмическое описание.* Если по заданному консультационному алгоритму построена структурная схема САК в том или ином элементном базисе, говорят, что алгоритмическое описание переведено в *структурное* описание, представляющее собой структурную модель заданного консультационного алгоритма, т. е. алгоритма функционирования САК.

Альтернативой *структурному моделированию* может служить *программное моделирование,* при котором консультационный алгоритм представляют в виде совокупности (множеств) *команд,* каждая из которых определяет операцию над множеством исходных или промежуточных данных (операндов) и номер следующей команды.
В качестве операнд могут рассматриваться как отдельные одно- и многоразрядные числа, хранящиеся в запоминающих устройствах (ЗУ), так и наборы значений входных сигналов (в частности, одного сигнала), которые также могут быть предварительно записаны в ЗУ в виде многоразрядных чисел.
Операции могут быть *одноместными,* т. е. выполняемыми над одним операндом, и *многоместными* (в основном двухместными), выполняемыми над несколькими операндами.

Совокупность команд, однозначно описывающая заданный консультационный алгоритм, называют *программой,* а представление консультационного алгоритма в виде программы — *программным моделированием алгоритма*. Для хранения программы может быть использовано ЗУ.

Следует заметить, что структурой САК, аппаратурно реализующей его алгоритм функционирования, обеспечивается активное выполнение последнего в соответствии с воздействующими на САК входными сигналами. В отличие от такой аппаратурной реализации при структурном моделировании консультационного алгоритма программа, реализующая тот же консультационный алгоритм при программном моделировании, является пассивной, т. е. программа является некоторой записью консультационного алгоритма. Для чтения этой записи и выполнения команд программы при программном моделировании используется универсальное дискретное устройство —

процессор. Процессор обеспечивает выполнение любой, но в один и тот же момент одной операции, имеющейся в программе. При этом заметим, что в связи с успехами микроэлектроники широкое распространение получили *микропроцессоры*.

Таким образом, вместо структурной схемы устройства, аппаратно реализующего весь консультационный алгоритм при его структурном моделировании, при программном моделировании процессор (или микропроцессор), представляющий собой универсальное устройство, настраиваемое на аппаратурную реализацию одновременно только одной операции, обеспечивает выполнение всего консультационного алгоритма путем последовательного выполнения четко разграниченных отдельных операций (актов алгоритма), определяемых программой, хранящейся в ЗУ.

Очевидно, в САК при программном моделировании могут быть реализованы только те консультационные алгоритмы, которые удается представить в виде программы, команды которой содержат операции, реализуемые в процессоре (микропроцессоре) данной САК. Однако на практике число выполняемых операций в процессоре или микропроцессоре имеется такое, что с их использованием можно описать любой консультационный алгоритм. Такой набор операций называют *функционально полным.* Вместе с тем функционально полные наборы могут заметно отличаться друг от друга составом операций. При этом один набор позволяет описать один класс алгоритмов более компактно, а другой набор — иной класс алгоритмов. В этом случае говорят, что такие наборы операций являются проблемно-ориентированными, т. е. специализированными для описания для консультационных алгоритмов того или иного класса.

В САК широкое распространение получили *многопроцессорные* и *многомикропроцессорные системы.* Легко понять, что в таких системах одновременно в разных процессорах (микропроцессорах) могут выполняться несколько (по числу процессоров) команд. Поэтому такие системы могут быть эффективно использованы для одновременного выполнения в САК нескольких программ, соответствующих или нескольким консультационным алгоритмам, или нескольким параллельным ветвям одного параллельного консультационного алгоритма.

Структурное (аппаратурное) и программное моделирование являются крайними принципами реализации консультационного алгоритма. Среди промежуточных между ними принципов реали зации консультационного алгоритма отметим два наиболее интересных: реализация структуры САК в базисе однородных сред и микропрограммный принцип.

На практике значительно большее распространение получил микропрограммный принцип реализации консультационного алгоритма. При этом следует отметить, что в отличие от программного управления при микропрограммном управлении вместо процессора используется некоторый *блок микропрограммного управления* (БМУ). В отличие от процессора он обрабатывает не считанные из ЗУ операнды, а снимаемые с консультируемых объектов сигналы (параметры), на основе которых вырабатывает сигналы для формирования рекомендаций теми или иными консультационными процессами. Такие консультационные сигналы называются *микрооперациями,* а их совокупность, выполняемая за один такт работы САК, — *микрокомандами.* Совокупность микрокоманд, обеспечивающая реализацию консультационного алгоритма, называется *микропрограммой.*

Каждая микрокоманда, включающая набор микроопераций и адрес следующей микрокоманды, может, как и команда, храниться в ЗУ (ЗУ микропрограмм) или реализоваться аппаратурно-структурной схемой в том или ином элементном базисе. Заметим при этом, что процессор САК может быть построен так, что выполняемые в нем команды реализуются или аппаратурно, или в блоке микропрограммного управления в соответствии с микропрограммами, определяющими элементарные действия процессора при выполнении той или иной команды. В последнем случае говорят, что САК имеет процессор с микропрограммным управлением. Наиболее часто микропрограммное управление используется в микропроцессорах.

Таким образом, программная и микропрограммная реализации консультационного алгоритма предполагают наличие программы (или микропрограммы) и аппаратурных средств в виде процессора или микропроцессора, поэтому фактически такая реализация является программно-аппаратурной. Однако, как правило, ее называют просто программной, а термин программно-аппаратурной реализации используется в том случае, когда отдельные частные алгоритмы реализуются аппаратурно, например, в ФБ, а другие — программно в ЦБУ иерархической или распределенной САК.

3.5. Элементы теории марковских процессов

3.5.1. Постранство состояний. Эволюция системы

Пусть имеется некоторая консультируемая проблема, которая представлена физической системой *S*, которая с течением времени может менять свое состояние. Как мы знаем, состояние системы в каждый момент времени можно характеризовать набором численных значений ее параметров. Эти параметры будем называть *фазовыми координатами системы*, а состояние системы изображать в виде точки *s* с этими координатами в некотором условном *фазовом пространстве*. Тогда изменению состояния системы в процессе ее эволюции соответствует некоторая *траектория точки s в фазовом пространстве*. Фазовое пространство может быть различным в зависимости от числа параметров, характеризующих состояние системы, и от мощности множества возможных состояний системы. В зависимости от числа параметров системы фазовое пространство может быть одномерным, двумерным и вообще *n*-мерным,
(*n* — произвольное целое положительное число). Множество возможных состояний системы (т. е. множество возможных значений параметров системы и их комбинаций) может быть *конечным, счетным или несчетным*. В соответствии с этим фазовое пространство может быть *дискретным* либо *непрерывным*. Часть фазового пространства, в пределах которого реализуется консультационный процесс, будем называть *консультационным пространством*.

Процесс эволюции системы во времени также может протекать непрерывно либо дискретно. Будем говорить, что процесс в системе протекает дискретно, если состояние системы меняется лишь в определенные моменты времени, которые можно пронумеровать.
Рассмотрим случайный процесс в некоторой физической системе, протекающий в дискретном фазовом пространстве:
$\mathcal{E} = \{0,1,2,...\}$,
причем множество моментов времени перехода системы из одного состояния в другое также дискретно, т. е.
$T=\{0, 1, 2, ...\}$.

Введем случайную величину $\xi_l \in \mathcal{E}$, соответствующую номеру состояния, в котором находится система в момент времени l. Обозначим через E_{li} событие, состоящее в том, что в момент времени l система находится в состоянии i, т. е. $E_{li} \sim (\xi_l = i)$.

Полное теоретико-вероятностное описание эволюции системы состоит в определении вероятности того, что для произвольных наборов $l_1, l_2, ..., l_k$ $(l_1 < l_2 < \ldots < l_k)$ и $i_1, i_2, ..., i_k$ имеет место событие

$$E_{l_1 l_2 \ldots l_k; \, i_1 i_2 \ldots i_k} = E_{l_1 i_1} \cap E_{l_2 i_2} \cap \ldots \cap E_{l_k i_k} = \bigcap_{t=1}^{k} E_{l_t i_t},$$

т. е. в заданные k моментов времени система находится в заданных состояниях.

Понятно, что вероятность события

$$E_{l_1 l_2 \ldots l_k, \, i_1 i_2 \ldots i_k}$$

жет быть в принципе вычислена через систему условных вероятностей следующим образом:

$$P(E_{l_1 i_1} \cap E_{l_2 i_2} \cap \ldots \cap E_{l_k i_k} = P(E_{l_1 i_1}) P(E_{l_2 i_2} | _{E_{l_1 l_1}}) \ldots$$
$$\ldots P(E_{l_k i_k} | _{E_{l_1 i_1} \cap E_{l_2 i_2} \cap \ldots \cap E_{l_{k-1} i_{k-1}}}). \quad (3.32)$$

Однако такое описание поведения системы является достаточно сложным. Вместе с тем существует класс случайных процессов, для которых требуемое описание может быть получено более простым путем. Это класс марковских случайных процессов.

3.5.2. Маркоский процесс. Цепи Маркова

Процесс, протекающий в физической системе, называется марковским (или процессом без последействия), если для каждого момента времени поведение системы в будущем зависит только от состояния системы в данный момент и не зависит от того, каким образом система пришла в это состояние. Иначе говорят, что процесс обладает марковским свойством.

Случайный процесс с дискретным временем будем называть случайной последовательностью или случайной цепью.

Случайная цепь, для которой в каждый момент времени дальнейшая последовательность событий зависит только от состояния системы в данный момент, называется марковской цепью.

Иначе, случайная цепь, обладающая марковским свойством, называется простой марковской. Математически это означает следующее:

для любого s= (2, 3, ..., k)

$$P(\xi_{l_s} = i_s | \xi_{l_{s-1}} = i_{s-1}, \xi_{l_{s-2}} = i_{s-2}, \xi_{l_{s-3}} = i_{s-3}, \ldots, \xi_{l_1} = i_1) =$$
$$= P(E_{l_s i_s} | _{E_{l_{s-1} i_{s-1}} \cap E_{l_{s-2} i_{s-2}} \cap \ldots \cap E_{l_1 l_1}}) =$$
$$= P(E_{l_s i_s} | _{E_{l_{s-1} i_{s-1}}}). \quad (3.33)$$

Тогда

$$P(E_{l_1 i_1} \cap E_{l_2 i_2} \cap \dots \cap E_{l_s i_s}) =$$
$$= P(E_{l_1 i_1}) P(E_{l_2 i_2} | E_{l_1 i_1}) P(E_{l_3 i_3} | E_{l_2 i_2}) \dots P(E_{l_s i_s} | E_{l_{s-1} i_{s-1}})$$

или

$$P\left(\bigcap_{t=1}^{s} E_{l_t i_t}\right) = P(E_{l_1 i_1}) \prod_{t=2}^{s} P(E_{l_t i_t} | E_{l_{t-1} i_{t-1}}).$$

Рассмотрим два смежных момента времени: l-й и (l+1)-й. Введем

$$w_{ij}(l, l+1) = P(E_{l+1, j} | E_{l, i}) \qquad (3.34)$$

— условную вероятность перехода системы за один шаг из состояния i в момент времени l в состояние j в момент времени (l+1).

Если вероятность перехода $w_{ij}(l, l+1)$ не зависит от момента времени, когда осуществляется переход, для всех $(i, j) \in \mathcal{E}$, т. е. $w_{ij}(l, l+1) = w_{ij}$, то соответствующая марковская цепь называется *однородной.*

Вероятность перехода $w_{ij} > 0$, если переход из состояния i в состояние j возможен за один шаг; в противном случае $w_{ij}=0$.

Совокупность вероятностей перехода w_{ij} для всех $(i, j) \in \mathcal{E}$ образует матрицу переходов **W**, часто называемую стохастической.

Элементы матрицы переходов удовлетворяют соотношениям

$$\sum_{j \in \mathcal{E}} w_{ij} = 1, \quad i \in \mathcal{E},$$
$$w_{ij} \geqslant 0, \quad (i, j) \in \mathcal{E}.$$

Рассмотрим совокупность вероятностей $\{\mathbf{P}(\xi_l = i)\}$, $i \in \mathcal{E}$, характеризующую распределение вероятностей пребывания системы в различных состояниях в момент времени l. Введенное распределение будем называть в дальнейшем вектодом-состояний системы в момент_времени l и обозначать через $\mathbf{P}(l)$, причем

$$\mathbf{P}(l) = (P_0(l) P_1(l) P_2(l) \dots), \qquad l \in T,$$

где

$$P_i(l) = P(\xi_l = i), \quad i \in \mathcal{E}. \qquad (3.35)$$

Вектор состояний $\mathbf{P}(0) = (P_0(0)\ P_1(0) P_2(0) \dots)$, соответствующий распределению вероятностей состояний системы в нулевой момент времени, будем называть начальным. Понятно, что при наличии пачального вектора системы $\mathbf{P}(0)$ и матрицы переходов **W** можно проследить эволюцию системы. Действительно,

$$P_j(1) = \sum_{i \in \mathscr{E}} P_i(0) w_{ij}, \quad j \in \mathscr{E}.$$

Это же соотношение в матричных обозначениях имеет вид

$$\mathbf{P}(1) = \mathbf{P}_2(0)\mathbf{W}. \tag{3.36}$$

Аналогично

$$\mathbf{P}(2) = \mathbf{P}(1)\mathbf{W} \tag{3.37}$$

или, подставляя (3.36) в (3.37),

$$\mathbf{P}(2) = \mathbf{P}(0)(\mathbf{W})^2.$$

Точно так же

$$\mathbf{P}(3) = \mathbf{P}(0)(\mathbf{W})^3$$

и, вообще,

$$\mathbf{P}(l) = \mathbf{P}(0)(\mathbf{W})^l. \tag{3.38}$$

С другой стороны, введем совокупность $w_{ij}^{(l)}$- условных вероятностей перехода системны из состояния i в состояние j за l шагов. Переход из состояния i в состояние j за l шагов может быть осуществлен различными путями. Условная вероятность $w_{ij}^{(l)}$ есть сумма вероятностей переходов из i в j для всех возможных путей.
При этом, в частности,

$$w_{ij}^{(1)} = w_{ij} \quad \text{и} \quad w_{ij}^{(2)} = \sum_{\nu \in \mathscr{E}} w_{i\nu}^{(1)} w_{\nu j}^{(1)}.$$

По индукции легко показать, что

$$w_{ij}^{(l+m)} = \sum_{\nu \in \mathscr{E}} w_{i\nu}^{(l)} w_{\nu j}^{(m)}, \quad (i, j) \in \mathscr{E}. \tag{3.39}$$

Система уравнений (3.39) носит название уравнений Чэпмена—Колмогорова.

В матричной форме эти уравнения записываются следующим образом:

$$\mathbf{W}^{(l+m)} = \mathbf{W}^{(l)} \mathbf{W}^{(m)}.$$

Используя матрицу переходов за l шагов и начальный вектор состояния системы, легко получить вектор состояния в момент l. Тогда

$$P_j(l) = \sum_{i \in \mathscr{E}} P_i(0) w_{ij}^{(l)}, \quad j \in \mathscr{E},$$

или

$$\mathbf{P}(l) = \mathbf{P}(0)\mathbf{W}^{(l)} \tag{3.40}$$

где $(\mathbf{W})^{(l)}$ — матрица вероятностей переходов за l шагов.
Сравнивая (3.38) и (3.40), получаем

$$\mathbf{W}^{(l)} = (\mathbf{W})^l.$$

Таким образом, матрица переходов за l шагов равна l -й степени матрицы переходов за один шаг.

3.5.3. Классификация состояний

Введем несколько определений.

Состояние j *достижимо* из состояния i, если существует такое k, что $w_{ij}^{(k)} > 0.$

Подмножество C множества возможных состояний $\mathcal{E}$ называется замкнутым, если за один шаг невозможны никакие переходы из состояния, входящего в C, в состояние, не входящее в C, т. е. $w_{ij} = 0$ для всех $j \notin C$.

Цепь называется *неприводимой*, если соответствующее ей множество всех возможных состояний не содержит никаких замкнутых подмножеств, кроме самого себя.

Состояние i называется *возвратным,* если вероятность того, что система, выйдя из этого состояния, когда-либо вернется в него же, равна единице. Если же эта вероятность меньше единицы, то состояние называется *невозвратным.*

Справедлива следующая теорема.

Теорема 1. Состояние i является возвратным тогда и только тогда, когда ряд

$$\sum_{l=0}^{\infty} w^{(l)}{}_{ii}$$

расходится, т. е.

$$\lim_{r\to\infty} \sum_{l=0}^{r} w_{ii}^{(l)} = \infty.$$

Смысл высказанного в теореме утверждения становится понятным, если принять во внимание, что

$$\lim_{r\to\infty} \sum_{l=0}^{r} w_{ii}^{(l)}$$

легко интерпретируется как среднее число возвращений системы в состояние i. В самом деле, введем случайную величину

$$\eta_{li} = \begin{cases} 0, & \text{если } \xi_l \neq i, \\ 1, & \text{если } \xi_l = i. \end{cases}$$

Тогда

$$\eta_i = \sum_{l=1}^{\infty} \eta_{il}$$

есть, очевидно, число пребываний системы в состоянии i и

$$M[\eta_i] = \sum_{l=1}^{\infty} M[\eta_{il}] = \sum_{l=1}^{\infty} w_{ii}^{(l)} = \lim_{r\to\infty} \sum_{l=1}^{r} w_{ii}^{(l)}$$

есть среднее число возвращений системы в состояние i.

Теперь понятно, что если исходное состояние i является возвратным, то система с вероятностью 1 за бесконечное число шагов бесконечно много раз возвратится в i. Если же состояние i является невозвратным, то за бесконечное число шагов система с вероятностью 1 лишь конечное число раз побывает в состоянии i, другими словами, после некоторого конечного числа шагов она никогда больше не возвратится в i.

Отсюда следует, что если i — возвратное состояние и j достижимо из i, то и i, в свою очередь, достижимо из j. Действительно, в противном случае, выйдя из состояния i, система с положительной вероятностью $w^{(l)}_{ij}$ = α попадала бы в состояние j, из которого i недостижимо; таким образом, вероятность возвращения в i была бы не больше чем 1— α, что противоречит возвратности i. Теперь понятно, что если состояние j достижимо из возвратного состояния i, то и состояние j является возвратным.

Наконец, нетрудно видеть, что если цепь Маркова имеет конечное число состояний, причем каждое из них достижимо из любого другого состояния, то все они являются возвратными. Действительно, если имеется лишь конечное число состояний, то за бесконечное число шагов хотя бы в одном из них система побывает бесконечное число раз, т. е. хотя бы одно из состояний системы является возвратным. Поскольку по условию из него можно с положительной вероятностью перейти в любое другое состояние, то все они являются возвратными.
Состояние i называется *поглощающим,* если вероятность ухода системы из этого состояния в любое другое равна нулю, т. е.

$$w_{ij}^{(1)} = 0,\ j \in \mathscr{E},\ j \neq i.$$

Ясно, что если в системе имеется хотя бы одно поглощающее состояние, то ни одно из состояний системы не является возвратным.
Предположим, что в момент времени 0 система находится в произвольном, но фиксированном состоянии i, т. е. $\xi_0=i$. Пусть $f_i^{(l)}$ — вероятность того, что первое возвращение в i произойдет в момент l. При этом ясно, что

$$f_i^{(1)} = w_{ii},$$
$$f_i^{(2)} = w_{ii}^{(2)} - f_i^{(1)} w_{ii},$$
$$\cdots\cdots\cdots\cdots\cdots$$
$$f_i^{(l)} = w_{ii}^{(l)} - f_i^{(1)} w_{ii}^{(l-1)} - f_i^{(2)} w_{ii}^{(l-2)} - \ldots - f_i^{(l-1)} w_{ii}.$$

Сумма

$$F_i = \sum_{l=1}^{\infty} f_i^{(l)}$$

может быть интерпретирована как вероятность того, что система, выйдя из состояния i, когда-либо вернется в это состояние.

Если $F_i=l$, то это возвращение обязательно произойдет. Введем параметр

$$\mu_i = \sum_{l=1}^{\infty} l f_i^{(l)},$$

равный математическому ожиданию времени возвращения.
Возвратное состояние i, для которого $\mu_i=\infty$, называется нулевым. Возвратное состояние, для которого возвращение возможно лишь через число шагов, кратное d, называется периодическим (с периодом d). Возвратное состояние, не являющееся ни нулевым,ни периодическим, называется *эргодическим*.

3.5.4. Предельный вектор

В п. 3.5.2 была введена вероятность $w^{(l)}_{ij}$ перехода системы из состояния i в состояние j за l шагов. Во многих консультационных задачах требуется изучить асимптотическое поведение $w^{(l)}_{ij}$ при $l \to\infty$.

Введем распределение $\{\pi_k\}$, которое будем называть *стационарным*, если

$$\pi_j = \sum_{i \in \mathscr{E}} \pi_i w_{ij}, \quad j \in \mathscr{E}. \tag{3.41}$$

Соотношение (3.41) в векторно-матричной форме имеет вид

$$\mathbf{\Pi} = \mathbf{\Pi W}, \tag{3.42}$$

где $\mathbf{\Pi}=(\pi_0, \pi_1, ..., \pi_k, ...)$—стационарное распределение (распределение $\{\pi_k\}$ часто называют вектором предельных (финальных) вероятностей).

Возможность существования стационарного распределения устанавливается следующей теоремой.

Теорема 2. Если все состояния неприводимой цепи являются эргодическими, то для любой пары состояний i и j

$$\lim_{l\to\infty} w_{ij}^{(l)} = \pi_j$$

существует и не зависит от i, причем $\pi_j=1/\mu_j$, где μ_j— среднее время возвращения в состояние j. Величины π_j удовлетворяют уравнениям

$$\pi_j = \sum_{i\in\mathscr{E}} \pi_i w_{ij}, \quad j \in \mathscr{E},$$

где

$$\sum_{j\in\mathscr{E}} \pi_j = 1.$$

Распределение { π_j } является при этом единственным. Заметим, что если множество состояний конечно и $\mathscr{E} = \{0, 1, 2, ... n\}$, то

$$\lim_{l\to\infty} \mathbf{W}^{(l)} = \begin{vmatrix} \pi_0 & \pi_1 & \pi_2 & \ldots & \pi_j & \ldots & \pi_n \\ \pi_0 & \pi_1 & \pi_2 & \ldots & \pi_j & \ldots & \pi_n \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \pi_0 & \pi_1 & \pi_2 & \ldots & \pi_j & \ldots & \pi_n \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \pi_0 & \pi_1 & \pi_2 & \ldots & \pi_j & \ldots & \pi_n \end{vmatrix}, \quad (3.43)$$

поэтому, в соответствии с (3.38)

$$\mathbf{\Pi} = \mathbf{P}(0) \lim_{l\to\infty} \mathbf{W}^{(l)} = (\pi_0, \pi_1, \ldots, \pi_j, \ldots, \pi_n)$$

и не зависит от начального распределения.

Таким образом, предельный вектор существует в том и только в том случае, если все состояния неприводимой цепи являются эргодическими.

Понятно, что если цепь имеет конечное число возможных состояний, каждое из них достижимо из любого другого состояния и не является периодическим, то предельный вектор для такой цепи существует. Если же состояния цепи являются периодическими, то $\lim\limits_{l\to\infty} \mathbf{W}^{(l)}$ не существует, хотя стационарное распределение, удовлетворяющее (3.41), может существовать.

Так, например, для цепи с двумя состояниями и матрицей переходов

$$\mathbf{W} = \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix}$$

предельной матрицы нет, так как

$$\mathbf{W}^{(2)}=\begin{vmatrix}1 & 0\\ 0 & 1\end{vmatrix},\quad \mathbf{W}^{(3)}=\begin{vmatrix}0 & 1\\ 1 & 0\end{vmatrix},\ \ldots,\ \mathbf{W}^{(2k)}=\begin{vmatrix}1 & 0\\ 0 & 1\end{vmatrix},$$

$$\mathbf{W}^{(2k+1)}=\begin{vmatrix}0 & 1\\ 1 & 0\end{vmatrix},\ \ldots,$$

в то время как предельное распределение существует и

$$\Pi=|\,{}^1/_2,\quad {}^1/_2\,|.$$

Совокупность линейных алгебраических, уравнений (3.41), дополненная условием нормировки

$$\sum_{j\in\mathscr{E}}\pi_j=1,$$

позволяет рассчитать компоненты предельного вектора. Важно заметить, что существование предельного распределения и численные значения его компонент не зависят, как уже было сказано, от начального распределения вероятностей состояний системы **P**(0), а целиком определяются лишь стохастической матрицей **W**. Систему, для которой предельный вектор существует, будем называть эргодической.

3.5.5. Отображение марковской цепи в виде графа

Рассмотрим стохастическую матрицу **W** марковской цепи. Если некоторый элемент w_{ij} этой матрицы отличен от нуля, переход из состояния *i* в состояние *j* возможен за один шаг.

Множество всех состояний системы и выполнимых переходов очень удобно отображать в виде графа. Введем следующее определение. *Граф* есть совокупность вершин, соединенных направленными дугами. Соответствие между множеством состояний и возможных переходов системы, с одной стороны, и множеством вершин и дуг графа, с другой, установим следующим образом.

Множество состояний системы отображается совокупностью вершин графа, а возможные переходы системы— в виде дуг графа, причем направление дуги указывает, из какого состояния и в какое переходит система. Каждой дуге графа припишем число, равное соответствующей вероятности перехода, системы за один шаг. Таким образом, матрица переходов системы **W** и ее граф взаимно однозначно соответствуют друг другу.

Пусть, например, матрица переходов системы имеет вид

$$\mathbf{W}=\begin{Vmatrix} 1/2 & 0 & 1/2 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 1/4 & 0 & 1/4 & 1/2 \\ 1 & 0 & 0 & 0 \end{Vmatrix}.$$

Соответствующий этой системе граф изображен на рис. 3.52.

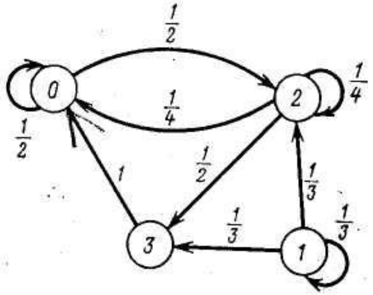

Рис. 3.52.

3.5.6. Примеры применения теории цепей Маркова

Пример 1 (случайное «блуждание» рекомендации). Рассмотрим случайное, при котором сформированная рекомендация из каждой целочисленной точки i своего первоначального использования на следующем шаге с вероятностью p переходит в соседнюю справа точку j=i+1 своего последующего использования и с вероятностью q — в соседнюю слева точку j = i—1, (p+ q =1). Ясно, что в этом случае

$$w_{ij}^{(1)}=\begin{cases} p, & j=i+1, \\ q, & j=i-1, \\ 0, & (j=i)\vee(|j-i|>1). \end{cases}$$

Матрица перехода **W** системы имеет вид

$$\mathbf{W}=\begin{Vmatrix} . & . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & q & 0 & p & 0 & 0 & 0 & 0 & . & . & . \\ . & . & . & 0 & q & 0 & p & 0 & 0 & 0 & . & . & . \\ . & . & . & 0 & 0 & q & 0 & p & 0 & 0 & . & . & . \\ . & . & . & 0 & 0 & 0 & q & 0 & p & 0 & . & . & . \\ . & . & . & 0 & 0 & 0 & 0 & q & 0 & p & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . & . \end{Vmatrix}$$

Понятно, что каждое состояние этой цепи достижимо из любого другого состояния. Непосредственно можно убедиться в том, что

$$w_{ii}^{(l)} = \begin{cases} 0, & l = 2k+1, \\ C_{2k}^k p^k q^k, & l = 2k,\ k > 0. \end{cases}$$

Используя формулу Стерлинга, при $k \to \infty$ получаем

$$C_{2k}^k p^k q^k = \frac{(2k)!}{(k!)^2} p^k q^k \approx \frac{\sqrt{4\pi k}\left(\frac{2k}{e}\right)^{2k} p^k q^k}{\left(\sqrt{2\pi k}\left(\frac{k}{e}\right)^k\right)^2} = \frac{(4pq)^k}{\sqrt{\pi k}},$$

где $4pq = (p+q)^2 - (p-q)^2 = 1 - (p-q)^2 \leq 1$, причем знак равенства имеет место лишь при $p = q = 1/2$.
Таким образом, при $k \to \infty$

$$w_{ii}^{(2k)} \approx \frac{1}{\pi k}(4pq)^k,$$

откуда следует, что ряды

$$\sum_{k=1}^{\infty} w_{ii}^{(2k)} \quad \text{и} \quad \sum_{k=1}^{\infty} \frac{(4pq)^k}{\sqrt{\pi k}}$$

сходятся или расходятся одновременно.
Если $p \neq q$, то $4pq < 1$ и ряд

$$\sum_{k=1}^{\infty} w_{ii}^{(2k)}$$

сходится в соответствии с признаком Даламбера, так как

$$\lim_{k\to\infty} \frac{a_{k+1}}{a_k} = \lim_{k\to\infty} \frac{(4pq)^{k+1}\sqrt{\pi k}}{(4pq)^k \sqrt{\pi(k+1)}} = 4pq \lim_{k\to\infty} \sqrt{\frac{k}{k+1}} = 4pq < 1.$$

В этом случае каждое состояние i системы является невозвратным Интуитивно ясно, что при этом сформированная рекомендация будет неограниченно удаляться от ее использования в положительном направлении (при $p>q$) или в отрицательном (при $p<q$), рано или поздно навсегда покидая любое фиксированное состояние (использование) i. Если же $p—q$, то $4pq=1$ и ряд

$$\sum_{k=1}^{\infty} w_{ii}^{(2k)} = \frac{1}{\sqrt{\pi}} \sum_{k=1}^{\infty} \frac{1}{\sqrt{k}}$$

расходится, так как члены этого ряда, начиная с некоторого, становятся больше соответствующих членов гармонического ряда

$$\Gamma = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots,$$

который, как известно, расходится. В этом случае все состояния системы являются возвратными и сформированная рекомендация при неограниченно продолжающемся симметричном случайном использовании бесконечное число раз возвращается в каждое из состояний использования. Для системы со случайным блужданием предельный вектор не существует, так как ни при $p = q$, ни при $p \neq q$ состояния цепи не являются эргодическими.

Пример 2. Система передачи сформированных рекомендаций работает в режиме, называемом нормальным, до появления сбоев в n сообщениях о передаче сформированных рекомендаций подряд. В этом случае система переходит в режим аварии, в котором остается до тех пор, пока очередное сообщение о передаче сформированных рекомендаций не будет принято правильно. После этого система возвращается в нормальный режим. Пусть вероятность появления сбоя в очередном сообщении о передаче сформированных рекомендаций равна p, а вероятность безошибочного приема сформированных рекомендаций равна q=1—p.

Введем множество возможных состояний системы:

E_0 — сбоев нет;

E_1 — имеется сбой в одном сообщении;

E_2 — имеется сбой в двух сообщениях подряд;

…………………………………………………………………

E_n — имеется сбой в n сообщениях подряд.

Понятно, что процесс, протекающий в такой системе, является марковским. Множество возможных состояний системы конечно и содержит n+1 элементов. Матрица переходов системы имеет вид

$$\mathbf{W} = \begin{vmatrix} q & p & 0 & \ldots & 0 & 0 \\ q & 0 & p & \ldots & 0 & 0 \\ q & 0 & 0 & \ldots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ q & 0 & 0 & \ldots & p & 0 \\ q & 0 & 0 & \ldots & 0 & p \\ q & 0 & 0 & \ldots & 0 & p \end{vmatrix}.$$

Граф состояний и переходов изображен на рис. 3.53.

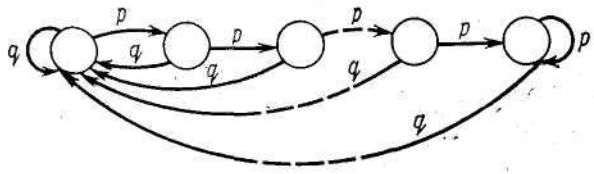

Рис. 3.53.

Очевидно, что каждое состояние системы достижимо из любого другого состояния. Поэтому, учитывая, что множество возможных состояний системы конечно, можно утверждать, что в этой системе существует предельный вектор.

Отыщем компоненты этого вектора, используя соотношение

$$\mathbf{\Pi} = \mathbf{\Pi W}$$

или

$$\pi_0 = \sum_{i=0}^{n} \pi_i w_{i0},$$

$$\pi_1 = \sum_{i=0}^{n} \pi_i w_{i1},$$

$$\cdots\cdots\cdots$$

$$\pi_n = \sum_{i=0}^{n} \pi_i w_{in}. \tag{3.44}$$

В рассматриваемом случае система уравнений (3.44) приобретает вид

$$\pi_0 = q\sum_{i=0}^{n} \pi_i = q,$$

$$\pi_1 = \pi_0 p = pq,$$

$$\pi_2 = \pi_1 p = p^2 q,$$

$$\cdots\cdots\cdots$$

$$\pi_{n-1} = \pi_{n-2} p = p^{n-1} q,$$

$$\pi_n = \pi_{n-1} p + \pi_n p. \tag{3.45}$$

Из последнего уравнения системы (3.45) имеем

$$\pi_n = (p/q)\,\pi_{n-1} = p^n.$$

Таким образом, предельный вектор системы **П** имеет вид

$$\Pi = (q,\ pq,\ p^2 q,\ \ldots,\ p^{n-1} q,\ p^n).$$

Пример 3. На выходе приемного устройства радиолокатора в результате локации цели появляется отраженный сигнал. Из-за наложения на этот сигнал внутренних шумов приемника и изрезанности диаграммы направленности вторичного излучения цели принимаемый сигнал флуктуирует. Пусть вероятность превышения сигналом некоторого заранее выбранного порога равна p (вероятность непревышения, естественно, равна q=1—p).

Сформируем следующие рекомендации работы устройства обнаружения целей: а) цель следует считать обнаруженной, если сигнал превышает порог при двух последовательных локациях; б) цель следует считать потерянной, если сигнал не превышает порог дважды подряд.

Проанализируем работу устройства. Рекомендуется ввести следующее множество возможных состояний системы:

E_0 — исходное состояние. После очередной локации система должна оставаться в этом состоянии с вероятностью q и переходить в состояние E_1 с вероятностью p;

E_1 — состояние, соответствующее однократному превышению порога. Если при очередной локации сигнал будет превышать порог (вероятность этого события равна p), то система должна переходить в состояние E_2, в противном случае — должна возвращаться в исходное состояние (вероятность этого события равна q).

E_2 — состояние, соответствующее двухкратному превышению порога подряд (обнаружение цели). Система должна оставаться в этом состоянии с вероятностью p и переходить и состояние E_3 с вероятностью q;

E_3 — состояние, соответствующее однократному непревышению порога после обнаружения цели. Если после очередной локации сигнал превышает порог, система должна возвратиться в состояние E_2, в противном случае — перейти в исходное состояние E_0. Граф переходов системы приведен на рис. 3.54.

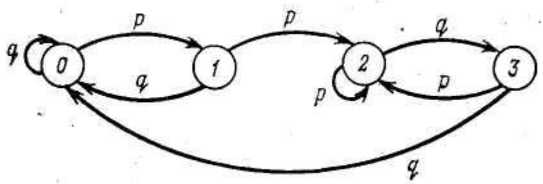

Рис. 3.54.

Выпишем матрицу переходов системы

$$\mathbf{W} = \begin{vmatrix} q & p & 0 & 0 \\ q & 0 & p & 0 \\ 0 & 0 & p & q \\ q & 0 & p & 0 \end{vmatrix}.$$

Поскольку множество состояний конечно и все они достижимы из j любого другого, система обладает эргодическим свойством и предельный вектор существует.

Для отыскания компонент этого вектора используем соотношение

$$\mathbf{П} = \mathbf{ПW}.$$

Так как в рассматриваемом примере **П**=($\pi_0, \pi_1, \pi_2, \pi_3$), то

$$\pi_0 = \sum_{i=0}^{3} \pi_i w_{i0}, \qquad \pi_1 = \sum_{i=0}^{3} \pi_i w_{i1},$$

$$\pi_2 = \sum_{i=0}^{3} \pi_i w_{i2}, \qquad \pi_3 = \sum_{i=0}^{3} \pi_i w_{i3},$$

откуда

$$\pi_0 = q(\pi_0 + \pi_1 + \pi_3), \qquad \pi_1 = p\pi_0,$$

$$\pi_2 = p(\pi_1 + \pi_2 + \pi_3), \qquad \pi_3 = q\pi_2.$$

Решим эту систему, дополнив ее условием нормировки

$$\pi_0 + \pi_1 + \pi_2 + \pi_3 = 1,$$

решение которой дает нам следующий результат использования сформированных рекомендаций

$$\pi_0 = \frac{q^2}{1 - pq}; \qquad \pi_1 = \frac{pq^2}{1 - pq}; \qquad \pi_2 = \frac{p^2}{1 - pq}; \qquad \pi_3 = \frac{qp^2}{1 - pq}.$$

3.5.7. Асимптотическое поведение неэргодических систем

Практический интерес имеет изучение асимптотического поведения системы, не все состояния которой являются эргодическими. Рассмотрим, например, марковскую цепь, граф переходов которой изображен на рис. 3.55.

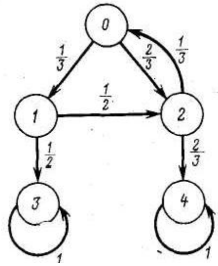

Рис. 3.55.

Эта цепь содержит два поглощающих состояния (i = 3 и i = 4). Понятно, что предельный вектор такой системы в том смысле, как он был введен ранее, не существует. В самом деле, состояние, в котором окажется эта система через достаточно большое число шагов, зависит от исходного. Если исходным состоянием было одно из состояний множества {0, 1, 2}, то система придет, в конце концов, в одно из двух поглощающих состояний. Если же исходным было одно из поглощающих состояний, то система навсегда в нем и останется.

Таким образом, для характеристики асимптотического поведения такой системы уже недостаточно введения предельного вектора **П**, так как он будет различным в зависимости от исходного состояния.

В связи с этим введем матрицу **B**=(b_{ij}), строки которой определяли бы предельные распределения вероятностей различных состояний системы, причем номер строки указывал бы номер исходного состояния, т. е.

$$\mathbf{B}=\begin{Vmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{i1} & b_{i2} & \dots & b_{in} \\ \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{Vmatrix} = \begin{Vmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \dots \\ \mathbf{B}_i \\ \dots \\ \mathbf{B}_n \end{Vmatrix},$$

где $\mathbf{B}_i(b_{i1}b_{i2}\ \dots\ b_{in})$ — вектор распределения предельных вероятностей системы, если исходным было состояние с номером i.

Метод расчета элементов матрицы **B** состоит в следующем. Множество всех поглощающих состояний обозначим через T, а множество всех остальных состояний системы (они все являются невозвратными) обозначим через $\bar{T}$. Пусть множество возможных состояний системы содержит k поглощающих. Перенумеруем все состояния системы таким образом, чтобы поглощающим состояниям соответствовали бы последние k номеров, т, е.

$$\bar{T}=\{i : i\in\mathscr{E},\quad 0\leqslant i\leqslant n-k\},$$

$$T=\{i : i\in\mathscr{E},\quad n-k+1\leqslant i\leqslant n\}.$$

Тогда матрица переходов **W** системы будет иметь следующую структуру:

$$\mathbf{W}=\begin{Vmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{O} & \mathbf{I} \end{Vmatrix}.$$

Подматрица **Q** содержит вероятности переходов из невозвратных состояний в невозвратные, а подматрица **R** — вероятности переходов из невозвратных состояний в поглощающие. По аналогии с (3.43) запишем

$$\mathbf{B} = \lim_{l \to \infty} \mathbf{W}^{(l)}. \tag{3.46}$$

Возведя (путем непосредственного перемножения) **W** в l-ю степень, имеем

$$\mathbf{W}^{(l)} = \begin{vmatrix} \mathbf{Q}^{(l)} & \mathbf{B}^*{}_l \\ \mathbf{O} & \mathbf{I} \end{vmatrix}, \tag{3.47}$$

где

$$\mathbf{B}^*{}_l = (\mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \ldots + \mathbf{Q}^{l-1})\mathbf{R} = (\mathbf{I} - \mathbf{Q})^{-1}(\mathbf{I} - \mathbf{Q}^l)\mathbf{R}.$$

Выполнив предельный переход по l, в результате получим

$$\lim_{l \to \infty} \mathbf{Q}^l = \mathbf{O},$$

$$\lim_{l \to \infty} \mathbf{B}^*{}_l = \lim_{l \to \infty} (\mathbf{I} - \mathbf{Q})^{-1}(\mathbf{I} - \mathbf{Q}^l)\mathbf{R} = (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{R} = \mathbf{B}^*. \tag{3.48}$$

Таким образом, объединяя (3.46) — (3.48), можно записать, что

$$\mathbf{B} = \begin{vmatrix} \mathbf{O} & \mathbf{B} \\ \mathbf{O} & \mathbf{I}^* \end{vmatrix}.$$

Заметим, что для подматрицы **B*** выполняется следующее соотношение:

$$\mathbf{B}^* = \mathbf{R} + \mathbf{Q}\mathbf{B}^\#, \tag{3.49}$$

справедливость которого становится ясной из следующих соображений.

Пусть $i \in \overline{T}$ является начальным состоянием процесса и $j \in T$ есть некоторое поглощающее состояние. Выйдя из i, процесс может поглотиться в j на первом шаге или на одном из последующих шагов. Вероятность захвата j-м поглощающим состоянием на первом шаге равна w_{ij}. Другими исходами первого шага могут быть захват каким-либо другим поглощающим состоянием (тогда достигнуть j-го состояния будет невозможно) или переход в некоторое другое,, например k-е ($k \in \overline{T}$), невозвратное состояние. В последнем случае процесс поглотится в состоянии j с вероятностью b_{kj}. Следовательно,

$$b_{is} = w_{ij} + \sum_{k \in T} w_{ik} b_{kj}.$$

Матричным аналогом этого соотношения и является (3.49).

Покажем теперь, что

$$\mathbf{WB} = \mathbf{B}. \tag{3.50}$$

Действительно,

$$\mathbf{WB} = \begin{vmatrix} \mathbf{Q} & \mathbf{R} \\ 0 & 1 \end{vmatrix} \begin{vmatrix} 0 & \mathbf{B}^* \\ 0 & 1 \end{vmatrix} = \begin{vmatrix} 0 & \mathbf{QB}^* + \mathbf{R} \\ 0 & 1 \end{vmatrix},$$

и с учетом (3.49) получаем требуемое.

Матричное уравнение (3.50) трансформируется в совокупность уравнений вида

$$\mathbf{WP}_j = \mathbf{P}_j, \quad j = 0, 1, \ldots, n, \tag{3.51}$$

где

$$\mathbf{P}_j = \begin{vmatrix} b_{0j} \\ b_{1j} \\ \cdots \\ b_{ij} \\ \cdots \\ b_{nj} \end{vmatrix}$$

— вектор-столбец, содержащий вероятности поглощения в j-м состоянии для различных исходных состояний.

Векторно-матричные уравнения (3.51) решаются уже описанным выше способом, путем преобразования к системе алгебраических уравнений

$$\begin{aligned} &w_{00}b_{0j} + w_{01}b_{1j} + \ldots + w_{0n}b_{nj} = b_{0j}, \\ &w_{10}b_{0j} + w_{11}b_{1j} + \ldots + w_{1n}b_{nj} = b_{1j}, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &w_{n0}b_{0j} + w_{n1}b_{1j} + \ldots + w_{nn}b_{nj} = b_{nj}. \end{aligned} \tag{3.52}$$

Система (3.52) может быть упрощена, если учесть структуру матриц **W** и **B**. Действительно, так как

$$w_{ij} = \begin{cases} 0, & i \in T, \quad j \in T, \quad i \neq j, \\ 1, & i \in T, \quad j \in T, \quad i = j, \end{cases}$$

$$b_{ij} = \begin{cases} 0, & (j \in \overline{T}) \vee [(j \in T) \wedge (i \in T) \wedge (i \neq j)], \\ 1, & (j \in T) \wedge (i = j), \end{cases}$$

то уравнения (3.52) трансформируются к виду

$$\begin{aligned} &w_{00}b_{0j} + w_{01}b_{1j} + \ldots + w_{0,\, n-k}b_{n-k,\, j} + w_{0j} = b_{0j}, \\ &w_{10}b_{0j} + w_{11}b_{1j} + \ldots + w_{1,\, n-k}b_{n-k,\, j} + w_{1j} = b_{1j}, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &w_{n-k,\, 0}b_{0j} + w_{n-k,\, 1}b_{1j} + \ldots + w_{n-k,\, n-k}b_{n-k,\, j} + \\ &+ w_{n-k,\, j} = b_{n-k,\, j}, \\ &\qquad j = n-k+1, \ldots, n. \end{aligned} \tag{3.53}$$

Итак, решая систему уравнений (3.53) для различных j $(j = n—k+1.....n)$, последовательно заполняем матрицу **B**.

Как уже указывалось, строки этой матрицы задают распределения вероятностей поглощения для различных исходных состояний процесса.

Проиллюстрируем изложенную методику на примере марковской цепи, граф переходов которой изображен на рис. 3.55.

Пример.

$$\mathbf{W} = \begin{vmatrix} 0 & {}^1/_3 & {}^2/_3 & 0 & 0 \\ 0 & 0 & {}^1/_2 & {}^1/_2 & 0 \\ {}^1/_3 & 0 & 0 & 0 & {}^2/_3 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix} \quad \overline{T} = \{0;\ 1;\ 2\}, \quad T = \{3;\ 4\}.$$

$$w_{00}b_{0j} + w_{01}b_{1j} + w_{02}b_{2j} + w_{0j} = b_{0j},$$
$$w_{10}b_{0j} + w_{11}b_{1j} + w_{12}b_{2j} + w_{1j} = b_{1j},$$
$$w_{20}b_{0j} + w_{21}b_{1j} + w_{22}b_{2j} + w_{2j} = b_{2j} \quad (j = 3;\ 4).$$

$j = 3$

$$\frac{1}{3}b_{13} + \frac{2}{3}b_{23} = b_{03},$$
$$\frac{1}{2}b_{23} + \frac{1}{2} = b_{13},$$
$$\frac{1}{3}b_{03} = b_{23}.$$

Решая эту систему уравнений, имеем

$$b_{03} = {}^3/_{13}, \quad b_{13} = {}^7/_{13}, \quad b_{23} = {}^1/_{13}.$$

$j = 4$

$$\frac{1}{3}b_{14} + \frac{2}{3}b_{24} = b_{04}$$
$$\frac{1}{2}b_{24} = b_{14}$$
$$\frac{1}{3}b_{04} + \frac{2}{3} = b_{24}.$$

В результате решения этой системы уравнений, имеем

$$b_{04} = {}^{10}/_{13}, \quad b_{14} = {}^6/_{13}, \quad b_{24} = {}^{12}/_{13}.$$

Таким образом, матрица **B** имеет вид

$$\mathbf{B} = \begin{vmatrix} 0 & 0 & 0 & {}^{3}/_{13} & {}^{10}/_{13} \\ 0 & 0 & 0 & {}^{7}/_{13} & {}^{6}/_{13} \\ 0 & 0 & 0 & {}^{1}/_{13} & {}^{12}/_{13} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}.$$

Хорошей проверкой правильности решения задачи является выполнение условия нормировки

$$\sum_{j=0}^{n} b_{ij} = 1$$

для строк матрицы B.

Заметим, что вычислительная процедура отыскания элементов матрицы **B** по объему эквивалентна k -кратному решению системы из $n—k+1$ линейных алгебраических уравнений с таким же числом неизвестных. В то же время, если иметь в виду, что исходное состояние системы обычно фиксируется, интерес представляет лишь одна строка матрицы **B**, номер которой равен номеру исходного состояния. В связи с этим представляется целесообразной разработка метода расчета предельного распределения, не использующего громоздкую процедуру вычисления элементов **B**. С этой целью преобразуем исходную марковскую цепь в псевдоэргодическую путем введения фиктивных переходов из поглощающих состояний в начальное, приписав им некоторую вероятность α. При этом для каждого из поглощающих состояний вероятность перехода в себя должна быть уменьшена на α. С учетом этого граф переходов, например, для цепи, анализ которой проведен в рассмотренном выше примере, будет иметь вид, показанный на рис. 3.56.

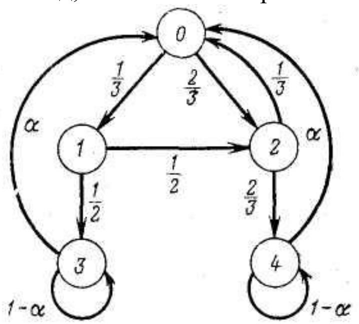

Рис. 3.56.

Здесь для определенности в качестве исходного выбрано нулевое состояние цепи.

Предельный вектор такой цепи уже может быть рассчитан в результате решения векторно-матричного уравнения

$$\mathbf{\Pi}(\alpha)=\mathbf{\Pi}(\alpha)\,\mathbf{W}(\alpha),$$

$$\mathbf{W}(\alpha)=\left|\begin{array}{c:c} \mathbf{Q} & \mathbf{R} \\ \hdashline \begin{matrix}\alpha\\ \alpha\\ \cdot\\ \cdot\\ \cdot\\ \alpha\end{matrix}\ \mathbf{O} & \mathbf{I}(1-\alpha) \end{array}\right|. \tag{3.54}$$

С учетом перенумерации состояний искомый вектор решения может быть записан следующим образом:

$$\mathbf{\Pi}(\alpha)=|\,\mathbf{X}_{\text{н}}(\alpha)\,\vdots\,\mathbf{X}_{\text{п}}(\alpha)\,|, \tag{3.55}$$

где $\mathbf{X}_{\text{н}}(\alpha)$—вектор-строка, содержащий n—k+1 компонент, соответствующих непоглощающим состояниям цепи;

$\mathbf{X}_{\text{п}}(\alpha)$ —вектор-строка, содержащий k компонент, соответствующих поглощающим состояниям.

Покажем теперь, что компоненты предельного вектора **П**(α), получаемого в результате решения уравнения (3.54), после предельного перехода по α в точности соответствуют компонентам первой строки матрицы **B**. Введем матрицы

$$\boldsymbol{\alpha}=\left|\begin{matrix}\alpha\\ \alpha\\ \cdot\\ \cdot\\ \cdot\\ \alpha\end{matrix}\right|;\quad \mathbf{Q}_1=\left|\begin{matrix}q_{00}\\ q_{10}\\ \cdot\\ \cdot\\ \cdot\\ q_{n-k,\,0}\end{matrix}\right|;\quad \tilde{\mathbf{Q}}=\left|\begin{matrix} q_{01}q_{02} & \cdots & q_{0,\,n-k}\\ q_{11}q_{12} & \cdots & q_{1,\,n-k}\\ \cdots & \cdots & \cdots\\ \cdots & \cdots & \cdots\\ q_{n-k,\,1}q_{n-k,\,2} & \cdots & q_{n-k,\,n-k}\end{matrix}\right| \tag{3.56}$$

с размерностями, соответственно равными $k\times1$, $(n—k+1)\times1$, $(n—k+1)\times(n—k)$. С учетом (3.55) и (3.56) перепишем (3.54) в виде

$$|\,\mathbf{X}_{\text{н}}(\alpha)\,\vdots\,\mathbf{X}_{\text{п}}(\alpha)\,|=|\,\mathbf{X}_{\text{н}}(\alpha)\,\vdots\,\mathbf{X}_{\text{п}}(\alpha)\,|\left|\begin{array}{c:c:c}\mathbf{Q}_1 & \tilde{\mathbf{Q}} & \mathbf{R}\\ \hdashline \boldsymbol{\alpha} & \mathbf{O} & \mathbf{I}(1-\alpha)\end{array}\right|$$

или

$$|\mathbf{X}_{н}(\alpha) \vdots \mathbf{X}_{п}(\alpha)| = |\mathbf{X}_{н}(\alpha)\mathbf{Q}_1 + \mathbf{X}_{п}(\alpha)\alpha \vdots \mathbf{X}_{н}(\alpha)\tilde{\mathbf{Q}} | \mathbf{X}_{н}(\alpha)\mathbf{R} +$$
$$+ \mathbf{X}_{п}(\alpha)\mathbf{I}(1-\alpha)| =$$
$$= \left| \mathbf{X}_{н}(\alpha)\mathbf{Q}_1 + \alpha \sum_{j=n-k+1}^{n} x_j | \mathbf{X}_{н}(\alpha)\tilde{\mathbf{Q}} \vdots \mathbf{X}_{н}(\alpha)\mathbf{R} + \mathbf{X}_{п}(\alpha)(1-\alpha) \right|,$$

откуда

$$\mathbf{X}_{н}(\alpha) = \mathbf{X}_{н}(\alpha)\mathbf{Q} + \mathbf{A}^T,$$
$$\mathbf{X}_{п}(\alpha) = \mathbf{X}_{н}(\alpha)\mathbf{R} + \mathbf{X}_{п}(\alpha)(1-\alpha), \qquad (3.57)$$

где

$$\mathbf{A} = \begin{vmatrix} \alpha \sum\limits_{j=n-k+1}^{n} x_j(\alpha) \\ 0 \\ 0 \\ \vdots \\ 0 \end{vmatrix}$$

— вектор размерности $(n—k+1)\times 1$; T — знак транспонирования. Из первого уравнения системы (3.57) имеем

$$\mathbf{X}_{н}(\alpha) = \mathbf{A}^T(\mathbf{I}-\mathbf{Q})^{-1}. \qquad (3.58)$$

Второе уравнение системы упрощается к виду

$$\mathbf{X}_{п}(\alpha) = \frac{1}{\alpha}\mathbf{X}_{н}(\alpha)\mathbf{R}. \qquad (3.59)$$

Подставляя (3.58)в (3.59), получаем

$$\mathbf{X}_{п}(\alpha) = \frac{1}{\alpha}\mathbf{A}^T(\mathbf{I}-\mathbf{Q})^{-1}\mathbf{R} =$$
$$= \begin{vmatrix} \sum\limits_{j=n-k+1}^{n} x_j(\alpha) \\ 0 \\ \vdots \\ 0 \end{vmatrix} (\mathbf{I}-\mathbf{Q})^{-1}\mathbf{R}. \qquad (3.60)$$

С учетом условия нормировки

$$\sum_{j=0}^{n} x_j(\alpha) = 1,$$

а также имея в виду (3.48), перепишем (3.60) следующим образом:

$$\mathbf{X}_п(\alpha) = \begin{vmatrix} 1 - \sum_{j=0}^{n-k} x_j(\alpha) \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{vmatrix} (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{R} =$$

$$= \begin{vmatrix} 1 - \sum_{j=0}^{n-k} x_j(\alpha) \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{vmatrix} \mathbf{B}^* = \mathbf{B}^*_0 - \mathbf{B}^*_0 \sum_{j=0}^{n-k} x_j(\alpha),$$

(3.61)

где $\mathbf{B}^*_0$ —нулевая строка матрицы $\mathbf{B}^*$.

Если теперь принять во внимание, что

$$\lim_{\alpha \to 0} x_j(\alpha) = x_j(0) = 0, \quad j = 0,\ 1,\ 2, \ldots,\ n-k,$$

то окончательно имеем

$$\mathbf{X}_п = \lim_{\alpha \to 0} \mathbf{X}_п(\alpha) = \mathbf{B}^*_0; \qquad \mathbf{X}_н = \lim_{\alpha \to 0} \mathbf{X}_н(\alpha) = \mathbf{O},$$

что и требовалось,

Таким образом, предельное распределение вероятностей системы с произвольным числом поглощающих состояний может быть найдено в результате решения векторно-матричного уравнения (3.54) с последующим предельным переходом по α.

Заметим, что методика расчета предельного вектора не меняется, если изменить начальное состояние. Однако фиктивные переходы из поглощающих состояний в начальное необходимо ввести соответствующим этому изменению образом. Проиллюстрируем изложенную методику на примере цепи, граф которой изображен на рис. 3.55. Граф соответствующей псевдоэргодической цепи изображен на рис. 3.56.

Матрица переходов для этого графа имеет вид:

$$\mathbf{W}(\alpha) = \begin{vmatrix} 0 & 1/3 & 2/3 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 1/3 & 0 & 0 & 0 & 2/3 \\ \alpha & 0 & 0 & 1-\alpha & 0 \\ \alpha & 0 & 0 & 0 & 1-\alpha \end{vmatrix}.$$

Рассчитаем компоненты предельного вектора **П**(α). С этой целью решим следующую систему уравнений:

$$\pi_0(\alpha) = {}^1/_3\pi_2(\alpha) + \alpha\,[\pi_3(\alpha) + \pi_4(\alpha)],$$
$$\pi_1(\alpha) = {}^1/_3\pi_0(\alpha),$$
$$\pi_2(\alpha) = {}^2/_3\pi_0(\alpha) + {}^1/_2\pi_1(\alpha),$$
(3.62)
$$\pi_3(\alpha) = \frac{1}{2}\pi_1(\alpha) + (1-\alpha)\,\pi_3(\alpha),$$
$$\pi_4(\alpha) = \frac{2}{3}\pi_2(\alpha) + (1-\alpha)\,\pi_4(\alpha).$$
(3.63)

Решая подсистему (3.62), имеем

$$\pi_0(\alpha) = \frac{18}{13}\alpha\,[\pi_3(\alpha) + \pi_4(\alpha)],$$
$$\pi_1(\alpha) = \frac{6}{13}\alpha\,[\pi_3(\alpha) + \pi_4(\alpha)],$$
$$\pi_2(\alpha) = \frac{15}{13}\alpha\,[\pi_3(\alpha) + \pi_4(\alpha)].$$
(3.64)

После упрощения (3.63) и использования (3.64) получим

$$\pi_3(\alpha) = \frac{3}{13}[\pi_3(\alpha) + \pi_4(\alpha)],$$
$$\pi_4(\alpha) = \frac{10}{13}[\pi_3(\alpha) + \pi_4(\alpha)].$$

Если учесть теперь, что

$$\lim_{\alpha\to 0}[\pi_3(\alpha) + \pi_4(\alpha)] = \lim_{\alpha\to 0}\{1 - [\pi_0(\alpha) + \pi_1(\alpha) + \pi_2(\alpha)]\} = 1,$$

то искомый предельный вектор имеет вид

$$\Pi = |\,0\;\;0\;\;0\;\;{}^3/_{13}\;\;{}^{10}/_{13}\,|.$$

Как и следовало ожидать, вычисленные компоненты предельного вектора совпадают с элементами нулевой строки матрицы **В**, полученной ранее.

3.5.8. Применение теории марковских цепей для оценки эффективности консультруемых проблем

Процесс функционирования консультируемой проблемы определяется алгоритмом работы системы, который обеспечивает приспосабливающееся к изменениям внешней среды поведение системы в соответствии с логикой ее алгоритма. Реакция алгоритма на ту или иную комбинацию внешних воздействий определяет эффективность системы в этой конкретной ситуации. Оценка эффективности системы на всем множежестве возможных входных воздействий может быть получена, если будет найдено соответствующее распределение вероятностей реализации различных реакций на выходе алгоритма системы.

Рекомендуется множество консультируемых ситуаций, каждая из которых соответствует фиксированной комбинации входных воздействий на систему, пронумеровать и образовать алфавит **А**. Аналогичным образом рекомендуется сформировать алфавит возможных реакций системы **Ф**. Таким образом, *алгоритм — это алфавитный оператор, отображающий элементы множества **А** на элементы множества **Ф**.*

Поставим консультационную задачу - отыскать распределения вероятностей реализации различных реакций на выходе консультационного алгоритма при фиксированных статистических характеристиках и структуре входных воздействий на систему.

Введем предварительно несколько определений. Обозначим через $E(s)$ множество состояний, в которых может оказаться система в процессе функционирования, если исходным является состояние s. Тогда будем говорить, что в консультационном алгоритме имеются циклы, если существует хотя бы одно состояние s_i такое, что $s_i \in E(s_i)$. Кроме того, будем считать, что в консультационном алгоритме имеются пересечения, если существует хотя бы одна пара s_i и s_k таких, что

$$E(s_i) \cap E(s_k) \neq \varnothing .$$

Если консультационнiй алгоритм имеет простую ветвящуюся структуру без циклов и пересечений, искомое распределение может быть легко получено. В самом деле, ветвящийся консультационный алгоритм без циклов и пересечений схематически может быть представлен в виде дерева (рис. 3.57), состоящего из узлов и соединяющих их направленных дуг.

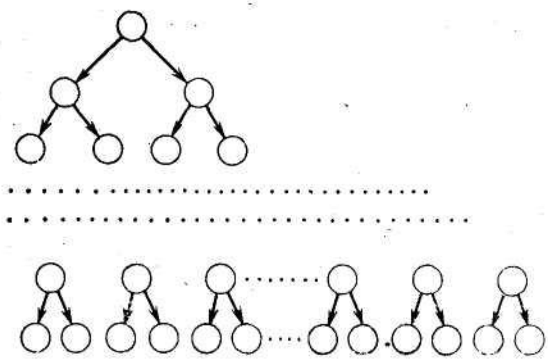

Рис. 3.57.

Узлам этого дерева соответствуют логические операторы консультационного алгоритма, статистика работы которых по различным ветвям соответствует содержанию входной информации о состоянии внешней среды и системы и определяется априорным распределением вероятностей реализации различных вариантов входных воздействий, т. е. элементов алфавита **А**. Каждой дуге, соединяющей два каких-либо смежных узла, в соответствии с содержанием входной информации может быть приписана вероятность выполнения консультационного алгоритма именно по этой дуге. Понятно, что эта вероятность может в случае необходимости учитывать надежность элементов системы, реализующих выполнение консультационного алгоритма по выбранной дуге. Множество оконечных узлов соответствует множеству элементов алфавита **Ф**.

Если консультационный алгоритм не имеет циклов и пересечений, каждому оконечному узлу дерева, очевидно, соответствует одна и только одна ветвь, соединяющая этот узел с начальным узлом дерева. Вероятность попадания в этот конечный узел поэтому может быть определена как произведение вероятностей прохождения дуг, образующих выбранную ветвь. Однако описанная процедура не может быть реализована, если дерево, соответствующее консультационному алгоритму, имеет циклы и пересечения. Так, например, прямой подсчет распределения вероятностей попадания в оконечные узлы затруднен для консультационного алгоритма, дерево которого изображено на рис. 3.58.

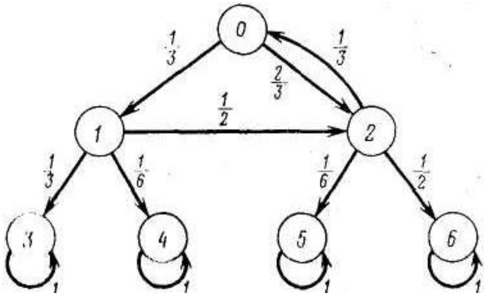

Рис. 3.58

Распределение вероятностей реализации различных исходов работы консультационного алгоритма для любой системы в принципе может быть получено непосредственным статистическим моделированием консультационного алгоритма функционирования системы. Однако вычислительные трудности ограничивают возможности использования такого подхода для оценки эффективности реальных консультируемых проблем.

Рассмотрим в связи с этим аналитический метод отыскания закона распределения вероятностей попадания в оконечные узлы для консультационных алгоритмов, дерево которых имеет произвольную (в смысле наличия циклов и пересечений и их количества) структуру.

В общем случае (для получения распределения различных реакций алгоритма на входные воздействия) представим консультационный алгоритм как реализацию некоторого дискретного марковского процесса, т. е. в виде простой марковской цепи. При этом множеству состояний цепи поставим в соответствие множество узлов дерева консультационного алгоритма, а множеству вероятностей перехода из одного состояния в другое — множество вероятностей прохождения соответствующих дуг. Поскольку оконечные узлы дерева консультационного алгоритма соответствуют поглощающим состояниям цепи, каждому из этих состояний необходимо приписать вероятность перехода в самих себя, равную единице.

Таким образом, консультационному алгоритму функционирования системы может быть поставлена в соответствии некоторая неэргодическая марковская цепь, изучение поведения которой может быть проведено методами п. 3.5.7.

Поскольку начальное состояние консультационного алгоритма известно заранее, интерес представляет лишь одна строка матрицы **B**, номер которой равен номеру исходного состояния. Приведем без пояснений результат решения задачи по отысканию матрицы **B** для консультационного алгоритма, граф которой изображен на рис. 3.58:

$$\mathbf{B} = \begin{vmatrix} 0 & 0 & 0 & 2/13 & 1/13 & 5/26 & 15/26 \\ 0 & 0 & 0 & 14/39 & 7/39 & 3/26 & 9/26 \\ 0 & 0 & 0 & 2/39 & 1/39 & 3/13 & 9/17 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}.$$

Знание распределения вероятностей реализации различных реакций консультационного алгоритма может быть использовано для оценки эффективности системы. Итак, пусть вектор **P** представляет собой закон распределения вероятностей различных реакций системы на внешние воздействии. При наличии модели системы нетрудно оценить эффективность системы для каждой из реакций консультационного алгоритма. Обозначим соответствующее множество оценок через

$$R = \{r_1, r_2, \ldots, r_m\}.$$

Тогда легко рассчитать $M[r]$ —математическое ожидание эффективности системы на всем множестве входных воздействий

$$M[r]=\sum_{i=1}^{m} r_i P_i. \tag{3.65}$$

Знание компонент вектора **Р** позволяет определить и другие оценки эффективности системы, например вероятность того, что эффективность системы находится в заданных пределах:

$$\text{Вер}\{r_{min}\leqslant r\leqslant r_{max}\}=\sum_{i\in I_0} P_i,$$
$$I_0=\{i: i\in[1,m],\quad r_{min}\leqslant r\leqslant r_{max}\}, \tag{3.66}$$

или вероятность того, что эффективность системы не ниже заданной:

$$\text{Вер}\{r_{\text{зад}}\leqslant r\}=\sum_{i\in I_1} P_i,$$
$$I_1=\{i: i\in[1,m],\quad r_{\text{зад}}\leqslant r_i\}, \tag{3.67}$$

или дисперсию случайного значения эффективности системы

$$Д[r]=\sum_{i=1}^{m}(r_i-M[r])^2 P_i=\sum_{i=1}^{n} r_i^2 P_i-\left(\sum_{i=1}^{n} r_i P_i\right)^2.$$

3.6. Введение в системы массового обслуживания

Функционирование многих консультационных систем носит характер обслуживания поступающих в систему заявок на формирование рекомендаций. Для выполнения совокупности действий или операций, подразумеваемых под понятием «обслуживание», в системе имеются специальные *каналы* или *линий.* Например, в телефонии заявками являются вызовы, возникающие на АТС в момент снятия абонентом телефонной трубки, а обслуживанием — предоставление линии, занимаемой абонентом на время разговора; на бензозаправочной станции в качестве заявок выступают автомобили, прибывающие на заправку, а каналами являются заправочные колонки. Аналогичные ситуации наблюдаются в системах посадки самолетов, разгрузки судов, в парикмахерских, магазинах и т. д.

Изучение обслуживания отдельной заявки сводится, как правило, к выяснению того, произошло ли событие, понимаемое под словом «обслуживание», к определению длительности обслуживания (или времени занятости обслуживающего канала), оценке качества обслуживания и т. д. При обслуживании потока заявок — совокупности заявок со специальным законом чередования их во времени — возникают дополнительные задачи: определение доли обслуженных заявок и доли заявок, получивших отказ, определение относительного времени занятости и простоя каналов и др.

Если с точки зрения содержания обслуживания все заявки равноправны и играет роль лишь сам факт поступления или непоступления заявки в данный момент времени, поток называется потоком однородных событий. Каждая заявка в этом случае характеризуется моментом t_j поступления ее в систему, а поток последовательностью моментов t_1, t_2, ..., t_k, … или законом, определяющим чередование моментов t_j.

В практике консультирования важную роль играют случайные потоки заявок (консультруемых проблем). Чтобы задать случайный поток, достаточно указать совместный (многомерный) закон распределения случайных величин t_1, t_2, ..., t_k, ... Для удобства часто вместо величин t_j рассматриваются интервалы ζ_j между последовательными заявками:

$$\begin{aligned} t_1 &= \zeta_1, \\ t_2 &= \zeta_1 + \zeta_2, \\ &\ldots\ldots \\ t_k &= \zeta_1 + \zeta_2 + \ldots + \zeta_k, \\ &\ldots\ldots\ldots\ldots \end{aligned}$$

Описание потока при помощи интервалов ζ_j эквивалентно заданию t_j; в самом деле, зная распределение ζ_j, можно получить распределение t_j, и наоборот.

Закон распределения случайных величин ζ_j задают в виде совместных функций распределения при всевозможных значениях $k \geq 1$

$$F(z_1, z_2, \ldots, z_k) = \mathbf{P}\{\zeta_1 < z_1, \zeta_2 < z_2, \ldots, \zeta_k < z_k\}.$$

Обычно рассматриваются только непрерывные случайные величины ζ_j, описываемые функциями плотности $f(z_1, z_2, ..., z_k)$.

Известно, что оперирование над многомерными распределениями отличается исключительной громоздкостью. Математическое описание оказывается более простым для некоторых важных классов потоков однородных событий.

Если совместная функция плотности может быть представлена в виде

$$f(z_1, z_2, \ldots, z_k) = f_1(z_1) f_2(z_2) \ldots f_k(z_k) \tag{3.65}$$

(случайные величины ζ_j - независимы), соответствующий поток называется потоком с *ограниченным последействием.* Если, кроме того, ζ_2, ζ_3 … распределены одинаково, поток называется *рекуррентным.* Поток называется *ординарным,* если $\lim\limits_{t\to 0}[\psi(t_0,t)/t]=0$ при любом t_0, где $\psi(t_0, t)$ — вероятность появления двух или более заявок в интервале времени (t_0, t_0+t).

Пусть $\vec{k} = (k_1, \ldots, k_n)$ — любой целочисленный вектор, $\vec{t} = (t_1, t_2,\ldots, t_n)$, где $0< t_1< t_2<\ldots< t_n$. Определим $p_{\vec{k}}(t_0, \vec{t})$ как вероятность появления k_1 событий в интервале (t_0, t_0+t_1),…, k_n событий в интервале (t_{n-1}, t_n).

Если $p_{\vec{k}}(t_0, \vec{t})$ не зависит от t_0, а определяется только величинами $\vec{t}$ и $\vec{k}$, то поток называется *стационарным.* Для стационарных потоков справедливо равенство

$$f_2(z) = f_3(z) = \ldots = f_k(z) = f(z). \tag{3.66}$$

Математическое ожидание

$$\mathbf{M}\zeta_j = \int_0^\infty z f(z)\,dz = m \qquad (j>1) \tag{3.67}$$

представляет собой среднее значение длительности интервала времени между последовательными заявками. Для ординарного стационарного потока величина

$$\lambda = 1/m \tag{3.68}$$

называется *интенсивностью* потока и выражает среднее число заявок, поступающих в единицу времени.

Для стационарного потока с ограниченным последействием имеет место формула Пальма

$$f_1(z_1) = \lambda\left[1 - \int_0^{z_1} f(z)\,dz\right], \tag{3.69}$$

позволяющая найти распределение интервала ζ_1, если известно распределение интервалов ζ_j при $j > 1$.

Рассмотрим в качестве примера стационарный ординарный поток с ограниченным последействием, имеющий *равномерное* распределение интервалов времени между заявками

$$f(z) = \begin{cases} 1/b, & z \in (0,\ b), \\ 0, & z \overline{\in} (0,\ b). \end{cases} \tag{3.70}$$

Тогда

$$\mathbf{M}\zeta = b/2, \qquad \lambda = 2/b.$$

По формуле Пальма

$$\frac{2}{b}\left(1-\int_0^{z_1}\frac{dz}{b}\right)=\frac{2(b-z_1)}{b^2}=\lambda\left(1-\frac{\lambda}{2}z_1\right).$$

Таким образом,

$$f_1(z_1)=\begin{cases}\lambda\left(1-\dfrac{\lambda}{2}z_1\right), & z_1\in(0,\ b),\\ 0, & z_1\overline{\in}(0,\ b).\end{cases} \qquad (3.71)$$

Легко видеть, что

$$\mathbf{M}\zeta_1 = b/3.$$

Если вероятность $p_k\ (t_0,\ t)$ поступления k заявок в интервале $(t_0,\ t_0+t)$ не зависит от чередования событий до момента t_0, т. е. если условная вероятность $p_k\ (t_0,\ t)$, вычисленная при любом предположении о чередовании событий до момента t_0, равна безусловной вероятности того же события, поток называется потоком *без последействия.*

Можно показать, что единственным стационарным ординарным потоком без последействия является так называемый *простейший* поток или поток Пуассона, для которого

$$p_k(t_0,\ t)=\frac{(\lambda t)^k}{k!}\mathrm{e}^{-\lambda t} \qquad (3.72)$$

или

$$f(z)=\lambda\mathrm{e}^{-\lambda z}. \qquad (3.73)$$

В соответствии с формулой Пальма можно убедиться, что для простейшего потока

$$f_1(z)=\lambda\mathrm{e}^{-\lambda z}=f(z).$$

В общем случае такое соотношение несправедливо [см., например, (3.71)].

Пусть обслуживающая система (консультационная система) состоит из n каналов, одновременно и независимо друг от друга обслуживающих заявки (проблемы). Канал может находиться в одном из двух состояний: свободном и занятом. Заявка, поступившая в систему в момент t_j, либо принимается к обслуживанию (если имеется свободный канал), либо остается в системе в течение некоторого времени γ (если все каналы заняты). Не позднее чем в момент $t_j+\gamma$

заявка должна быть принята к обслуживанию; в противном случае она получает отказ и покидает систему. Обычно рассматривают три класса систем: системы с отказами ($\gamma = 0$), системы с ожиданием ($\gamma = \infty$) и смешанные системы ($0 < \gamma < \infty$).

Для характеристики канала используется величина η — время занятости канала (длительность обслуживания). В общем случае γ и η случайные величины с заданными законами распределения. Могут быть сделаны различные предположения относительно порядка привлечения каналов к обслуживанию заявок (в порядке номеров каналов, в порядке очередности освобождения, в случайном порядке и т. д.), а также приема заявок на обслуживание (в порядке очереди, по минимальному времени, оставшемуся до получения отказа, в случайном порядке и т. д.).

Показателями эффективности систем массового обслуживания служат: 1) для систем с отказами — средняя доля отказов, вероятность обслужить все заявки в течение заданного интервала времени, 2) для систем с ожиданием — среднее время ожидания, средняя длина очереди и др., 3) для смешанных систем — все перечисленные величины.

В приводимых ниже примерах рассматриваются и более сложные системы массового обслуживания.

Исследование систем массового обслуживания может быть выполнено аналитически при некоторых (иногда весьма стеснительных с практической точки зрения) предположениях относительного характера потока заявок и свойств системы обслуживания. В качестве примера приведем наиболее элементарный случай. В n-канальную систему с отказами поступает простейший поток заявок с интенсивностью λ. Длительность обслуживания — показательно распределенная случайная величина с конечным математическим ожиданием $\bar{\eta}$. Заявки принимаются к обслуживанию в порядке очереди, а каналы привлекаются в порядке их номеров. Тогда, при $t \to \infty$ значение средней доли отказов (в данном случае, в силу стационарности процесса, она также имеет смысл вероятности отказа для заявки, поступившей в наудачу выбранный момент времени) будет равно

$$R = \frac{(\lambda\bar{\eta})^n}{n!} \Big/ \sum_{k=0}^{n} \frac{(\lambda\bar{\eta})^k}{k!} \quad \text{(формула Эрланга).}$$

Заметим, что данная формула, одна из сравнительно немногих, справедлива и в случае произвольного распределения времени обслуживания.

Для исследования систем массового обслуживания используется также метод статистического моделирования (Монте-Карло). Ниже мы рассмотрим некоторые типичные примеры математического описания процессов функционирования систем массового обслуживания. Здесь весьма эффективным оказывается использование аппарата марковских процессов.

Пример 1. Пусть имеется n-линейная консультационная система массового формирования рекомендаций, а именно: система в которой формируются рекомендации по решению консультационных задач консультируемой проблемы, т.е. система состоит из n параллельно работающих устройств (средств, участвующих в формировании рекомендаций). В систему поступает рекуррентный поток требований по формированию рекомендаций, т. е. интервалы между последовательно поступающими требованиями являются независимыми одинаково распределенными случайными величинами [функция распределения $A(x)$]. Все устройства работают независимо. Время выполнения требования на формирование рекомендации на любом устойстве— случайная величина с заданной функцией распределения $B(x)$. Требование на формирование рекомендации, заставшее все n устройства занятыми, становится в общую очередь; формирование рекомендации производится в порядке поступления требований на формирование рекомендации.

Обозначим через $w(t)$ «виртуальное время ожидания» в момент t, определяемое следующим образом. Если бы в момент t в консультационную систему поступило требование на формирование рекомендации, постороннее по отношению к входящему потоку, и присоединилось к общей очереди формирования рекомендации, то ее время ожидания составляло бы $w(t)$. Иначе говоря, $w(t)$ — случайная величина, равная длительности промежутка времени от момента t до момента освобождения хотя бы одного из устройств от требований, т.е. от сформированных рекомендаций, имевшихся в системе в момент времени t. Именно, если обозначить через $v(t)$ число требований по формированию рекомендации, находящихся в системе в момент t, то

$$w(t) = \begin{cases} 0, & \text{если } \nu(t) < n, \\ \inf\{\tau : \tau > t, \ \ \nu(\tau) < n\} - t, & \text{если } \nu(t) \geqslant n. \end{cases}$$

Ясно, что $w(t)$ в общем случае не является марковским консультационным процессом (исключение будет иметь место в случае n = 1, если, к тому же, входящий поток требований на формирование рекомендации — пуассоновский).

Введем следующие случайные величины: z_i *(i)*, $1 \le i \le n$, — время с момента *t* до момента окончания формирования рекомендации *j*-м устройством требований, поступивших ранее момента *t* (если *i* -е устройство в данный момент времени свободно, полагаем $z_i(t) = 0$);
z_{n+1} *(t)* — время с момента *t* до поступления в систему очередного требования на формирование рекомендации, $(n + 1)$-мерный случайный консультационный процесс

$$z(t) = (z_1(t),\quad z_2(t),\ \ldots,\ z_n(t),\ z_{n+1}(t))$$

будет марковским, что непосредственно очевидно. Траектория этого процесса ведет себя следующим образом: $z_{n+i}(t)$ убывает с единичной скоростью до того момента времени, когда она обращается в нуль; в этот момент $z_{n+1}(t)$ принимает случайное значение, независимое от течения консультационного процесса и равное следующему интервалу между поступлением требований на формирование рекомендации, и снова начинается убывание с единичной скоростью.

Все координаты $z_1(t)$, $z_2(t)$.....$z_n(t)$ убывают с единичной скоростью до достижения нулевого значения. Если какая-нибудь из них, скажем $z_i(t) = 0$, обратилась в нуль, дальнейшее убывание прекращается, и начиная с этого момента $z_i(t) = 0$. Так происходит до тех пор, пока z_{n+1} *(t)* не обратится в нуль. В этот момент общая загрузка консультационной системы увеличивается на необходимую случайную длительность формирования рекомендации η с функцией распределения *B(x)* вновь поступившего требования на формирование рекомендации. Но как изменятся координаты $z_i(t)$? Очевидно, новое требование на формирование рекомендации поступит туда, где придется меньше ожидать. Именно, пусть

$$z_{i_0}(t) = \min_{1 \leqslant j \leqslant n} z_j(t).$$

Тогда

$$z_j(t+0) = z_j(t), \qquad j \neq i_0, \quad j \leqslant n,$$

$$z_{i_0}(t+0) = z_{i_0}(t) + \eta.$$

В случае, если минимум достигается на двух или нескольких значениях *j*, можно условиться, что поступающее требование на формирование рекомендации направляется на устройство, скажем, с наименьшим номером.

Очевидно, зная значение консультационного процесса *z(t)*, можно найти интересующее нас значение *w(t)*:

$$w(t) = \min_{1 \leqslant j \leqslant n} z_j(t).$$

Пример 2. Рассмотрим ту же самую консультационную систему массового формирования рекомендаций по решению консультационных задач консультируемой проблемы, но, так сказать, более подробно, интересуясь не только загрузкой устройств, но также числом находящихся в очереди требований на формирование рекомендации.

Пусть *v(f)* — число требований на формирование рекомендации в консультационной системе в момент *t.* При $v(t) \leq n$ выполняются все имеющиеся требования на формирование рекомендации; при $v(t) > n$ в очереди находится $v(t) - n$ требований на формирование рекомендации. Введем случайный консультационный процесс

$$z(t) = (z_0(t),\ z_1(t),\ \ldots,\ z_{|z_0(t)|}(t)),$$

где

$$|z_0(t)| = \begin{cases} z_0(t) + 1 & \text{при } z_0(t) \leqslant n, \\ n + 1 & \text{при } z_0(t) > n, \end{cases} \tag{4.74}$$

т. е. $|z_0(t)| - 1$ — число занятых устройств в момент *t* (величину $|z_0(t)|$ иногда называют рангом состояния $z_0(t)$);

$z_k(t)$, $k \geq 2$, — время с момента *t* до момента окончания формирования рекомендации $(k - 1)$-м устройством, которое участвует в формировании рекомендации (в случае $|z_0(t)| = 1$ координаты $z_k(t)$, $k \geq 2$, не определяются);

$z_1(t)$ — время с момента *t* до момента поступления очередного требования на формирование рекомендации.

Введенные функции времени $z_1(t)$, $z_2(t)$, ... убывают с единичной скоростью, пока какая-нибудь из них не обратится в нуль. Если это была $z_1(t)$, новым значением ее будет случайная величина, равная очередному интервалу между поступлением требований на формирование рекомендации; если в нуль обратилась одна из $z_2(t)$, $z_3(t)$ ...., то $z_0(t)$ уменьшается на 1; если до этого было $z_0(t) > n$, обратившаяся в нуль $z_i(t)$заменяется случайной величиной, равной длительности формирования рекомендации, взятой из очереди; если же $z_0(t) \leq n$, новые значения $z'_i(t)$ координат определяются равенствами

$$z'(t) = z_j(t), \quad j < i; \quad z'_j(t) = z_{j+1}(t), \quad j \geqslant i, \tag{3.75}$$

т. е. обратившаяся в нуль координата вектора *z(t)* вычеркивается. При таком подходе необходимость выбора минимальной координаты отпадает; в то же время вместо консультационного процесса фиксированной размерности мы получили консультационный процесс переменной размерности, зависящей от числа занятых устройств.

Легко видеть, что этот консультационный процесс является марковским.

Пример 3. Рассмотрим ту же систему, но с учетом возможности выхода из рабочего состояния и восстановления приборов, которые участвуют в формировании рекомендации. В момент $t = 0$ все приборы являются исправными; задается распределение времени безотказной работы каждого из них. Если прибор вышел из строя, он поступает в j восстанавливающую систему. Это, в свою очередь, m-линейная система массового обслуживания с ожиданием ($1 \leq m \leq n$). Рассмотрим вариант, при котором консультируемая проблема жестко закрепляется за прибором, т. е. в случае отказа последнего ожидает его восстановления. Если необходимое время обслуживания (формирования рекомендации) равно η, причем до отказа прибора требование (консультируемая проблема) обслуживалось на протяжении времени x, то после восстановления необходимое время обслуживания будет составлять η —x.

Функционирование системы описывается марковским процессом

$$z(t) = (z_0(t), z_1(t), \ldots, z_{|z_0(t)|}(t)),$$

где $z_0(t) = (z_{00}(t), z_{01}(t), z_{02}(t), \ldots, z_{02n}(t)$ — многомерный дискретный параметр, указывающий состояние всех приборов и число требований в системе при $1 \leq i \leq n$;

$$z_{0i}(t) = \begin{cases} 0, & \text{если } i\text{-й прибор в момент } t \text{ исправен,} \\ 1, & \text{если } i\text{-й прибор в момент } t \text{ восстанавливается,} \\ j+1, & \text{если } i\text{-й прибор в момент } t \text{ находится в состоянии ожидания ремонта, причем его место в очереди } j\text{-е;} \end{cases}$$

$$z_{0,n+i}(t) = \begin{cases} 0, & \text{если } i\text{-й прибор свободен от требований в момент } t, \\ 1, & \text{если } i\text{-й прибор обслуживает требование в момент } t, \\ 2, & \text{если } i\text{-й прибор находится в момент } t \text{ в неисправном состоянии, причем в момент отказа он обслуживал некоторое требование;} \end{cases}$$

$z_{00}(t)$ — число требований в системе в момент t;

$z_1(t)$ — время до поступления очередного требования;

$z_{i+1}(t)$, $1 \leq i \leq n$, — время до отказа i -го прибора, если $z_{0i}(t) = 0$, время до окончания восстановления i-го прибора, если $z_{0i}(t) = 1$; время, необходимое для восстановления 1-го прибора, если $z_{0i}(t) = 2$;

$z_{n+i+1}(t)$, $1 \leq i \leq \min(n, z_{00}((t)$, — время до окончания обслуживания i-го требования из числа требований, которые либо обслуживаются в момент i, либо обслуживались ранее и в момент t не обслуживаются из-за неисправности прибора (требованию, обслуживаемому прибором с меньшим номером, приписывается меньший номер);

$|z_0(t)| = n + \min\{n, z_{00}(t)\} + 1$ —число непрерывно меняющихся координат.

До тех пор, пока все $z_i(t)$, $i\geq 1$, положительны, они изменяются по линейному закону. При этом $z_{i+1}(t)$, $1 \leq i \leq n$, убывают с единичной скоростью при i, для которых $z_{0i} \leq 1$, и остаются неизменными при остальных i; что касается $z_i(t)$, $i\geq n+1$, то те из них, которые соответствуют времени до окончания обслуживания действительно обслуживаемых требований, убывают с единичной скоростью; остальные же остаются постоянными.

При обращении $z_1(t)$ в нуль новым значением этой переменной будет время до поступления следующего требования; $z_{00}(t)$ получает приращение 1. Если все $z_{0,\,n+i}(t)$, $i\geq 1$, $1 \leq i \leq n$, остальные координаты останутся неизменными; в противном случае находится l как наименьшее i', для которого $z_{0,\,n+i'}(t)=0$. Это будет номер того прибора, на который посылается требование, поступившее в момент t. В этом случае новые значения $z'_i(t)$ координат вектора $z(t)$ определяются следующим образом:

$$z'_{00}(t)=z_{00}(t)+1;$$

$$\left.\begin{aligned} &z'_{0,n+l}(t)=1;\\ &z'_{0i}(t)=z_{0i}(t), \quad i\neq n+l;\\ &z'_{i+1}(t)=z_{i+1}(t), \quad 1\leqslant i\leqslant n; \end{aligned}\right\} \tag{3.76}$$

координата $z'_{n+i+1}(t)$, $1 \leq i \leq \min\{n, z_{00}(t)\}$, образуется так: пусть t' определено равенством

$$l'=\sum_{i=2}^{l-1} z_{0,\,n+i},$$

тогда

$$z'_{n+i+1}(t)=z_{n+i+1}(t), \quad 1\leqslant i\leqslant l'; \tag{3.77}$$

$z'_{n+l+2}=\eta$, где η — длительность обслуживания поступившего требования;

$$z'_{n+i+2}=z_{n+i+1}(t), \quad i>l. \tag{3.78}$$

В следующем примере будет рассматриваться однолинейная консультационная система массового формирования рекомендаций по решению консультационных задач консультируемой проблемы; методика рассмотрения многолинейных консультационной систем должна быть ясна из предыдущего.

Пример 4. В консультационную система массового формирования рекомендаций по решению консультационных задач консультируемой проблемы поступает рекуррентный поток требований на формирование рекомендаций. Процесс формирования рекомендаций выполняется в порядке очереди поступивших требований на формирование рекомендаций. С каждым требованием связаны две случайные величины η и γ, η — длительность формирования рекомендации; γ— характеристика ограничения, имеющая следующий смысл. Если на протяжении времени γ требование на формирование рекомендаций не будет взято на выполнение процесса формирования рекомендации, оно покинет систему. В противном случае, если требование взято в работу через время x после поступления, оно может ожидать окончания формирования рекомендации не более $a(\gamma - x)$ единиц времени, где $a > 0$ — некоторая постоянная (допускается и случай $a = \infty$). Если за это время требования не будет выполнено, оно покинет систему.

В эту схему включаются в качестве частных случаев система с неограниченным ожиданием ($\gamma = \infty$), система с ограниченным временем ожидания начала работы ($a = \infty$), система с ограниченным временем пребывания в ней требований ($a = 1$).

Пусть $z_0(t)$ обозначает число требованийс находящихся в системе в момент t. Введем (при $z_0(t) \geq 1$) следующие переменные; $z_1(t)$ — время с момента t до момента поступления следующего требования; $z_{i+1}(t)$, $1 \leq i \leq z_0(t)$, —оставшееся время обслуживания i-го требования (требование, обслуживаемое в данный момент времени, считается первым); $z_{z_0(t)+i+1}(t)$, $1 \leq i \leq z_0(t)$,— оставшийся к моменту t запас ожидания. Начальным запасом ожидания для поступающего требования будет γ; пока требование ожидает в очереди, запас ожидания убывает с единичной скоростью, когда требование будет взято на обслуживание, запас ожидания будет убывать со скоростью $1/a$. Многомерный случайный процесс

$$z(t) = (z_0(t), z_1(t), \ldots, z_{2z_0(t)+1}(t))$$

будет марковским.

Пусть в момент t поступает новое требование. Обозначив через $z_i(t)$ и $z'_i(t)$ соответственно старые и новые значения координат вектора $z(t)$, получим:

$$z'_0(t) = z_0(t) + 1;$$

$z'_1 = \zeta$, (времени до поступления следующего требования):

$$z'_{i+1}(t) = z_{i+1}(t), \quad 1 \leqslant i \leqslant z_0(t);$$

$z_{z_0(t)+2}(t) = \eta$ (временя обслуживания вновь поступившего требования):

$$z'_{z_0(t)+i+2}(t) = z_{z_0(t)+i+1}(t), \quad 1 \leqslant i \leqslant z_0(t);$$

$z'_{2,\ z0(t)+3}(t) = \gamma$ (запасу ожидания вновь поступившего требования).

Если обратится в нуль координата $z_{i+1}(t)$, $1 \le i \le z_0(t)$, либо $z_{z_0(t)+i+1}(t)$, $1 \le i \le z_0(t)$, то $z_0(t)$ уменьшается на 1, а обе координаты $z_{i+1}(t)$ и $z_{z_0(t)+i+1}(t)$ вычеркиваются из записи вектора $z(t)$.

В интервалах времени, когда все $z_i(t)$, $i \ge 1$, положительны, $z_1(t)$ и $z_2(t)$ убывают с единичной скоростью; $z_3(t)$, ..., $z_{z0(t)+1}(t)$ остаются постоянными; $z_{Zo(t)+2}(t)$ убывает со скоростью $1/a$; $z_{z_0(t)+i+1}(t)$, $2 \leqslant i \leqslant z_0(t)$, убывают с единичной скоростью.

Пример 5. Пусть имеется средство формирования рекомендаций, способное одновременно выполнять процесс формирования рекомендаций для сколько угодно поступающих требований по формированию рекомендаций. Пусть η — величина работы, которую нужно выполнить для формирования одной рекомендации. Будем считать, что η — случайная величина, причем соответствующие величины для различных требований независимы в совокупности и одинаково распределены. При условии, что в системе находятся n требований (n = 1, 2, 3, ...), темп формирования рекомендаций каждой из них составляет α_n. Это означает, что оставшаяся работа убывает со скоростью α_n. Входящий поток требований на выполнение работы по формированию рекомендации, как и ранее, будем считать рекуррентным. Для описания подобной консультационной системы введем случайный консультационный процесс

$$z(t) = (z_0(t),\ z_1(t),\ z_2(t),\ \ldots,\ z_{|z_0(t)|}(t)),$$

где $z_0(t)$ — число требований в системе в момент t; $z_1(t)$ — время до поступления очередного требования; $z_{i+1}(t)$— оставшаяся величина работы 1-го требования по формированию рекомендации;

$|z_0(t)| = z_0(t) + 1$.

Из принятых условий следует, что $z_i'(t) = -\alpha_{z0(t)}$, $i \ge 1$. При обращении некоторой из $z_{i+1}(t)$ в нуль она просто вычеркивается из записи вектора $z(t)$, а $z_0(t)$ уменьшается на 1. При обращении в нуль переменной $z_1(t)$, что означает поступление нового требования, новым значением этой переменной будет случайная величина ζ, равная времени до поступления следующего требования.

Легко установить, что процесс $z(t)$ является марковским.

Пример 6. Имеется однолинейная консультационная система с ожиданием, в которую поступают два независимых рекуррентных потока требований на формирование рекомендаций. В момент поступления требования из первого потока выполнение требования из второго потока прерывается и при возобновлении предыдущее время выполнения теряется (вновь реализуется случайное время формирования рекомендации). Функционирование подобной консультационной системы описывается марковским процессом $z(t) = (z_0(t), z_1(t).....)$, где $z_0(t)= (z_{01}(t), z_{02}(t)); z_{0i}(t)$— число находящихся в системе в момент t требований из i-го потока (i = 1,2); $z_i(t)$ — время с момента t до поступления в систему очередного требования i-го потока (i = 1,2); $z_3(t)$ — время до окончания формирования рекомендации, средством, формрующим рекомендации, в момент t, если таковое имеется. В теории консалтинга будем пользоваться описанием консультционных систем массового формрования рекомендаций при помощи высказывательных функций. Для решения многих задач, связанных с исследованием консультционных систем массового формрования рекомендаций, можно обойтись описанием лишь небольшого числа так называемых *особых* состояний. К ним обычно относятся состояния системы в моменты поступления нового требования на формрование рекомендации, начала и окончания формирования очередной рекомендации, состояние, при котором некоторое требование на формирование рекомендация покидает систему, исчерпав полностою свое время ожидания, и т. д. Достижение системой каждого особого состояния понимается как наступление одного из заранее предусмотренных *событий,* а процесс ее функционирования — как чередование этих событий во времени.

Для описания события можно использовать высказывательную функцию, реализующую соотношения между моментами достижения, системой соответствующих особых состояний (принимающую значение 1, когда данное событие наступает, и значение 0 — в противном случае). Высказывательные функции, относящиеся к отдельным событиям, связываются операциями булевой алгебры таким образом, чтобы полученная высказывательная функция описывала последовательность событий в процессе функционирования системы. Такое представление процесса функционирования системы может быть использовано для формального построения алгоритмов его моделирования на ЭВМ.

4. Консультационные процессы формирования оптимальных рекомендаций

4.1. Основные компоненты консультационных процессов формирования рекомендаций

На основе представления консультационного процесса как задачи формирования наиболее обоснованных (оптимальных) рекомендаций ниже будут выявлены инвариантные компоненты этого процесса. Данные компоненты с учетом особенностей их формирования являются исходной информацией для определения структуры автоматизированного формирования рекомендаций на базе консультационных модулей.

Рассматриваемый (рис. 4.1) консультационный процесс формирования рекомендаций является частью жизненного цикла формируемых рекомендаций, занимая место между процессом формирования консультационного задания (КЗ) на формирование рекомендаций и реализацией их заказчиком (клиентом).

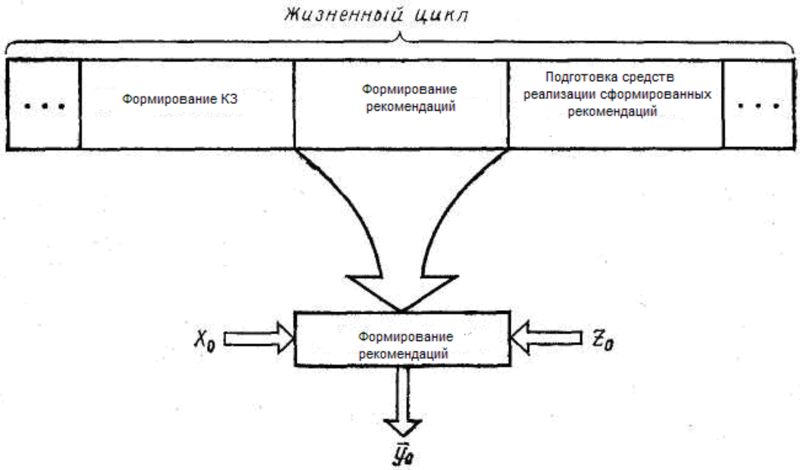

Рис. 4.1. Выделение процесса формирования рекомендаций из жизненного цикла (X_0 – консультационное задание на формирование рекомендаций; Z_0 – прогноз условий реализации сформированных рекомендаций; $\bar{y}_0$ - рациональный вариант сформированных рекомендаций)

При автономном рассмотрении формирования рекомендаций как элемента жизненного цикла необходимо, следуя известным принципам системного анализа, заменить его связи с другими элементами этого

цикла соответствующими реакциями, обычно называемыми внешними условиями. Такие реакции при выделении консультационногопроцесса формирования рекомендаций из жизненного цикла определяются, с одной стороны, консультационным заданием на формирование рекомендаций решения задач консультируемой проблемы, а с другой — возможностями научно-технического потенциала, который может быть использован при реализации сформированных рекомендаций. Консультационное задание на формирование рекомендаций содержит директивную информацию, которую далее будем считать представленной в виде некоторого множества X_0 с координатами
$x_0 = \{x_{01}, x_{02}, ..., x_{0n}\}$. Возможности научно-технического потенциала определяются как текущим уровнем технологий, так и наличием соответствующих средств, обеспечивающих реализацию сформированных рекомендаций. Эти возможности будем представлять в виде некоторого множества Z_0 с координатами $z_0 = \{z_{01}, z_{02}, ..., z_{0k}\}$, которые далее рассматриваются как обобщенный показатель, отражающий влияние возможных способов формирования и реализации для того или иного варианта сформированных рекомендаций на их показатели.

Будем полагать, что практически каждая сформированная рекомендация может быть описана некоторым списком (вектором) «числовых» параметров y_0, часть которых может изменяться дискретно $(y^{д}_0)$, а другая часть — непрерывно $(y^{н}_0)$, т. е.

$$y_0=\{ y^{д}_0, y^{н}_0\}.$$

При этом числовой характер дискретных переменных не означает обязательной необходимости количественного измерения обозначаемых ими понятий. Например, если консультируемой проблемой является летательный аппарат (ЛА), то рекомендация по выполнению схемы соединения ступеней Л А (тандемная, пакетная) может определяться дискретной переменной, имеющей соответственно значения 1, 2. В качестве других примеров дискретных параметров, по которым формируются рекомендации в отношении создания облика ЛА, можно назвать число ступеней, тип двигательной установки по ступеням и т. д. Примерами непрерывных параметров ЛА, по которым формируются рекомендации их использования, могут служить стартовая масса, площадь крыла, относительные конечные массы ступеней, начальные тяговооруженности, давление в камере сгорания и т. д. При этом часть параметров, определяющих один вариант формируемых рекомендаций, может не присутствовать при определении другого. В частности, при формировании рекомендации по определению крылатого ЛА обязательно должны присутствовать, например, параметры крыла, которые отсутствуют в перечне параметров баллистического аппарата.

*Введем формальное понятие «**концепция сформированной рекомендации**», под которой будем понимать следующую тройку: совокупность «наименований» параметров, определяющих сформированную рекомендацию, т. е. состав вектора y_0; конкретные значения дискретных параметров* $<y^д_0>$ *(*здесь и далее через (a) обозначаются значения вектора *a); принятые множества возможных изменений непрерывных параметров* $\Delta y^н_0$.

Каждую i-ю концепцию сформированной рекомендации будем обозначать Ω (i), а множество рассматриваемых концепций —

$$\Omega = \{\Omega\ (i)\}_{i=1,\ 2.....\ N_Q}\ .$$

Тогда

$$\Omega\ (i)=\{\ y_0(i), <y^д_0(i)>,\ \Delta y^н_0\ (i)\}\ , \qquad (4.1)$$

где y_0 (i) — набор переменных, описывающих i-ю концепцию сформированной рекомендации.

При этом будем учитывать часто имеющую место зависимость

$$y^н_0\ (i)= y^н_0\ (<y^д_0(i)>),$$

Она может быть проиллюстрирована следующим примером. При различных значениях дискретного параметра ЛА «тип системы подачи топлива в камеру сгорания», например, вытеснительная система и система с турбонасосным агрегатом, каждая из этих типов систем описывается отличающимися переменными.

Множество вариантов сформированных рекомендаций i-й концепции, обозначаемое далее $\Im_0(i)$, может быть представлено в виде:

$$\Im_0(i)=\{\ <y^д_0(i)>, (<y^н_0(i)>\}_{\left\langle y^н_0(i)\right\rangle \in \Delta y^н_{0(i)}}$$

Используя понятие «концепция», далее будем считать, что рассматриваемые в консультационном процессе варианты формируемых рекомендаций могут отличаться концептуально, т. е. у них могут отличаться y_0, $\Delta y^н_0$,$<y^д_0>$, а также в рамках одной концепции, т. е. отличаться значениями $y^н_0$ в рамках определенного диапазона $\Delta y^н_0$. Необходимость такого разделения вызвана, в основном, невозможностью получения единого формального описания концептуально отличающихся формируемых рекомендаций, требуемого для их математического исследования. В результате математические модели, описывающие концептуально отличающиеся формируемые рекомендации, носят разрывный характер.

В то же время подавляющее большинство известных математических методов, используемых в консультационном процессе, эффективны лишь при наличии непрерывных моделей. В результате имеет место объективная необходимость проведения процесса анализа альтернативных вариантов сформированных рекомендаций в два этапа. На первом из них из каждой группы вариантов сформированных рекомендаций одной концепции с активным использованием формальных методов численного анализа и оптимизации выделяется *рациональный вариант*, а на втором — производится непосредственное *сравнение этих выделенных вариантов*.

Приведенная формализация понятия «концепция сформированных рекомендаций», которой будем следовать далее, не противоречит ее пониманию, понятому в настоящее время. Она позволяет в рамках единой формальной модели дать описания концептуально отличающихся вариантов сформированных рекомендаций и в то же время предоставляет формальные условия, разделяющие сформированные рекомендации, каждая из которых требует для своего автоматизированного рассмотрения свойственные только этой концепции математические модели и исходные данные.

Внешние условия (X_0, Z_0) при конкретной реализации консультационного процесса выступают в роли ограничений на формируемые рекомендации. Однако при известных ограничениях возможны, как правило, несколько вариантов сформированных рекомендаций, удовлетворяющих им, т. е. соответствующих заданному КЗ и реализуемых. В дальнейшем такие сформированные рекомендации будем называть допустимыми и обозначать множество таких сформированных рекомендаций через Y_0:

$$Y_0 = \{y_0 : V_0(x_0, y_0, z_0) \leq 0\},$$

где $V_0(x_0, y_0, z_0) \leq 0$— некоторое формальное условие допустимости сформированных рекомендаций y_0 при заданных X_0, Z_0.

В случае неединственности варианта сформированных рекомендаций появляется необходимость всесторонней и объективной оценки имеющихся сформированных рекомендаций, выбора лучших среди них, которые и должны быть переданы на следующий этап жизненного цикла, в данном случае — на этап реализации, начинающийся технологической подготовкой средств реализации.

Для оценки альтернативных вариантов сформированных рекомендаций, в общем случае, используется ряд показателей (критериев), каждый из которых определяет то или иное «качество» выполнения консультационных задач X_0, консультируемой проблемой y_0, на функционирование которой наложены ограничения, связанные с удовлетворением условиям Z_0. Исходя из этого, необходимо наличие правила F_0, ставящего в соответствие $\{x_0, z_0, y_o\}$ значения компонент вектора критериев, обозначаемого K_0, т. е.

$$K_0 = F_o \{x_0, z_0, y_o\} .$$

В случае векторного критерия предполагается, что существует некоторое правило (правило обоснования), позволяющее по значениям критериев на всем множестве Y_0 определить наиболее предпочтительный (лучший) вариант сформированных рекомендаций $\bar{y}_0$. Обозначим данное правило W_0.

Итак, в формализованном виде задача формирования рационального варианта сформированных рекомендаций в отношении консультируемых проблем (П) может быть представлена в виде следующего кортежа:

$$\Pi = <X_o, Z_o, \Omega_o, V_o, Y_o, K_{0)} F_o, W_o, \bar{y}_0), \qquad (4.2)$$

где X_0 — множество задач, возлагаемых на консультируемую проблему; Z_0 — условия реализуемости сформированных рекомендаций, учитывающие имеющиеся средства и достигнутый научно-технический уровень; Ω_0 — множество рассматриваемых концепций сформированных рекомендаций; V_0 — формальное правило, выделяющее из множества рассматриваемых вариантов сформированных рекомендаций допустимые варианты;

Y_0 — множество допустимых вариантов сформированных рекомендаций; K_0 — вектор критериев, оценивающих предпочтительность допустимых вариантов сформированных рекомендаций; F_0 — правило, ставящее в соответствие каждым

$\{x_0, z_0, y_0\}$ значения вектора критериев; W_0 — правило выбора наиболее обоснованного варианта сформированных рекомендаций из множества допустимых; $\bar{y}_0$ — рациональный вариант сформированных рекомендаций.

Структурная схема данного кортежа показана на рис. 4.2.

Рис. 4.2. Структура действий в процессе формирования рационального варианта рекомендаций

Из этой схемы видно, что она в формализованной форме охватывает основные этапы консультационной деятельности, связанной с формированием рационального варианта рекомендаций. Осуществление этих этапов в автоматизированном режиме является основой для формирования структуры консультационного процесса и формирования рекомендаций на базе консультационных модулей. В то же время реализация только этих этапов применительно к задаче формирования рекомендаций недостаточна вследствие чрезвычайно большой размерности вектора y_0, определяющего вариант сформированных рекомендаций в отношении сложных консультируемых проблем.

4.2. Процесс формирования рекомендаций как совокупность консультационных операций

Размерность вектора y_0, определяющего сложные консультируемые проблемы, чрезвычайно велика (например, для ЛА она характеризуется числом порядка 10^4—10^8), что делает непосредственное решение консультационной задачи формирования рекомендаций по выбору значений этого вектора нереальным. В связи с этим в практической деятельности при формировании рекомендаций для сложных консультируемых проблем сложился подход, согласно которому весь процесс формирования рекомендаций сводится к автономному решению частных консультационных задач (выполнению консультационных операций) с последующим согласованием получающихся результатов.

Проведем простейший анализ связи обоснованности варианта сформированных рекомендаций в целом с числом и масштабом консультационных операций, на которые расчленен консультационный процесс формирования рекомендаций для создания проекта изделия. Будем характеризовать консультационную операцию ее сложностью, степенью обоснованности и возможными потерями из-за недостаточной обоснованности рекомендации. Измерителем сложности может быть трудоемкость консультационной операции или косвенно выражающая ее характеристика.

Количественная оценка степени обоснованности наиболее естественным образом может быть построена путем формирования рекомендации по планированию минимальной дополнительной деятельности, которую следовало бы провести для практически полного устранения сомнений в том, что разработанный вариант сформированных рекомендаций является наилучшим. Трудоемкость этой предполагаемой деятельности характеризует степень обоснованности консультационной операции. Если она минимальна, то консультационная операция хорошо обоснована.

При таком подходе планирование отдельной консультационной операции основывается на первоначальной разработке «идеальной модели» этой операции, которая с минимальным превышением ограничений, составляющих условия проведения операции, позволила бы получить полностью обоснованную рекомендацию. Затем разрабатывается план проведения реальной консультационной операции с учетом всех ограничений на возможность ее проведения и как его «дополнение» до плана «идеальной модели» — план дополнительной деятельности. Трудоемкости реализации этих двух планов и являются двумя основными характеристиками консультационной операции. Для удобства можно нормировать степень обоснованности, относя трудоемкость дополнительной деятельности к общей трудоемкости реализации «идеальной модели».

Обозначим буквами $T_и$, T и $\bar{T}$ трудоемкости операции по реализации «идеальной модели», консультационной операции и дополнительной деятельности. Если S — степень обоснованности консультационной операции, то целесообразно принять

$$S = 1 - \frac{\bar{T}}{T_и}$$

откуда $0 \le S \le 1$.

Очевидно, $T + \bar{T} \ge T_и$. В противном случае деятельность по реализации «идеальной модели» было бы целесообразно построить как исходную консультационную операцию, вслед за которой выполняется дополнительная деятельность, так как трудоемкость этого процесса $T + \bar{T}$ была бы меньше, чем $T_и$.

Тогда

$$S \le T/T_и,\ S(0) = 0,\ S(T_и) = 1$$

и зависимость обоснованности консультационной операции от ее трудоемкости имеет характер, показанный на рис. 4.3 сплошной линией.

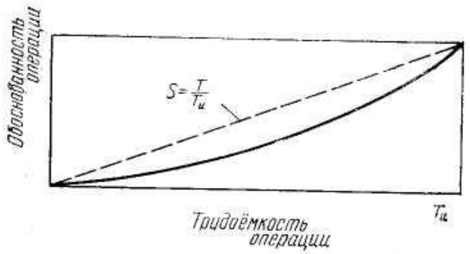

Рис. 4.3. Взаимозависитость обоснованности и трудоемкости консультационной операции

Естественно принять, что функция S (Г) вогнута и *S″ (T)* ≥0.

При рассмотрении нескольких консалтинговых операций в качестве единой консалтинговой операции их трудоемкости складываются, а степени обоснованности осредняются с весами, равными трудоемкости «идеальных моделей» рассматриваемых операций. При этом к объединяемым консультационным операциям следует добавить операцию управления их совместным функционированием. Данная операция характеризуется собственными трудоемкостью и степенью обоснованности, определяемой аналогично предыдущему через «идеальную модель» такой операции управления, при которой результат выполнения составной операции практически тождествен результату ее выполнения как единой консультационной операции. Тогда

$$T_\Sigma = \sum_{i=0}^{n} T_i; \quad S_\Sigma = \sum_{i=0}^{n} T_{u_i} S_i \Big/ \sum_{i=0}^{n} T_{u_i},$$

где *n* — число объединяемых консультационных операций, а индекс 0 соответствует операции управления. Допустим, что зависимости $S_i\ (T_i)$ подобны в отношении масштабного параметра $T/T_и$ (см. рис. 4.3), т. е.

$$S_i = S\ (T_i/T_и).$$

С учетом этого

$$T_\Sigma = \sum_{i=0}^{n} T_i; \quad S_\Sigma = \sum_{i=0}^{n} T_{u_i} S(T_i / T_u) \Big/ (T_{u_0} + \sum_{i=1}^{n} T_{u_i}).$$

Перейдем к рассмотрению задачи о рациональном уровне декомпозиции этого процесса на отдельные операции, т. е. о том, на сколько консультационных операций какого уровня сложности целесообразно его расчленить с тем, чтобы при заданных затратах получить максимально обоснованный вариант сформированных рекомендаций. Решение этой задачи состоит в максимизации $S_{\sum}$ при фиксированном значении $T_{\sum}$.

1. Будем полагать, что консультационные операции, на которые расчленяется консультационный процесс формирования рекомендаций, имеют примерно равный масштаб:

$$T_{\text{и}_i} \approx T_{\text{и}},\ i = 1,2, \dots,\ n$$

В этом случае

$$T_{\Sigma} = \sum_{i=0}^{n} T_i\ ,$$

$$S_{\Sigma}\,\frac{T_{u_0} S\left(\dfrac{T_0}{T_{u_0}}\right) + T_u \sum_{i=1}^{n} S\left(\dfrac{T_i}{T_u}\right)}{T_{u_0} + T_u n}$$

и оптимизация значений T_i при фиксированной общей трудоемкости $T_{\sum}$ для получения максимальной степени обоснованности S_{Σ}, очевидно, приводит к $T_i = T,\ i = 1, 2, \dots\ n$.

Тогда

$$T_{\Sigma} = T_0 + nT;$$

$$S_{\Sigma}\,\frac{T_{u_0} S\left(\dfrac{T_0}{T_{u_0}}\right) + T_u n S\left(\dfrac{T_i}{T_u}\right)}{T_{u_0} + T_u n};$$

$$0 \le T \le T_{\text{и}},\ 0 \le T_0 \le T_{n0}$$

(отсчет значений T и T_0 ведется от принятого за нуль уровня, обеспечивающего минимальную приемлемую надежность решения соответствующих операций). Исключив переменную T_0, перейдем к

$$S_{\Sigma} = \frac{1}{T_{u_0} T_u n}\left[T_{u_0} S\left(\frac{T_{\Sigma} - nT}{T_{u_0}}\right) + nT_u S\left(\frac{T}{T_u}\right)\right],$$

где

$$\max\left(0,\frac{T_{\Sigma}-T_{u_0}}{n}\right)\le T\le \min\left(T_{u,}\,\frac{T_{\Sigma}}{n}\right).$$

Вычислим

$$\frac{\partial^2 S_{\Sigma}}{\partial T^2}=\frac{1}{T_{u_0}+T_u n}\left[\frac{n}{T_{u_0}}S''\left(\frac{T_{\Sigma}-nT}{T_{u_0}}\right)+\frac{n}{T_u}S''\left(\frac{T}{T_u}\right)\right]\ge 0.$$

Таким образом, зависимость S_{Σ} *(Т)* вогнута и максимум S_{Σ} достигается на границах отрезка.

2. Примем, что трудоемкость реализации «идеальных моделей» консультационной и управленческой операции зависит от их размерности по степенному закону:

$T_и = \kappa m^{\beta}$; $T_{и0} = An^{\alpha}$; $\alpha \ge 1$, $\beta \ge 1$,

где m — число консультационных параметров, приходящихся в среднем на одну консультационную операцию, а α и β — показатели скорости возрастания сложности операций в зависимости от их размерности.

3. Рассмотрим консультационные процессы формирования рекомендаций двух видов в зависимости от схемы взаимодействия консультационных операций, на которые они расчленяются. При схеме «звезда» каждая операция взаимодействует со всеми другими операциями.

Степень взаимодействия можно охарактеризовать коэффициентом γ, указывающим среднее относительное число параметров, которым описывается взаимодействие с каждой операцией. Тогда

$$m=\frac{N}{n}(1+\gamma(n-1))=N(\gamma+\tfrac{1-\gamma}{n}).$$

Видно, что при значительном увеличении числа консультационных операций размерность каждой из них стремится к γN, т.е. практически не убывает. Поэтому увеличение их числа более чем $(1 - \gamma)/(0{,}05\gamma)$ нецелесообразно.

Если рассмотреть взаимодействия между консультационными операциями по схеме «лента», при которой степень взаимодействия характеризуется относительным увеличением числа консультационных параметров каждой операции за счет взаимодействия на величину γ, то

$$m=\frac{N}{n}(1+\gamma),\quad n>1.$$

В этом случае сложность каждой консультационной операции может быть сделана сколь угодно малой.

Значение T^*_Σ суммарной трудоемкости, необходимой для реализации консультационного процесса формирования рекомендаций как единственной консультационной операции с максимальной обоснованностью $S_\Sigma = 1$,

При $T_\Sigma < T^*_\Sigma$ нерасчлененный консультационный процесс формирования рекомендаций будет приводить к разработке рекомендаций с обоснованностью, меньшей 1. В то же время его оптимальное членение во многих случаях позволяет получить обоснованность рекомендаций равную единице. Необходимая для этого трудоемкость

$$\overline{T}_\Sigma = A^\alpha{}_n + kN^\beta\left(\gamma + \frac{1-\lambda}{n}\right)^\beta n \text{ (для схемы «звезда»);}$$

$$\overline{T}_\Sigma = A^\alpha{}_n + kN^\beta\left(\frac{1+\lambda}{n}\right)^\beta n \text{ (для схемы «лента».}$$

Исследуем оптимальную степень членения консультационного процесса формирования рекомендаций на примере квадратичной зависимости трудоемкости операции от ее сложности (α=β=2).

Для схемы «звезда» оптимальное число консультационных операций определится из соотношения

$$\text{Р}_{\text{зв}}(\gamma,n) = \frac{1}{n}\left(\frac{1}{n} + \frac{\gamma}{1-\gamma}\right)\left(\frac{1}{n} - \frac{\gamma}{1-\gamma}\right) = \frac{2A}{kN^2(1-\gamma)^2},$$

или

$$lg\text{Р}_{\text{зв}}(\gamma,n) = lg\frac{2}{(1-\gamma)^2} + \lg\frac{A}{k} - 2\lg N.$$

Аналогично для схемы «лента»

$$-3lgn = lg\frac{2}{(1-\gamma)^2} + \lg\frac{A}{k} - 2\lg N.$$

Соответствующие номограммы легко построить.

При отсутствии перекрытия (γ = 0) для обеих схем оптимальное число консультационных операций, обеспечивающее полную обоснованность сформированных рекомендаций при минимальных трудозатратах, задается формулой

$$n = 1{,}26\sqrt[3]{k/A}\,N^{\frac{2}{3}}$$

Необходимая трудоемкость в этом случае

$$\overline{T}_{\Sigma} = 2{,}37\sqrt[3]{Ak^2N^4}$$

Пусть трудоемкость выполнения равноценных по сложности (числу переменных) консультационных и управленческих операций примерно одинакова, т. е. $A = k$. Тогда

$$\overline{T}_{\Sigma} = 2{,}37AN^{\frac{4}{3}}$$

Таким образом, вместо квадратичного роста трудоемкости этих операций в зависимости от сложности последних их расчленение позволяет реализовать замедленный рост трудоемкости (с показателем 1,333 вместо 2). Выигрыш в трудоемкости показан в табл. 4.1.

Таблица 4.1

Размерность консультаци-онного процесса (число пере менных)	Уменьшение трудоемкост и за счет отималь-ного членения (раз)	Оптимальн ое число операций	Размерность отдельной консультационн ой операции (число переменных)
10	2,5	4	2-3
100	11	27	4
1000	42	126	8
10000	195	585	17

При схеме «лента» в случае взаимозависимости операций, на которые разбивается консультационный процесс, формула для оптимального числа операций содержит поправочный коэффициент

$$\sqrt[3]{(1+\gamma)^2} \ :$$

$$n = 1{,}26(1+\gamma)^{0{,}667}\sqrt[3]{k/A}N^{0{,}667}.$$

Необходимую трудоемкость

$$\overline{T}_{\Sigma} = 2{,}37R(\gamma)\sqrt{Ak^2N^4}$$

также вычисляют с использованием поправочного коэффициента

$$R(\gamma) = \left(\sqrt[3]{4(1+\gamma)^4} + \frac{1}{\sqrt[3]{2(1+\gamma)^2}} \right) \frac{1}{2,37},$$

значения которого при 0 ≤γ≤0,2 изменяются от 1,00 до 1,15. Таким образом, при схеме «лента» взаимовлияние консультационных операций не приводит к качественным изменениям. При схеме «звезда» сложность отдельных консультационных операций оказывается, за счет взаимовлияния, значительно большей, чем при схеме «лента». Это приводит к тому, что при наличии взаимовлияния операций их оптимальное число существенно уменьшается (табл. 4.2). Соответственно резко возрастает размерность отдельных консультационных операций, так что они, в свою очередь, оказываются нереализуемыми. В этом случае необходимо исходить из данной допустимой сложности консультационной операции $m_{\text{доп}}$:

$$n(1-\gamma)N/(m_{\text{доп}}-\gamma N).$$

Отсюда следует, что членению по схеме «звезда» со сложностью консультационных операций $m_{\text{доп}}$ поддаются консультационные процессы со сложностью не выше $m_{\text{доп}}/\gamma$. Затем может быть подсчитана необходимая трудоемкость формирования рекомендаций. Если она окажется неприемлемо большой, это означает, что при располагаемой мощности консультационной организации сформировать рекомендации со степенью обоснованности, равной единице, невозможно.

Таблица 4.2

Размерность процесса формирования рекомендаций (число переменных)	Уменьшение трудоемкости за счет оптимального членения (раз)		Оптимальное число операций		Размерность отдельной консультаци-онной операции (число переменных	
	γ=0,01	γ=0,1	γ=0,01	γ=0,1	γ=0,01	γ=0,1
10	2,31	1,77		3	3-4	3-4
100	9,22	2,72	18	8	6	21
1000	24	2,78	63	9	26	200
10000	25	2,78	99	9	200	2000

Таким образом, *приведенный упрощенный анализ показывает, что научно обоснованное членение консультационного процесса формирования рекомендаций на консультационные операции позволяет существенно повысить качество формирования рекомендаций (степень обоснованности сформированных рекомендаций).*

Учитывая сказанное, представление консультационного процесса формирования рекомендаций кортежем (4.2) должно быть дополнено, исходя из условия членения этого процесса на отдельные консультационные задачи.

В работе априорно не используется какая-либо конкретная методика членения консультационного процесса формирования рекомендаций на отдельные консультационные задачи. Считается, что такое членение является прерогативой ЛФР,ов, определяющих содержание этих задач, в том числе и используемые в процессе их решения критерии оценки альтернативных консультационных рекомендаций. В таком случае для формирования последовательности действий при автоматизированном консультировании структура консультационных задач, составляющих консультационный процесс формирования рекомендаций, должна быть представлена в общем виде.

Возможны два вида членения исходной задачи формирования рекомендаций на *совокупность частных консультационных задач*. Первый из них порождает вертикальную, а второй — горизонтальную структуру консультационного процесса формирования рекомендаций. Каждый из этих видов имеет свои специфические особенности. Они сказываются как на условиях, устанавливающих факт взаимозависимости результатов автономного решения частных консультационных задач, так и на процедурах согласования.

Рассмотрим вначале особенности, свойственные первому виду членения, согласно которому *весь консультационный процесс формирования рекомендаций представляется как множество последовательных уровней развития консультационного процесса*. На каждом из этих консультируемоя проблема рассматривается со все возрастающей степенью подробности. При этом вначале, не вдаваясь в детали, анализируются принципиально различающиеся варианты, определяемые ограниченным набором наиболее существенных для консультируемой проблемы рекомендаций [компонент вектора (y_0)]. Число этих вариантов несравненно меньше, чем число альтернативных вариантов рекомендаций вообще, определяемых полным вектором $y_{0.}$

Другими словами, вначале принимаются рекомендации, касающиеся принципиальных параметров, определяющих консультируемую проблему. Далее сформированные рекомендации, признанные лучшими на данном уровне и характеризующие пока лишь общие вопросы консультируемой проблемы, рассматривается в различных возможных вариантах его детализации и из них выбирается рациональный вариант дальнейшего развития рекомендаций и т. д.

При рассмотрении последовательных уровней детализации сформированных рекомендаций, принятых не предыдущем уровне, ограничивают допустимые варианты рекомендаций на последующих уровнях. В то же время возможно, что рекомендации, признанные рациональными на одном из уровней, при своей дальнейшей детализации оказываются нерациональными или нереализуемыми. Это свидетельствует о том, что взаимосвязи между сформированными рекомендациями могут проходить как от верхних уровней детализации рекомендаций к нижним, так и наоборот.

Связи типа сверху-вниз определяются тем, что результаты выполнения предыдущих уровней являются директивными данными для последующих уровней. Связи типа снизу-вверх обусловлены тем, что при анализе рекомендаций на каждом уровне должна учитываться информация о рекомендациях, которые будут приняты на последующих уровнях. Такая двусторонняя взаимосвязь различных уровней определяет хорошо известный итерационный характер консультационного процесса формирования рекомендаций.

Условия выделения из исходной консультационной задачи уровней детализации рекомендаций аналогичны рассмотренному выше выделению процесса формирования рекомендаций из жизненного цикла, а представление детализации на каждом уровне может быть аналогичным рассмотренному выше представлению процесса формирования рекомендаций в целом.

Обозначим каждый уровень детализации через S и введем на множестве этих уровней линейный порядок: $S_i \succ S_j$ если результаты выполнения S_j являются директивной информацией для S_i. В этом случае процесс формирования рекомендаций можно представить в виде

$$\Pi=(S_i)_{i=1,2,\dots,N^s};$$

$$S_i=<x_i,z_i,\Omega_i,V_i.K_i,F_i,W_i,\bar{y}_i>; \quad (4.3)$$

$$\bigcup_{i=1}^{N^s} y_i = y_0, \quad (4.4)$$

где N^s — число уровней детализации.

При этом реакции x_i определяются результатами выполнения операций детализации на более высоких, чем S_i, уровнях:

$$x_i = \varphi'_i\,(x_0, y_1, y_2, \ldots, y_{i-1}); \; i=2,\ldots,N^s; \; x_1=x_0, \qquad (4.5)$$

а z_i — результатами решения задач на более низких, чем S_i уровнях:

$$z_i = \tilde{\varphi}''_i\,(y_{i+1}, y_{i+2}, \ldots, y_{N^s}, z_0); \; i = 1, 2, \ldots, N^s - 1; \; z_{N^s} = z_0. \qquad (4.6)$$

Операторы φ'_i в выражении (4.5) задают трансформацию решений предыдущих уровней в исходные данные для выполнения текущего уровня детализации. В частности, они могут трактоваться как выделение из всех ранее принятых рекомендаций тех, которые являются для S_i исходными данными.

Операторы $\tilde{\varphi}''_i$ задают трансформацию рекомендаций с последующих уровней детализации в исходные данные для рассматриваемого уровня. Их основным отличием от операторов φ'_i, непосредственно задающих связи, передающие информацию с уровня на уровень, является то, что они передаваемую информацию кроме того преобразуют. Необходимость такого преобразования связана с тем, что размерность каждого вектора z_i существенно меньше размерности вектора аргументов в выражении (4.6), т. е.

$$|z_i| < \left| \bigcup_{i=i+1}^{N^s} y_i \right|,$$

так как только при этом условии проявляется целесообразность расчленения процесса формирования рекомендаций на уровни детализации рекомендаций.

Операторы $\tilde{\varphi}''_i$ целесообразно рассматривать как композицию операторов φ''_i и Q:

$$z_i = Q\,(\varphi''_i\,(y_{i+1}, y_{i+2}, \ldots, y_{N^s}, z_0)) \qquad (4.7)$$

Здесь Q — оператор, трансформирующий информацию с нижних уровней детализации в исходные данные для рассматриваемого уровня (т. е. в z_i), а операторы φ''_i определяют взаимосвязи между S_i и $\{S_j\}_{j=i+1,i+2,\ldots,N^s}$ так как не все $\{y_j\}_{j=i+1,i+2,\ldots,N^s}$ влияют на результаты выполнения S_i. Таким образом, при определении структуры информационного взаимодействия уровней детализации рекомендаций достаточно рассматривать операторы φ'_i, φ''_i (рис. 4.4).

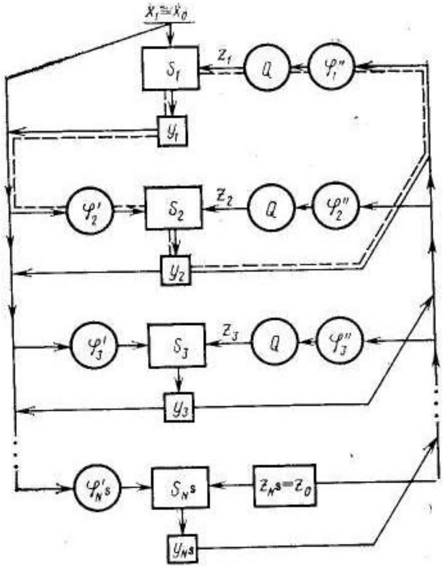

Рис. 4.4. Структура взаимосвязей между уровнями детализации рекомендаций (- - - обозначен один из итерационных циклов)

Как правило, конкретная информация о рекомендациях уровней, следующих за рассматриваемым, не известна. Поэтому в роли реакции z выступает некоторая прогнозная информация, получаемая путем преобразования прогнозируемых вариантов рекомендаций последующих уровней в эти данные. Такое преобразование производится оператором Q, один из вариантов реализации которого, имеющий место при использовании математического моделирования как основы для анализа и выбора сформированных рекомендаций, базируется на решении задачи идентификации математических моделей.

Характерной чертой взаимосвязей между различными уровнями детализации рекомендаций является их изменчивость в процессе формирования. Она определяется двумя факторами. Первый из них связан с тем, что при смене концепций формирования рекомендаций как на предыдущих, так и последующих уровнях возможны изменения в составе векторов y_i (j = 1, 2, ..., N^s) и, соответственно, изменение операторов φ'_i, φ''_i .

Второй из рассматриваемых факторов, определяющих динамичность взаимосвязей между различными уровнями детализации рекомендаций, связан с тем, что *основным принципом членения исходной задачи формирования рекомендаций на уровни детализации является ранжирование параметров консультируемых проблем по степени их влияния на принятые критериальные показатели и показатели соответствия заданию на формирование рекомендаций*. Степень этого влияния является показателем целесообразности рассмотрения того или иного параметра на определенном уровне. При смене используемых критериальных показателей возможно перераспределение сформированных рекомендаций по уровням детализации и, как следствие, трансформация операторов φ'_i, φ''_i, отражающих взаимосвязи между рекомендациями различных уровней детализации.

Итак, *в основе членения консультационного процесса формирования рекомендаций по уровням детализации лежит условие равнозначности влияния параметров одного уровня на принятые критерии оценки рекомендаций и показатели соответствия заданию на формирование рекомендаций*. В то же время известно, что по мере детализации рекомендаций число равнозначных параметров лавинообразно увеличивается, т. е. размерность консультационных задач, требующих решения на каждом последующем уровне, резко возрастает. Эти обстоятельства, а также потребность в распараллеливании работ с целью сокращения сроков формирования рекомендаций приводят к необходимости членения задач, требующих решения на каждом уровне детализации, на более мелкие. Это и определяет второй вид членения процесса формирования рекомендаций, порождающий горизонтальную структуру данного процесса.

В результате такого членения и *образуются консультационные операции, под которыми подразумеваются задачи анализа и выбора обоснованных сформированных рекомендаций, допускающие получение их законченных рекомендаций без дополнительного членения*. Операции будем обозначать через S и индексировать двумя числами натурального ряда: первое число определяет уровень, к которому принадлежит рассматриваемая операция, а второе — номер операции на рассматриваемом уровне. Соответственно будем индексировать и переменные, относящиеся к задачам той или иной операции.

Пусть N^s — число уровней детализации рекомендации, а N^s_i - число операций детализации на i-м уровне, $i = 1, 2, ..., N^s$. Тогда

$$S_i = \{S_j\}_{j=i+1,i+2,\ldots,\ N^s}\ (i=1,2,\ldots,\ N^s)$$

$$П = \{S_{ij}\}_{j=1,2,...,N^s_i}^{i=1,2,...,N^s}$$

Членение уровней детализации на консультационные операции приводит к «расслоению» векторов x_i, y_i, z_i на N^s_i подмножеств, в общем случае пересекающихся (рис. 4.5), а также порождает новый вид реакций — r_{ij}, которые отражают влияние на S_{ij} результатов выполнения других операций этого же уровня:

$$r_{ij} = \varphi'''_{ij}\ (y_{i1}, y_{i2}, ..., y_{i,i-1}, y_{i,i+1}, ..., y_{iN^s_i}); \quad (4.8)$$

$(i = 1, 2, ... , N^s, j = 1,2,…, N^s_i)$

где φ'''_{ij} — оператор, аналогичный φ'_i, φ''_i.

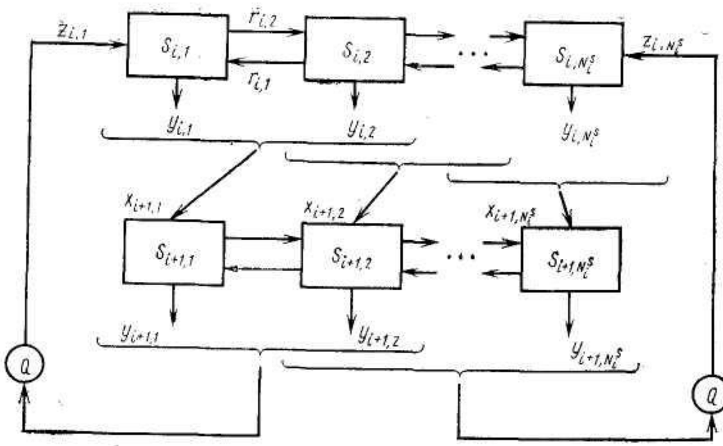

Рис. 4.5. Членение процесса формирования рекомендаций на консультационные операции

Как правило, реакции r_{ij} представляют собой часть компонент вектора y_{ij}, значения которых зависят от результатов выполнения соседних операций $r_{ij} \subseteq y_{ij}$. Фактически это означает взаимозависимость компонент вектора y_i, которая отрицается известными декомпозиционными методами, хотя это и не соответствует полностью действительности. Например, рассмотрим консультационные операции, при выполнении каждой из которых автономно рассматриваются разгонные блоки ступеней ЛА. Пусть в общем случае связи между этими операциями отсутствуют. В то же время, если постановка задачи формирования рекомендаций для проектирования ЛА пополняется условием равенства наружных диаметров разгонных блоков (без конкретного назначения требуемого диаметра), то взаимовлияние результатов автономного формирования рекомендаций для проектирования разгонных блоков, предусматривающего выбор рационального диаметра для каждого блока, очевидно. Аналогичная ситуация имеет место, если в постановке исходной задачи присутствует условие, ограничивающее длину ЛА. Следующая постановка задачи, когда при условии равенства диаметров разгонных блоков задано и требуемое их значение, снова приводит к отсутствию взаимосвязей между результатами выполнения указанных консультационные операций.

Приведенный пример показывает, что взаимосвязи между консультационными операциями одного уровня детализации (как и межуровневые связи) носят динамический характер. Вследствие этого разработанная для одной постановки задачи декомпозиционная схема формирования рекомендаций при изменениях этой постановки уже неприменима. Кроме того, сама разработка такой схемы далеко не всегда возможна.

Помимо возможного изменения постановки задачи формирования рекомендаций существует еще один фактор, определяющий динамичность связей между консультационными операциями одного уровня. Он состоит в изменении состава параметров формируемых рекомендаций при смене рассматриваемых концепций. В результате возможно, что при рассмотрении одних концепций при выполнении консультационных операций одного уровня детализации связи между ними отсутствуют, и они могут появиться при смене рассматриваемых концепций.

Итак, в общем случае, между консультационными операциями существуют два вида информационных связей: *вертикальные (порождаемые членением процесса формирования рекомендаций по уровням детализации) и горизонтальные (порождаемые членением каждого уровня детализации на консультационные операции)*. Связи каждого из этих видов в процессе формирования рекомендаций могут появляться и исчезать при изменении постановок задач, решаемых при выполнении консультационных операций, а также при изменении рассматриваемых в рамках этих операций консультационных концепций. В результате сказанного каждая консультационная операция может быть представлена в виде

$$S_{ij} = \langle x_{ij}, z_{ij}, \Omega_{ij}, r_{ij}, V_{ij}, Y_{ij}, K_{ij}, W_{ij}, \bar{y}_{ij\ j} \rangle \qquad (4.9)$$

$(i = 1, 2, \ldots. N^s, j = 1, 2, \ldots. N^s_i)$, где по сравнению с выражением (4.3) добавляется новая компонента — r_{ij}, выделяющая из y_{ij} те консультационные параметры, которые зависят от результатов выполнения соседних с S_{ij} операций.

Наличие информационной связи между каждой парой операций вызывает необходимость согласования результатов их выполнения или, другими словами, согласования результатов автономного функционирования консультационных модулей, на которые возложено выполнение этих операций.

Для того чтобы процесс согласования носил управляемый характер, обеспечивающий регулярное решение задач согласования, используют организационные связи между консультантами, выполняющими взаимосвязанные операции и входящими в состав соответствующих консультационных. Эти связи в процессе формирования рекомендаций, как правило, не изменяются. Они определяют, по сути, «каналы» организационного (административного) управления процессом согласования результатов автономного функционирования консультационных модулей, источники рассогласования которых определяются информационными связями.

Организационные структуры, как правило, являются древовидными и взаимодействие их элементов (консультационных модулей) может происходить либо строго по структурным связям (по подчиненности), либо по горизонтали, но только между элементами, имеющими общую ближайшую родительскую вершину. Это обеспечивает получение законченного решения задач согласования. В частности при взаимодействии консультантов различных уровней это достигается тем, что один из них имеет «право решающего голоса». Если взаимодействуют консультанты одного уровня, то в случае недостижения ими компромисса существует консультант, решения которого для них являются обязательными.

Организационные и информационные связи обычно не совпадают. В зависимости от соотношения между этими двумя видами связей информационные связи между консультационными модулями можно разделить на две группы (рис. 4.6).

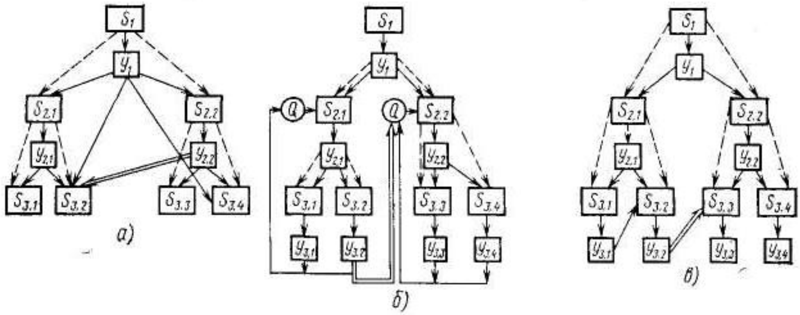

Рис. 4.6. Примеры информационных связей (———— - простые информационные связи; — — — — — - организационные связи; ═════ — комплексные информационные связи)

К первой из них относятся связи, совпадающие с организационными связями, а также между модулями, у которых совпадают ближайшие родительские вершины в организационной структуре. Задачи согласования, порождаемые наличием такого рода информационных связей (называемых далее простыми), в рамках заданной оргструктуры допускают их непосредственное решение.

Все оставшиеся связи относятся ко второй группе информационных связей, называемых далее комплексными. Особенностью согласования результатов выполнения консультационных модулей, находящихся в комплексном взаимодействии, является «разноподчиненность» входящих в их состав консультантов. Данное обстоятельство не позволяет рассчитывать на регулярное получение законченных результатов решения соответствующих задач согласования и требует сведения комплексных информационных связей к совокупности простых. Такое сведение всегда возможно, в частности, путем, преобразования каждой комплексной связи в горизонтальную, связывающую модули, которые:

выполняются консультантами, непосредственно подчиненными одному вышестоящему консультанту;

являются в оргструктуре родительскими вершинами для соответствующих консультационных модулей, находящихся в комплексном взаимодействии.

Принципиально возможны и другие трансформации комплексных связей в совокупности простых. Сведение комплексных информационных связей к простым является прерогативой организационного управления в САК и должно быть отражено в организационном обеспечении консультирующей САК. Применительно к формированию консультационных модулей важен тот факт, что в основу согласования результатов автономного функционирования КМ всегда могут быть положены задачи, порождаемые простыми информационными связями. Эти связи могут быть двух видов — *вертикальные и горизонтальные*.

Специфика задач согласования, порождаемых вертикальными связями, заключается в том, что:

1) результаты выполнения операций на более высоких уровнях являются директивной информацией для КМ, выполняющих операции нижеследующих уровней;

2) информация с нижних уровней детализации носит обобщенный («свернутый») характер и, как правило, является прогнозной.

Основным содержанием такого рода задач, далее называемых задачами вертикального согласования, является итерационное определение прогнозных рекомендаций z_{ij} ($i = 1, 2, ..., N^s — 1$;

$j = 1, 2, ..., N^s_i$) таким образом, чтобы они соответствовали рациональным рекомендациям операций последующих уровней детализации.

Специфика задач согласования, порождаемых горизонтальными связями (называемыми далее задачами горизонтального согласования), заключается в том, что:

приоритеты между результатами выполнения различных модулей отсутствуют;

информация, передаваемая от одного модуля к другому в процессе согласования, непосредственно описывает рассматриваемые при функционировании этих модулей формируемые рекомендации.

Основное содержание задач горизонтального согласования заключается в поиске компромисса между результатами функционирования «равноправных» модулей с возможностью подключения к этим работам вышестоящего в организационной структуре консультационного модуля (консультанта, входящего в состав этого вышестоящего модуля) в случае невозможности достижения компромисса.

В конечном счете, организация решения задач согласования должна обеспечить выполнение условия

$$\bar{y}_0 = \bigcup_{i=1}^{N^s} \bigcup_{j=1}^{N_i^s} \bar{y}_{ij}, \qquad (4.10)$$

свидетельствующего о совпадении y_0 — результатов решения исходной задачи консультирования в предположении ее решения без членения на консультационные операции с результатами решения этой же задачи $\overline{y}_{ij}$ *(i* = 1, 2, ..., N^s, *j* =1,2,..., N^s_i*)*, полученными путем выполнения совокупности консультационных операций.

Итак, рассмотрен переход от представления консультационного процесса в виде кортежа (4.2) к его представлению в виде выполнения совокупности консультационных операций. При этом каждая консультационная операция определяется в виде кортежа (4.9), который отличается от кортежа (4.2) следующим:

- реакции X_{ij} и Z_{ij} для каждой консультационной операции формируются по результатам выполнения операций более и менее высоких, соответственно, уровней детализации консультируемой проблемы. В кортеже (4.2) они являются известными константами;

- в рассмотрение вводятся реакции r_{ij}, которые отражают влияние на результаты функционирования того или иного консультационного модуля результатов выполнения соседних с ним модулей того же уровня детализации.

Зависимость реакции x_{ij}, $z_{ij,}$ r_{ij}, являющихся исходными данными для автономного выполнения текущей операции S_{ij}, от результатов выполнения других операций порождает необходимость решения задач согласования двух видов — вертикального и горизонтального.

Приведенное описание консультационного процесса, определяющее этот процесс как автономное выполнение информационно и организационно связанных консультационных модулей с последующим согласованием получающихся при этом результатов, позволяет перейти к непосредственному формированию структуры автоматизированного консультационного процесса на базе консультационных модулей.

4.3. Структура автоматизированного консультационного процесса на базе консультационных модулей.

Основываясь на изложенном представлении консультационного процесса, определим структуру этого процесса, реализуемого в САК. При этом будем основываться на рассмотрении САК как ОТС, консультационный процесс в которых составляют процедуры, связанные с формированием и функционированием КМ. В основу формулирования процедур, составляющих автоматизированный консультационный процесс, положим консультативные действия, приведенные на рис. 4.2 и связанные с определением элементов кортежа (4.9) применительно к отдельной консультационной операции как целевому компоненту КМ.

Первым из этих действий является выбор концепции Ω (здесь и далее, где очевидно, что речь идет об автономном выполнении консультационных операций, индексы операций опускаются), который носит, как правило, неформализуемый характер и основывается на опыте и интуиции консультантов. В дальнейшем это действие подробно не рассматривается как неавтоматизируемое. При этом полагают, что в результате его выполнения определяется множество концептуально одинаковых возможных формируемых рекомендаций Ω (4.1), описываемых непрерывной математической моделью.

Все последующие действия в процессе выполнения консультационных операций связаны с анализом сформированных консультантом концепций формируемых рекомендаций, соответствующих той или иной операции, и выделением из них наиболее обоснованного варианта. Причем анализ сформированных рекомендаций различных концепций ведется раздельно и его реализация является основой для определения формализуемых и рассматриваемых далее процедур функционирования САК.

Рассматриваемые процедуры могут быть разделены на *три группы*. К первой группе относятся процедуры, связанные с формированием КМ, достаточным для их автономного функционирования. Вторую группу составляет процедура автономного функционирования КМ, а третью — процедуры, обеспечивающие согласованность результатов автономного функционирования КМ. Формирование КМ, достаточное для их автономного функционирования, определяющим образом зависит от мощности множества допустимых рекомендаций *(Y),* в рамках рассматриваемой концепции. При этом принципиально возможны *три случая* (здесь и далее $|A|$ означает число элементов в множестве *A)* :

$$|Y| = 0;\ \ |Y| = 1;\ \ |Y|>1.$$

Если $|Y|=0$, то это означает, что в рамках рассматриваемой концепции сформированные рекомендации, удовлетворяющие директивным исходным данным, отсутствуют. Дальнейшие действия в этом случае должны быть связаны с переходом к рассмотрению следующей концепции или с изменением директивных исходных данных. Поскольку такое изменение не относится к компетенции выполняющего рассматриваемую операцию консультанта, его задачей в данном случае будет лишь сообщить о создавшейся ситуации вышестоящему консультанту. При этом действия по изменению директивных исходных данных при невозможности удовлетворения им переходят в класс задач вертикального согласования.

В дальнейшем при $| Y | = 0$ будем говорить, что по своему статусу формируемый консультационный модуль является невыполнимым.

Если $|Y|=1$, то это означает, что в рамках рассматриваемой концепции существует единственная сформированная рекомендация, удовлетворяющая директивным исходным данным. Определение этой рекомендации может быть произведено путем прямых расчетов. В этом случае будем говорить, что формируемый консультационный модуль имеет статус расчетного.

И, наконец, если $|Y| > 1$, то это означает, что число допустимых вариантов сформированных рекомендаций больше единицы и выделение из них того или иного варианта должно быть обосновано, что требует введения в рассмотрение критериев оценки предпочтительности вариантов и решения соответствующих оптимизационных задач. В данном случае будем говорить, что формируемый консультационный модуль имеет *оптимизационный статус.*

Таким образом, можно считать, что первой процедурой, определяющей функционирование САК, является определение статуса формируемых консультационных модулей. Исходными данными для ее выполнения являются директивные данные для рассматриваемой операции, а также вектор параметров, описывающих задаваемую консультантом концепцию формируемых рекомендаций. Выполняется эта процедура на базе использования формальных условий $V(x, y, z) \leq 0$, отраженных в математической модели консультируемой проблемы. В результате выполнения рассматриваемой процедуры консультанту должен быть сообщен статус формируемого им консультационного модуля, исходя из чего, он должен определить свои последующие действия.

В общем случае, как это было показано ранее, выполнение консультационных операций связано с сформированием рекомендаций при нескольких критериях. Их реализации при скалярном критерии, а также при расчетном статусе формируемых КМ являются частными случаями.

При использовании нескольких критериев выбора рациональных рекомендаций отсутствует возможность линейного упорядочивания имеющегося множества альтернатив по степени их предпочтения, как это имеет место при наличии одного критерия. В этом случае строгим является лишь построение на множестве альтернатив Y, так называемого множества недоминирующих элементов, — рекомендаций, оптимальных по Парето, которое будем обозначать в дальнейшем $Y_{\text{пар}}$.

Построение элементов множества Парето ($y \in Y_{пар}$) может быть произведено различными способами. Как пример, рассмотрим построение элементов $Y_{пар}$ путем скалярной оптимизации по одной из компонент вектора критериев при фиксированных значениях других его компонент. В этом случае общая схема реализации задачи выбора может быть представлена в виде, показанном на рис. 4.7.

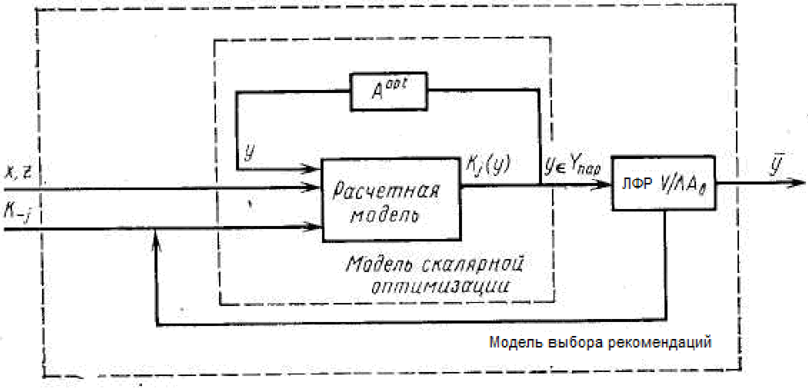

Рис. 4.7. Структура модели формирования информации для формирования рекомендаций на базе минимизации (максимизации) одной из компонент вектора критериев

Она включает:

1) расчетную модель консультационной операции, обеспечивающую вычисление свободной компоненты вектора критериев ($k_j \in K$) в зависимости от значений других его компонент ($K_{-j} = K/k_j$), а также от варианта сформированных рекомендаций ($<y>$) и исходных для рассматриваемой консультационной операции x, z;

2) алгоритм оптимизации (A^{opt}), находящийся в цепи обратной связи по отношению к расчетной модели и управляющий изменением варьируемых параметров y из условия нахождения экстремума по k_j;

3) ЛФР и/или формальный алгоритм выбора, управляющий переходом от одного элемента множества Парето к другому путем изменения значений компонент K_{-j}.

Построение такого рода схем и составляет в рамках сделанных выше допущений суть формирования КМ, каждый из которых является достаточным для автономного выполнения соответствующей консультационной операции. При этом имеется в виду, что алгоритмы оптимизации содержатся в соответствующих пакетах программ численных методов, формирование методик выбора представляет самостоятельную рассматриваемую ниже процедуру, а требуемая расчетная модель в конечном виде может не существовать. Данное обстоятельство может быть вызвано следующими причинами. Во-первых, даже решение одной и той же консультационной задачи, но, например, при использовании различных способов построения элементов множества Парето требует наличия различных расчетных моделей. Так, на рис. 4.8 показана схема, основанная на построении этих элементов на базе использования свертки критериев. Нетрудно заметить, что требуемая при этом расчетная модель отличается от той модели, которая необходима при построении парето-оптимальных решений на базе схемы, изображенной на рис. 4.7.

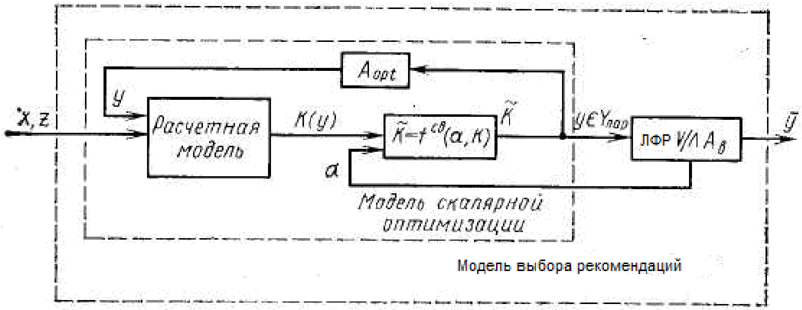

Рис. 4.8. Структура модели формирования информации для формирования рекомендаций на базе функции-свертки критериев

В частности, здесь необходима модель, с помощью которой определяют все компоненты вектора критериев, в то время как модель, представленная на рис. 4.8, вычисляет лишь одну из этих компонент, а другие для нее должны быть заданы.

Кроме того, структура и содержание консультационных модулей в значительной степени связаны с решаемыми задачами формирования рекомендаций, постановки которых могут варьироваться в достаточно широких пределах (например, формирование рекомендаций при проектировании нового изделия, доводка, модификация, учет готовых комплектующих элементов и т. д.). Как правило, такое варьирование вызывает изменения в составе исходных данных на выполнение консультационных операций, учитываемых ограничений и т. д. Данные изменения влияют, в первую очередь, на требуемые для выполнения этих операций расчетные модели.

В итоге можно сделать вывод, что в процессе формирования рекомендаций требуемые для выполнения каждой консультационной операции расчетные модели могут изменяться в достаточно широком диапазоне. Это делает предварительную «заготовку» таких моделей под каждую конкретную задачу в общем случае невозможной. В данных условиях необходимо обеспечить гибкое и оперативное формирование требуемых расчетных моделей из модулей ППП. Такие расчетные модели представляют собой некоторый агрегат, составленный из модулей пакета в определенных сочетаниях.

Резюмируя сказанное, вторую процедуру, связанную с автоматизированным формированием рекомендаций на базе КМ, будем определять как гибкое формирование расчетных моделей и обеспечение решений на их базе задач оптимизации и выбора сформированных рекомендаций в случае оптимизационного статуса КМ. Данная процедура совместно с процедурой определения статуса КМ составляют формирование консультационных модулей, необходимое для их автономного функционирования.

Непосредственно автономное функционирование КМ составляет самостоятельную процедуру автоматизированного формирования рекомендаций. Суть этой процедуры, в первом приближении, состоит в обоснованном выборе той или иной рекомендации из $Y_{\text{пар}}$. Можно предложить ряд подходов к выбору рациональных рекомендаций из $Y_{\text{пар}}$, которые условно можно разбить на *две группы*. Первую группу составляют методы, обеспечивающие полностью формализованный выбор рационального варианта из множества $Y_{\text{пар}}$ за счет задания некоторой априорной информации о сравнительной значимости различных критериев. Сюда относятся методы, основанные на использовании различных видов и способов свертки критериев и формирования некоторого единого критерия.

Ко второй группе относятся методы, основанные на участии в процессе формирования рекомендаций лиц, формирующих рекомендации (ЛФР). Сюда относятся методы, базирующиеся как на реконструкции функции-предпочтения на основе взаимодействия с ЛФР, так и неформальные методы непосредственного анализа ЛФР,ом элементов множества Парето.

В настоящей работе рассматривается подход, основанный на дозировании участия ЛФР в рамках формализованной процедуры выбора. Этот подход может также трактоваться как гибкое и оперативное построение методики выбора сформированных рекомендаций, учитывающее систему формируемых при этом предпочтений ЛФР, ряд формальных (аксиоматических) условий, а также регулярно поступающую к ЛФР информацию, полученную в результате построения по его заданию точек из $Y_{\text{пар}}$. В итоге процедура функционирования КМ далее рассматривается как последовательность действий ЛФР в рамках сформированного для автономного функционирования консультационного модуля, направленных на выделение из $Y_{пар}$ наиболее обоснованных сформированных рекомендаций.

Перейдем теперь к рассмотрению процедур третьей группы — процедур, обеспечивающих согласованность результатов автономного функционирования КМ. Необходимость в этих процедурах определяется тем, что результаты автономного функционирования отдельных модулей являются взаимозависимыми и, соответственно, должны быть согласованы. При этом взаимозависимость результатов КМ, как было указано выше, тождественна информационной связности этих модулей. Рассматриваемая группа состоит из двух процедур: выявления информационных связей между КМ и формирования среды для обмена информацией между ними в процессе согласования, результатов их автономного функционирования.

Первая из названных процедур непосредственно вытекает из динамичности информационных связей между различными консультационными модулями. Данная динамичность проявляется в том, что информационные связи между каждой парой операций то появляются, то исчезают, в частности, при смене рассматриваемых концепций формирования рекомендаций, критериев оценки этих рекомендаций и т. д. В процессе выполнения данной процедуры необходимо:

- выявлять наличие информационных связей между различными КМ в процессе формирования рекомендаций ;

- сообщать о наличии такого рода связей соответствующим ЛФР,ам, а также о том, какие из компонент рассматриваемых ими сформированных рекомендаций являются взаимосвязанными и не допускают их независимого выбора.

При рассмотрении данной процедуры в дальнейшем будем учитывать, что природа информационных связей различных видов (вертикальных — между уровнями детализации сформированных рекомендаций и горизонтальных — между консультационными операциями одного уровня детализации) существенно различна.

В заключение рассмотрим процедуру формирования среды обмена информацией между КМ в процессе согласования результатов их автономного функционирования. Состав этой среды (рис. 4.9) определяется, во-первых, математической моделью, связывающей переменные, являющиеся исходными данными и результатами автономного функционирования различных КМ.

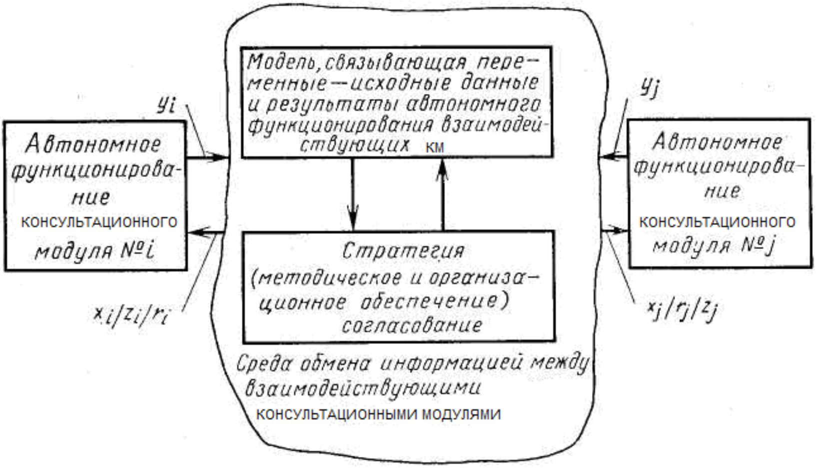

Рис. 4.9. Взаимодействие консультационных модулей

Необходимость такой модели вызвана тем, что взаимодействие некоторой пары КМ происходит через генерируемые ими параметры, которые в общем случае не перересекаются, но допускают их взаимный «пересчет» с использованием связей математической модели консультируемой проблемы. Эти связи, по сути, и составляют модель, входящую в состав среды обмена информацией между КМ. Случай, когда взаимодействие порождается пересечением параметров и характеристик, генерируемых различными КМ, является частным, которому соответствуют связи тождественности переменных, присутствующих в расчетных моделях рассматриваемых КМ. Здесь же заметим, что в роли указанных связей могут выступать не только связи непосредственно математических моделей или тождественности как их частный случай, но и связи других видов, в частности, связи идентификации, которые порождают вертикальное информационное взаимодействие.

Помимо связей, трансформирующих результаты выполнения одних КМ в исходные данные для других КМ, в состав среды обмена информацией также входят элементы методического обеспечения и их программно-информационное отражение, а также соответствующее организационное обеспечение. Их совокупность должна обеспечить целенаправленный поиск компромисса в процессе согласования результатов автономного функционирования КМ.

Формирование консультационных концепций, а также названные пять процедур (определение статуса КМ, формирование требуемых моделей, автономное функционирование КМ, выявление информационных связей КМ, формирование среды обмена информацией между взаимодействующими КМ) определяют в основном консультационный процесс автоматизированного формирования рекомендаций на базе консультационных модулей.

Заметим, что первые две процедуры, связанные с определением статуса КМ и гибким формированием требуемых расчетных и оптимизационных моделей, являются необходимыми для формирования автономно функционирующих КМ, но не достаточными. Это объясняется тем, что уже вначале автономного функционирования КМ необходимо знать, какие выбираемые в процессе этого функционирования параметры консультируемой проблемы являются «свободными», а какие так или иначе «стеснены» наличием других КМ. Данное обстоятельство делает целесообразным объединить названные процедуры в самостоятельный этап автоматизированного формирования рекомендаций на базе консультационных модулей — *этап формирования КМ для их автономного функционирования*.

На основе изложенного выше приведем *структуру консультационного процесса автоматизированного формирования рекомендаций на базе консультационных модулей* (рис. 4.10).

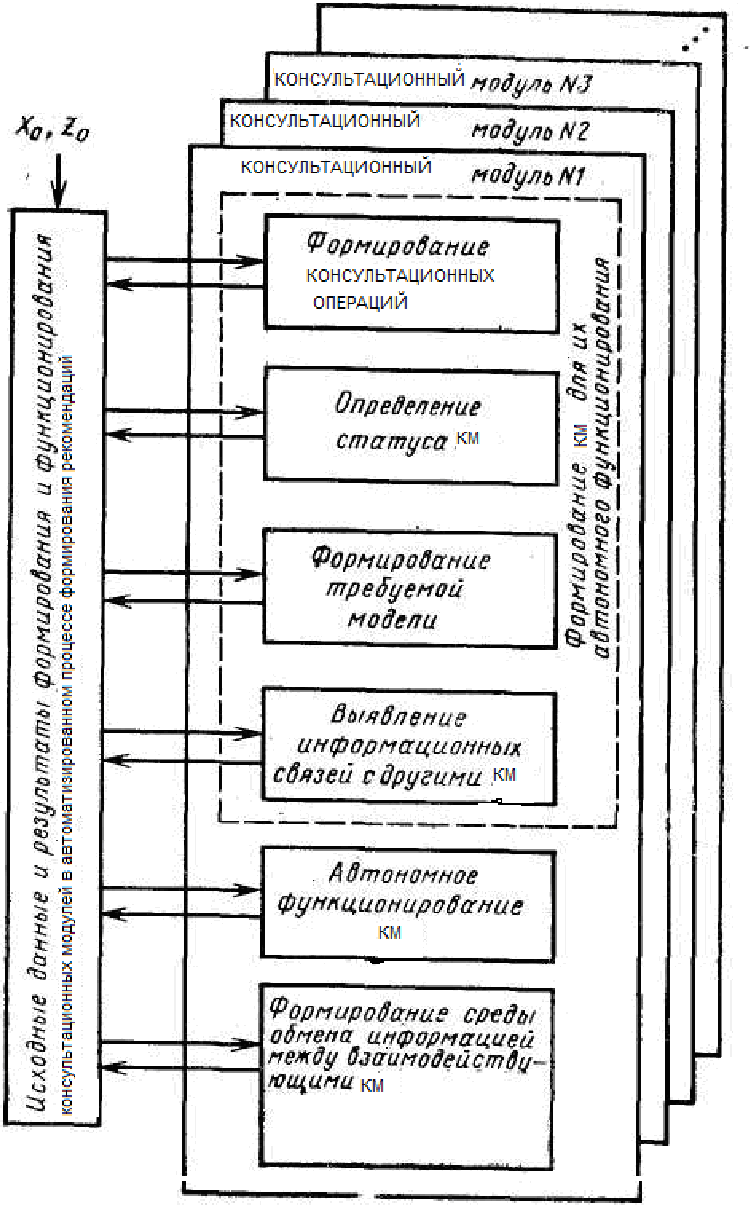

Рис. 4. 10. Структура процесса автоматизированного формирования рекомендаций на базе консультационных модулей

Данный процесс подразумевает, что в САК одновременно происходят процессы формирования и функционирования множества КМ. Для каждого КМ всегда вначале производится его формирование, а затем функционирование. Инициализацией для формирования того или иного модуля является задание на выполнение определенной операции. Это задание генерируется как результат функционирования КМ, в состав которого входит ЛФР, занимающий более высокое положение в структуре организационной системы САК, по сравнению с ЛФР, входящим в состав инициализируемого модуля. При этом полагается, что одновременно с определением требующей выполнения консультационной операции определяется и ЛФР, который, используя возможности управляющего модуля САК, производит формирование КМ и далее — участвует в процессе его функционирования.

Вначале формирования КМ определяется консультационная концепция и статус КМ, необходимого для анализа этой концепции. Если статус не невыполнимый, то формируется соответствующая рассматриваемой консультационной операции расчетная (если статус расчетный) или оптимизационная (если статус оптимизационный) модель. Кроме этого, определяются информационные связи с другими КМ. При этом выделяются параметры, значения которых так или иначе зависят от результатов функционирования рассматриваемых модулей. На этом формирование КМ завершается.

Далее происходит автономное функционирование сформированного КМ, состоящее в определении рациональных сформированных рекомендаций.

Если рассматриваемый КМ, например, выполняющий операцию S_α, информационно связан с некоторым другим, выполняющим операцию S_β, то еще на шаге формирования этого модуля выделяются те параметры (это могут быть различные сочетания компонент z_α, $r_\alpha \subseteq y_\alpha$), значения которых так или иначе зависят от определенной группы компонент y_β. Для реализации требуемого при этом «пересчета» результатов выполнения операции S_β в исходные данные для S_α формируется среда обмена информацией между соответствующими КМ. При этом определяются, как отдельные компоненты этой среды, методики согласования, задающие алгоритм координации процесса согласования. Данный процесс сводится, в основном, к итерационному обмену информацией о результатах автономного функционирования между взаимодействующими базе консультационными модулями (выполняющими операции S_α, S_β). Его целью, в общем случае, является построение множества Парето для набора критериев $\widetilde{K}_{\alpha,}\widetilde{K}_{\beta}$ и определение на нем точки компромисса. Этими критериями, например, могут быть отклонения основных критериев K_{α}, K_{β} от их «идеальных» значений, в роли которых выступают значения, получаемые при автономном выполнении каждой операции без учета их взаимодействия.

Описанный выше процесс автоматизированного формирования рекомендаций на базе КМ в целом адекватен сложившейся к настоящему времени практике формирования рекомендаций для сложных проблем. Однако входящие в этот процесс процедуры формирования вокруг ЛФР «среды», в которой он решает возлагаемые на него задачи, а также сами процедуры решения этих задач имеют новое содержание.

В дальнейшем будем говорить, что процедуры формирования КМ для их автономного функционирования, а также формирование среды взаимодействия КМ образуют процесс формирования КМ. Реализация этого процесса возлагается на управляющий модуль САК и является его основным назначением, так же как и поддержка функционирования КМ. Вопросы, связанные с реализацией процесса формирования КМ, а также структура и принципы работы управляющего модуля САК и функционирование КМ, состоящее в выборе наиболее обоснованных сформированных рекомендаций в условиях неопределенности, рассматриваются во второй книге настоящей работы

4.4. Методы задания предпочтения на множестве частных критериев качества

4.4.1. Постановка задачи

При автоматизированном формировании рекомендаций для решения задач консультируемой проблемы ЛФР взаимодействует с техническими средствами САК в интерактивном режиме. В процессе этого взаимодействия на основе анализа множества альтернативных вариантов сформированных рекомендаций, получаемых с помощью технических и программных средств САК, ЛФР должен сформировать рекомендации по выбору оптимального варианта решения задач консультируемой проблемы, т. е. решить задачу выработки предпочтения среди некоторого множества альтернативных вариантов сформированных рекомендаций решения задач консультируемой проблемы. Рекомендации ЛФР формирует на основе выбранных критериев. При существовании одного частного критерия формирование рекомендаций производится однозначно путем сравнения значений данного критерия для различных альтернативных вариантов.

В многокритериальных задачах оптимального формирования рекомендаций возникает необходимость объективной оценки важности частных критериев, включаемых в *аддитивный, мультипликативный или минимаксный критерий оптимальности*.

Оценивают важность частных критериев $F_i(\mathbf{X})$, i=l,…, n, с помощью весовых коэффициентов c_i, которые должны количественно отражать важность соответствующих частных критериев. Значения c_i выбирают исходя из анализа современного мирового уровня развития данной отрасли, из требований к консультируемой проблеме, из существующих возможностей реализации этих требований. Открытие новых физических принципов и разработка новых методов формирования рекомендаций могут существенно влиять на значения весовых коэффициентов.

Рассмотрим основные подходы к решению задачи выработки предпочтения на множестве частных критериев.

Экспертные оценки. В теории экспертных оценок разработан ряд методов проведения экспертизы. Наиболее эффективными в проводимых исследованиях оказались методы ранжирования и приписывания баллов.

Метод ранжирования заключается в следующем. Пусть экспертиза проводится группой из l экспертов (ЛФР), которые являются квалифицированными специалистами в той области, для которой формируются рекомендации решения задач консультируемой проблемы. Метод ранжирования основан на том, что каждого эксперта просят расставить частные критерии $F_i(\mathbf{X})$, i=l,…, n, оценки важности задачи консультируемой проблемы в порядке их важности. При этом цифрой 1 обозначают наиболее важный частный критерий (параметр), цифрой 2— следующий по степени важности частный критерий и т. д. Эти ранги преобразовывают таким образом, что ранг 1 получает оценку n, ранг 2 — оценку $(n$—1) и т. д. до ранга n, которому присваивается оценка 1, где n — число частных критериев. Зная преобразованный ранг $r_i^{(k)}$ 1-го критерия у k-го эксперта (k=1,…, l), весовые коэффициенты c_i, i=1,…, n определяют из следующего соотношения:

$$c_i=\sum_{k=1}^{i} r_i^{(k)}/\sum_{i=1}^{n}\sum_{k=1}^{i} r_i^{(k)},\ i=1,\dots,n \qquad (4.11)$$

Метод приписывания баллов основан на том, что эксперты оценивают важность частного критерия по шкале 0—10. При этом разрешается оценивать важность дробными величинами или приписывать одну и ту же величину из выбранной шкалы нескольким критериям. Зная балл $h_i^{(k)}$ i-го критерия у k-го эксперта, весовые коэффициенты c_i можно найти из (4.11), заменив в нем $r_i^{(k)}$ на

$$H_i^{(k)}/ = h_i^{(k)}/ \sum_{i=1}^{n} h_i^{(k)}$$

Последний называют весом, подсчитанным для i-го частного критерия $F_i(\mathbf{X})$ на основе оценок k-го эксперта.

Важное место занимает обработка результатов экспертных оценок. Если рассматривать результаты оценок каждого из экспертов как реализации некоторой случайной величины, то к ним можно применять методы математической статистики.

В общем случае при определении степени важности частного критерия $F_i(X)$ получают набор оценок $r_i^{(k)}$, $k = 1,...,l$, подлежащих статистической обработке. Среднее значение оценки

$$r_i= \sum_{k=1}^{l} (\mu_k r_i^{(k)})/ \sum_{k=1}^{l} \mu_k$$

где μ_k — коэффициент авторитета k-го эксперта ($0 \le \mu \le 1$). Среднее значение оценки r_i выражает коллективное мнение группы экспертов. Степень согласованности мнений экспертов характеризуется величиной называемой *дисперсией экспертных оценок.* Ясно, что чем меньше величины дисперсии, тем с большей уверенностью можно опираться на найденное значение r_i оценки степени важности частного критерия $F_i(\mathbf{X})$. Надежность экспертизы тем выше, чем меньшую долю среднего значения составляет среднеквадратический разброс оценок σ. Поэтому в качестве меры надежности приведенной экспертизы часто принимают $\beta = \sigma/ r_i$ и называют *вариацией.*

По среднему значению оценки r_i определяются весовые коэффициенты:

$$c_i = r_i / \sum_{i=1}^{n} r_{i,} \, i=1,...,n$$

Статистическая обработка результатов экспертных оценок подобна статистической обработке результатов измерений. На достоверность экспертизы существенно влияют такие факторы, как численный состав экспертной группы, уровень компетентности экспертов, состав вопросов, предъявляемых экспертам, и т. д.

Индивидуальные экспертные оценки также носят на себе печать случайности: на суждения эксперта влияют не только такие стабильные факторы, как его знания и опыт, но и множество случайных факторов (настроение, самочувствие, обстановка и т. п.).

Формальные процедуры. Методика формального определения весовых коэффициентов основана на анализе современного уровня развития конкретной отрасли, в области которой производится консультационная деятельность. Каждому значению вектора рекомендуемых параметров функционирования консультируемого объекта $\mathbf{X}_k$ соответствует альтернативный вариант S_k консултируемого объекта, качество (или эффективность) которого может быть оценено различными способами, и в частности в виде аддитивного критерия

$$F^{(k)}(\mathbf{X}) = \sum_{i=1}^{n} c_j f_i^{(k)}(\mathbf{X}),\ k = 1,\dots,m, \qquad (4.12)$$

где $f_i^{(k)}(\mathbf{X})$— нормированное значение i-го частного критерия для варианта S_k.

Для определения значений $f_i^{(k)}(\mathbf{X})$ построим матрицу параметров

$\Theta = [\upsilon_i^{(j)}]$, i=l,…,n, j=1,…, m, множества $S=\{S_1, ..., S_m\}$ альтернативных вариантов. В матрице Θ вектор-строка $\Theta_i=(\upsilon_1^{(j)}, .,., \upsilon_n^{(j)})$ описывает приемлемый вариант рекомендаций для $S_j \in S$ консультируемой проблемы. Для перехода от $\upsilon_i^{(j)}$ к $f_i^{(k)}(\mathbf{X}) = f_i^k$ при фиксированном векторе переменных функционирования $\boldsymbol{X}_к$ введем совокупность директивных значений параметров $\Theta = (\upsilon_1^{(0)}, .,., \upsilon_n^{(0)})$, устанавливаемых в КЗ на консультирование проблемы. Тогда нормированные (относительные) значения параметров определяются как

$$f_i^{(j)} = \upsilon_i^{(j)} / \upsilon_i^{(0)},\ i=1,\dots,n,\ j=1,\dots,m. \qquad (4.13)$$

Определим в матрице Θ величины υ_i^*— экстремальные (наилучшие) значения всех параметров. Очевидно, что идеальный вариант рекомендации $S_и$ должен описываться всеми υ_i^*, i=l,…,n .

Для оценки степени важности каждого параметра υ_i (или каждого нормированного значения параметра f_i) вводится система весов

$\mathbf{C}= (c_1 ..., c_n)$, которая должна отражать усилия, необходимые для достижения экстремальных значений параметров (увеличить значения таких параметров, как производительность, надежность и другие, или уменьшить значения массогабаритных, стоимостных и энергетических параметров). Правильный выбор системы весов открывает возможность целенаправленно воздействовать на улучшение тех или иных параметров рекомендации путем увеличения соответствующих весов c_i. Для осуществления этой возможности система весов не должна быть застывшей, а должна быть гибкой и должна меняться в зависимости от назначения (функционирования) консультируемой проблемы и состояния развития консультируемой отрасли в настоящий момент времени. В основу выбора системы весов положим принцип ограниченности общих затрат, необходимых для формирования и реализации рекомендаций. Это означает, что увеличение затрат на улучшение одних параметров неизбежно вызывает уменьшение затрат на улучшение других параметров.

Методика формального определения весовых коэффициентов базируется на выполнении последовательности процедур выработки предпочтения среди каждой пары показателей f_i и f_k.

Обозначим через f_{ik} значение показателя f_i в варианте рекомендации, в котором максимальные затраты сосредоточены на увеличении показателя f_k, а через f^*_i — наилучшее значение показателя f_i во множестве альтернативных вариантов S, т. е.

$$f^*_i = \underset{j=1,\dots,m}{ext}\{ f_i^{(j)} \} \;;\; f_{ik} = f_i \,|f_k = f^*_k$$

где $f_{ik} = f_i \,|f_k = f^*_k$ —значение показателя f_i в варианте S_j для которого

$$f^{(j)}_k = f^*_k$$

При этом

$$\Delta f^*_i = \ln| f^*_i - f_i^{(0)}| = \ln| f^*_i - 1| \qquad (4.14)$$

определяет близость директивного значения показателя

$f_i^{(0)} = \upsilon_i^{(0)}/\upsilon_i^{(0)} = 1$ к наилучшему f^*_i. Чем большее значение придается показателю f_i, тем меньше должно быть Δf^*_i. Следовательно, вес, придаваемый показателю f_i, должен быть обратно пропорционален величине Δf^*_i, что позволяет записать

$$c_i \approx 1/\Delta f^*_i.$$

Величина

$$\Delta f_{ik} = \ln| f^*_i - f_{ik} | \qquad (4.15)$$

определяет ухудшение показателя f_i в варианте рекомендации, в котором максимальные затраты сосредоточиваются на улучшении показателя f_k. Если величина Δf_{ik} мала, то это означает, что сосредоточение затрат на увеличении f_k не ухудшает существенно f^*_i и что, следовательно, поддержание f^*_i на высоком уровне не требует больших затрат и величина c_i должна быть взята малой, и наоборот. Следовательно, вес показателя f_i должен увеличиваться с увеличением Δf_{ik}. Это утверждение справедливо для любого $k \in \{1, ..., n\}$, откуда следует, что

$$c_i \approx \sum_{k=1}^{n} \Delta f_{ik} \qquad (4.16)$$

С учетом (4.15) и (4.16) можно записать

$$c_i = \sum_{k=1}^{n} \Delta f_{ik} / \Delta f^*_i = \sum_{k=1}^{n} l_{ik}, \qquad (4.17)$$

где $l_{ik}=\Delta f_{ik}/\Delta f^*_i$ будут тем больше, чем большее значение придается показателю f_i и чем сильнее сказывается на снижении этого показателя сосредоточение усилий на показателе f_k.

Следовательно, величины l_{ik} могут рассматриваться как относительные веса, показывающие относительное превосходство (доминирование) показателя f_i над f_k.

Однако использование (4.17) для определения весов наталкивается на ряд трудностей. Во-первых, описанная методика определения l_{ik} оказывается неприемлемой при $k=i$, т. е. величины l_{ii}, $i=1,\ldots, n$, остаются неопределенными. Ими, правда, можно было задаться произвольно, однако это вносит произвол в определение весов c_i. Во-вторых, величины l_{ik}, отражающие, как отмечалось, превосходство показателя f_i над показателем f_k и дающие соответствующие вклады в суммарный вес c_i, входят в (4.17) с коэффициентами, равными единице, т. е. не учитываются веса показателя f_k. В то же время превосходство l_{ik} параметра f_i над f_k может быть превосходством «сильного» над «слабым», поэтому значительность этого превосходства должна быть пропорциональна весу параметра f_k. Исходя из этого, следует заменить в (4.17) величины l_{ik} на $c_k\, l_{ik}$.

Такой подход приводит к необходимости использования для определения весов метода итерации. В нулевом приближении веса всех показателей принимаются одинаковыми и равными $c_i^{(0)}=1$. Далее, если определены веса r-го порядка, то переход к весам $r+1$-го порядка будем осуществлять по формуле

$$c_i^{(r+1)} = \sum_{k=1}^{n} c_k^{(r)}\, l_{ik},$$

согласно которой веса первого, второго и т. д. порядков будут

$$c_i^{(1)} = \sum_{k=1}^{n} c_k^{(0)}\, l_{ik} = \sum_{k=1}^{n} l_{ik};$$

$$c_i^{(2)} = \sum_{k=1}^{n} c_k^{(1)}\, l_{ik};$$

$$\cdots\cdots\cdots$$

Данный процесс довольно быстро приводит к установившейся системе весов, не зависящих от последующих итераций и от величин l_{ik}. В связи с этим значения l_{ik} можно выбирать произвольно, например равными 0,5 или 1. Нормированные веса всех показателей после проведения t итераций определяются как

$$c_i = c_i^{(t)} / \sum_{i=1}^{n} c_i^{(t)}$$

Если подмножество параметров $\overline{\Theta}_j \subset \Theta_j$, описывающее вариант рекомендации $S_j \in \overline{S}$, является подмножеством наилучших значений параметров

$$\Theta_j = (\vartheta_1^{(j)} = \vartheta_1^*, \ldots, \vartheta_q^{(j)} = \vartheta^*_q) ,$$

то возникает неопределенность при нахождении относительных весов c_i по формуле (4.17).

Для разрешения этой неопределенности перейдем от матрицы параметров $\Theta = \|\upsilon_i^{(j)}\|$ к матрице относительных показателей $\mathbf{A}=[f_i^{(j)}]$, $i=1,\ldots, n$, $j=1,\ldots,m$, элементы $f_i^{(j)}$ которой образуются согласно (4 .13).

Множество строк $\overline{A} = \{A_1 \ldots A_n\}$ матрицы **А** расщепим на два подмножества $\overline{A}_j$ и A'_j ($\overline{A} = \overline{A}_j \cup A'_j$), причем подмножество $\overline{A}_j$ (мощностью q) является порождением множества $\overline{\Theta}_j = (\vartheta_1^{(j)}, \ldots, \vartheta_q^{(j)})$ посредством преобразования (4.13), а A'_j (мощностью $n—q$) — порождением множества

$$\Theta \setminus \overline{\Theta}_j = (\vartheta_{q-1}^{(j)}, \ldots, \vartheta_n^{(j)}) .$$

Сформируем q матриц $\mathbf{A}^{(k)}$ относительных параметров размерности $(n\text{-}q+l)m$ таким образом, что множество строк $\overline{A}^{(k)}$ матрицы $\mathbf{A}^{(k)}$ образуется из одного элемента $A_k \in \overline{A}_j$ и всех элементов множества A'_j.

Для каждой матрицы $\mathbf{A}^{(k)}$, $k=1,\ldots, q$, определяют значения весовых коэффициентов $\widetilde{c}_k = (\widetilde{c}_k, \widetilde{c}_{k+1}, \ldots, \widetilde{c}_n)$ согласно описанной выше итерационной процедуре. Поскольку значения весовых коэффициентов c_k параметра f_k в матрицах $\mathbf{A}^{(k)}$, $k=1,\ldots, q$ определены относительно одного и того же подмножества параметров A'_j, то это позволяет найти доминирование параметра f_i над параметром f_k ($f_i, f_k \in A'_j$) как

$l_{ik} = \widetilde{c}_i / \widetilde{c}_k$, а затем все полученные l_{ik}, $i, k \in \{1, \ldots, q\}$ использовать для определения весовых коэффициентов c_i более высоких порядков.

Проиллюстрируем формальную методику определения весовых коэффициентов на примере.

Пример. Требуется сформировать рекомендации по выбору наилучшего варианта проектируемой системы автоматического регулирования и выбрать из них наилучшую.

При проектировании системы автоматического регулирования были сформированы рекомендации на три конкурирующих варианта, эквивалентных по функциональному назначению системы, S_1, S_2 и S_3, параметры которых приведены в табл. 4.3. Отметим, что все $\upsilon_i \le \upsilon^{(0)}_i$, i=1,2,3 и все параметры целесообразно минимизировать.

Таблица 4.3

Тип системы S_k	Параметры			Относительные показатели		
	υ^k_1, с	υ^k_2,Вт	υ^k_3, усл. ед.	f^k_1	f^k_2	f^k_3
S_1	0,1	10	15	0,17	0,5	1,0
S_2	0,3	13	5	0,5	0,65	0,33
S_3	0,6	17	10	1, 0	0,35	0,67
Требования технического задания $\upsilon^{(0)}_i$	0,6	20	15	1,0	1,0	1,0

Примечание. Здесь υ_1 — время регулирования; υ_2 — энергопотребление; υ_3— сложность аппаратурной реализации; $\upsilon^{(0)}_i$ — директивные значения параметров.

В связи с тем, что все требования ТЗ для всех систем выполнены и не требуется применять определенных усилий для достижения заданных директивных значений переменных, рекомендуется вместо соотношений (4.14) и (4.15), необходимых для определения весовых коэффициентов параметров, использовать выражения вида

$$\Delta f_i^* = |\, f_i^* - f_i^{(0)}| = |\, f_i^* - 1|; \quad \Delta f_{ik} = |f_i^* — f_{ik}\,|,$$

Из табл. 4.3 следует, что

$$\min_k \{f_1^{(k)}\} = f_i^* = 0{,}17; \ \min_k \{ f_2^{(k)})\} = f^*_2 = 0{,}35; \ \min_k \{ f_3^{(k)}) = f_3^* = 0{,}33.$$

В табл. 4.4 приведены результаты расчетов величин $f_i^{(k)}$, $c_i^{(1)}$ и c_i на 1-й итерации. Величины l_{ii} рекомендовано взять равными 0,5.

Т а б л и ц а 4.4

f_i	f_1	f_2	f_3	$c^{(1)}_i$	c_i
f_1	0,5	1,0	0,397	1,897	0,372
f_2	0,231	0,5	0,461	1,192	0,234

f_3	1,0	0,507	0,5	2,007	0,394

Рассмотрим формирование элементов первой строки табл. 4.4:
l_{11} = 0,5; l_{12}= $\Delta f_{12}/\Delta f_1^*$ = | f_1^*- f_{12}| / | f_1^*- 1 |= |0,17-l|/|0,17-l| = l,0;

l_{11}= |0,17-0,5|/|0,17-1|=0,397; $c_1^{(1)}\sum_{k=1}^{3} l_{1k}$ = 1,897;

$c_1 = c_1'/ \sum_{k=1}^{3} c_k^{(1)}{}_1$ = 1,897/5,096 = 0,372.

Рекомендуется использовать формулу $c_i^{(r+1)}=\sum_{k=1}^{n} c_i^{(r)}\ l_{1k}$ и рассчитать по ней весовые коэффициенты более высоких порядков. Результаты расчетов сведены в табл. 4.5.

Т а б л и ц а 4.5

№ итерации	c_1	c_2	c_3
1	0,372	0,234	0,394
2	0,349	0,234	0,417
3	0,350	0,238	0,412
4	0,351	0,237	0, 412
5	0,351	0,237	0, 412

Из таблицы видно, что значения весовых коэффициентов параметров стабилизировались к четвертой итерации.

Руководствуясь сформированными рекомендациями, используя выражение (4.12) определим значение аддитивного критерия для всех вариантов рекомендаций:

$F^{(1)}$ (**X**) = 0,590; $F^{(2)}$ (**X**) = 0,466; $F^{(3)}$ (**X**) =0,710.;

Согласно сформированных рекомендаций, целесообразно минимизировать значение аддитивного критерия. В рассматриваемом случае, наиболее предпочтительным оказывается вариант S_2.

В настоящее время получили распространение интерактивные методы решения многокритериальных задач, когда информация о важности и предпочтениях приходит как от ЛФР,а, так и от ЭВМ. Уточнение обобщенных критериев и упорядочивание критериев по важности рекомендуется производить на основе диалога ЛФР с ЭВМ. Часто для формирования наилучшей рекомендации ЛФР,у приходится решать задачи *структурной и параметрической оптимизации*. При этом модель формирования рекомендаций описывается как задача *многокритериальной оптимизации*. В этом случае используют интерактивный режим оптимизации или диалоговой оптимизации. ЛФР может изменить процесс формирования рекомендаций для решения задачи на любом этапе, параметры, метод решения, математическое описание задачи. Проблемами здесь являются разработка эффективных пакетов прикладных программ, сценариев диалога, эвристических и точных алгоритмов формирования рекомендаций с учетом расплывчатости и неопределенности интеллектуальной деятельности ЛФР,а.

В большинстве важных консультационных задач формирование рекомендаций связано с необходимостью полного органичного учета большого числа критериев (факторов). В таких случаях возникает проблема оценки и сравнения предпочтительности различных вариантов при *векторном критерии*. Общую постановку задачи векторного синтеза рекомендаций формулируют следующим образом: *каждая альтернатива рекомендации характеризуется вектором* $K=\{K_i;\ i=1, 2,..., n\}$ *показателей качества*. В n-мерном пространстве F^n показателей качества F_i каждой альтернативе соответствует единственное вполне определенное значение вектора K.

Обычно невозможно найти рекомендацию, удовлетворяющую максимуму либо минимуму всех локальных критериев. В этом случае рекомендацию приходится искать в области компромисса. Математически эта задача не имеет смысла, так как векторный максимум не определен. Его смысл определяется в итоге неформального анализа, в основе которого лежит принцип Парето. Оптимальность по Парето точки F_j относительно векторного критерия $K(F_j)$ означает, что *рекомендация F не может быть улучшеня ни по одному фактору без ухудшения по какому-нибудь другому из них*.

Рассмотрение задачи оптимизации имеет смысл в условиях применения рациональных рекомендаций. Рекомендация может быть охарактеризовано как рациональная, если она удовлетворяет по крайней мере трем требованиям: *непротиворечивости, целенаправленности, транзитивности.*

Непротиворечивость означает, что если лицо, формирующее рекомендацию, предпочитает альтернативу A_1 альтернативе A_2, то в тоже время оно не должно предпочитать альтернативу A_2 альтернативе A_1. Иными словами, если A_1 и A_2 — произвольные альтернативы из множества $A(A_i \subset A)$, то верно одно и только одно из следующих соотношений: $A_1<A_2$, $A_1>A_2$, $A_1 \approx A_2$. Первые два соотношения предпочтения, третье — соотношение безразличия.

Целенаправленность означает, что если альтернативу A_1 предпочитают альтернативе A_2, то при этом следует предпочитать стратегию S_1, приводящую к альтернативе A_1, стратегии S_2, приводящей к альтернативе A_2. Иными словами, если S_1 и S_2 — произвольные стратегии из множества $S(S_i \subset S)$, а A_1 и A_2 — произвольные альтернативы из множества A и $S_1 \rightarrow A_1$, $S_2 \rightarrow A_2$, то при условии $A_1 > A_2$ выполняется условие $S_1 > S_2$.

Транзитивность означает, что если лицо, формирующее рекомендации, предпочитает альтернативу A_1 альтернативе A_2, а альтернативу A_2 — альтернативе A_3, то при этом альтернативу A_1 следует предпочитать альтернативе A_3. Иными словами, если A_1, A_2, A_3 — произвольные альтернативы из множества A, причем $A_1 > A_2$ и $A_2 > A_3$, то $A1_г > A_3$.

Выбор наилучшего варианта возможен только при условии количественной оценки их качества. Определение численных значений представляет собой операцию измерения, предполагающую наличие соответствующей шкалы.

4.4.2. Измерение критериев и выбор шкал

Одной из важнейших задач при оценке формируемых и сформированных рекомендаций является задача измерения рассматриваемых критериев по выбранным шкалам.

В настоящем пункте рассматриваются вопросы измерения критериев, выбранных ЛФР. Будем считать измерением акт присвоения чисел объектам (вещам, предметам или событиям) согласно некоторой системе правил.

Для измерения выделяют три свойства чисел: *тождество, ранговый порядок и аддитивность*. Эти свойства задаются аксиомами A_1—A_9. Тождество обусловлено тремя аксиомами:

A_1. Либо $x = y$, либо $x \neq y$.

A_2. Если $x = y$, то $y = x$.

A_3. Если $x = y$ и $y = z$, то $x = z$.

Свойства рангового порядка обусловливаются двумя аксиомами:

A_4. Если $x > y$, то $y < x$.

A_5. Если $x > y$ и $y > z$, то $x > z$.

Свойство аддитивности обусловливается следующими аксиомами:

A_6. Если $x = y$ и $z > 0$, то $x + z > y$.

A_7. $x + z = z + x$.

A_8. Если $x = y$ и $z = p$, то $x + z = y + p$.

A_9. $(x + y) + z = x + (y + z)$.

Приведенные аксиомы позволяют выделить четыре уровня измерения (перечислены в порядке усиления): *шкалы наименований,* *шкалы порядка, шкалы интервалов и шкалы отношений*. Чем выше уровень шкалы, тем больше математических и статистических операций можно выполнять над полученными при измерении числами.

Определение 1. Множество Y выходов процесса формирования рекомендаций *P* вместе с заданным на нем множеством отношений R_{Y^i}, $i \in I$, $I=\{1, 2, ..., n)$, называется системой с отношениями и обозначается (рис. 4.11):

$$\overline{Y} =\{Y, R_{Y^i}),\ i \in I,\ I=\{1, 2, ..., n\},\ Y=\{y_1, y_2, ..., y_k).$$

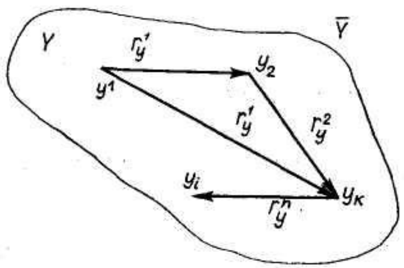

Рис. 4.11. Пример системы с отношениями.

Для практических целей построения моделей будем рассматривать конечные множества.

Определение 2. Отношением эквивалентности на множестве $Y=\{y_1, y_2, ..., y_k\}$ называется бинарное отношение r_y^1, если оно обладает следующими свойствами (рис. 4.12):

1) рефлексивности, т. е.

$$y_i\, r_y^1\, y_i\ \forall_i \in I,\ I=\{1,2,...,k\};$$

2) симметричности, т. е.

$$y_i\, r_y^1\, y_i \leftrightarrow (y_a\, r_y^1\, y_k,\ y_k\, r_y^1\, y_a);$$

$$\forall y_a, y_k \in Y$$

3) транзитивности, т. е.

$$y_a\, r_y^1\, y_k \wedge y_k\, r_y^1\, y_m \rightarrow y_a\, r_y^1\, y_m.$$

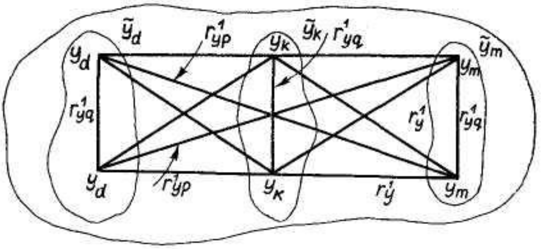

Рис. 4.12. Пример отношения эквивалентности.

Определение 3. Гомоморфизмом системы с отношениями Y на систему такого же типа $\overline{N}=(N_Y, R_{N^i}, i\in I, \mathrm{I}=\{1, 2,..., n\})$ называется такое отображение γ, при котором для всех $i\in I$ и $(y_1 ..., y_{ki})\in Y^{ki}$ имеет место

$$R_{Y^i}\ (y_1 ..., y_{ki}))= R_{N^i}\,(\gamma(y_1), \gamma(y_2),\ldots, \gamma(y_{ki}))\ .$$

Отображение γ будет гомоморфизмом системы $\overline{Y}$ в $\overline{N}$ тогда и только тогда, когда для всех $i\in I$ имеет место (рис. 4.13)

$$R_{Y^i} = \gamma_{ki}^{-1}(R_{N^i}).$$

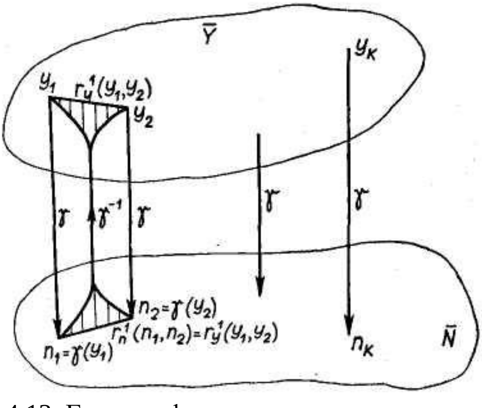

Рис. 4.13. Гомоморфизм системы с отношениями.

ШКАЛЫ НАИМЕНОВАНИЙ

Логическая основа шкал определена выше в аксиомах A_1—A_3. Построить шкалу наименований — это значит использовать число как название.

Свойство тождества чисел заключается в том, что две рекомендации либо тождественны, либо различны, а также, что рекомендации, равные одной и той же рекомендации, равны между собой. Аксиома 2 гласит, что отношение равенства симметрично.

Свойство тождества чисел используется для порождения бесконечного набора различных названий. Названием может быть и порядковый номер.

Можно нумеровать консультационные документы, показатели состояния процесса, действия и т. д. Это не будет значить ничего иного, кроме того, что каждый отдельный предмет должен иметь свое обозначение. Шкалы наименований используются для распознавания различных объектов или процессов.

Шкалы наименований по существу качественны, однако они допускают некоторые статистические операции. Можно сосчитать число представителей каждого класса, определить их частоту и найти наиболее многочисленный класс.

Даже такая простейшая шкала, как шкала наименований, позволяет осуществлять выбор, высказывая предпочтение. Так, если потенциальный пассажир находится на остановке, общей для двух трамвайных маршрутов, то он воспользуется не любым трамваем, а только тем, который ему нужен (предполагается несовпадение маршрутов). Однако выбор осуществляется не потому, что трамвай одного маршрута лучше другого, а потому, что его маршрут совпадает с маршрутом движения человека. Здесь выбор осуществляется на основании первых трех аксиом.

Отношения элементов шкалы наименований показаны на рис. 4.14.

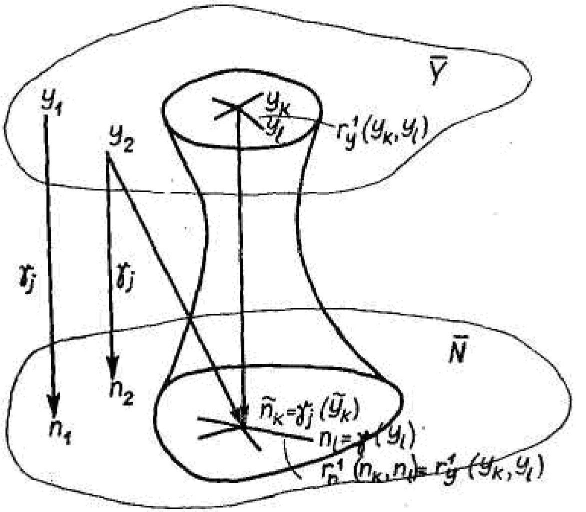

Рис. 4.14. Отношения в шкале наименований.

Определение 4. Взаимно-однозначное отображение системы (Y, r_y^1) в систему (N, r_n^1), где r_y^l — отношение эквивалентности, а r_n^1 — отношение равенства на множестве натуральных чисел N, называется шкалой наименований. На рис. 4.14 система с отношениями

$\bar{Y} = (Y, r_y^1)$ на множестве Y имеет только одно отношение — отношение эквивалентности r_y^1, которое отображается гомоморфно в отношение r_n^1.

ШКАЛЫ ПОРЯДКА

Первое усиление шкалы наименований происходит в том случае, когда мы имеем возможность сравнивать две рекомендации по одному общему признаку. Например, если рекомендации охарактеризованы по объему, то можно установить, какой из них больше. Согласно аксиоме A_4, упорядочивающее отношение r^2 асимметрично, т. е. $y_k r_{yi}^2 y_m$ обозначает, что рекомендация y_k по какому-то признаку i превосходит рекомендацию y_m. В случае простого порядка отношение $y_m r^2 y_k$ не имеет места, т. е. $y_m \tilde{r}^2 y_k$.

Если сравнивать таким образом каждую пару вещей в длинном перечне и если каждая тройка обнаруживает транзитивность (см. аксиому A_5), то можно построить шкалу простого порядка. Например, можно пронумеровать объемы по их величине, присваивая большему объему больший номер.

Так можно упорядочивать (ранжировать) людей по росту либо возрасту, автомашины по грузоподъемности. Под упорядочением здесь следует подразумевать только расстановку в определенной последовательности по убыванию либо возрастанию данного признака без учета количественной разницы.

Следует иметь в виду одно важное, упускаемое многими консультантами требование: неизменность условий среды s = =const. Иногда при упорядочении возникают трудности, связанные с нарушением транзитивности, например: команда *A* выиграла у команды *B,* команда *B* выиграла у команды *C,* но команда *C* выиграла у команды *A.* Здесь ошибка заключается в том, что учтено нарушения указанного выше требования. Условия каждой игры различны. Различны погодные условия, психофизическое состояние игроков, тактика игры и т. д. Результаты предыдущих игр могут вырабатывать устойчивые психологические мотивы действий, во многом определяющие результаты игры. Команда *A,* играющая с командой *B,* по своим качественным показателям отличается от команды *A* в игре с командой *C* даже при неизменном составе.

Можно привести пример аналогичного нарушения условия s=const из области обычных измерений длины нескольких образцов, выполненных из материалов с различным коэффициентом теплового расширения. Производя попарное сравнение таких образцов в условиях изменяющейся температуры окружающей среды, можно также получить впечатление нарушения любой из аксиом 3, 5, 6, 8. Особенно необходимо иметь это в виду при анализе случайных величин, каждое значение которых определяется изменениями условий среды. Рассмотренный выше пример, является ярким тому подтверждением.

Во многих практических случаях допускают равную оценку. При этом элементы образуют слабый порядок. Например, для шкалы слабого порядка возможно $A = B$, $B = C$ и $A>C$. Такой слабый порядок имеет место, когда мы не можем различить по громкости ни звуки *A* и *B,* ни звуки *B* и *C.* Но интервал между *A* и *C* различим. Рассмотренную ситуацию не следует считать субъективной, определяемой чувствительностью наших чувств. Наоборот, она отражает объективные условия, в которых устанавливается порядок предпочтения. Если бы человек не обладал органом слуха, то отпала бы необходимость, например, ранжировать автомобили по издаваемому ими шуму.

Таким образом, рекомендации, отображенные на шкалу простого

порядка, должны быть сравнимы и транзитивны по общему признаку.

В схемах простого порядка каждый элемент должен иметь более высокий или более низкий ранг, чем всякий другой элемент: частота появления любого подмножества равна единице. Во многих практических случаях допускается равная оценка появления классов, чтобы не проводить различий за границами наблюдений. При этом частота появления подмножества эмпирических объектов может быть больше единицы. Элементы, измеренные по такой шкале, образуют слабый порядок.

Логическое основание слабого порядка заключается в двух отношениях: *антисимметрии и транзитивности*. Примером антисимметрии служит отношение «≥» для действительных чисел. В словесной форме отношение антисимметрии выражается высказываниями вида: «y_k по меньшей мере так же хорошо, как и y_d». Для слабого порядка аксиомы A_4 и A_5 можно записать так:

A'_4. Либо $x \geq y$, либо $y \leq x$.

A'_5. Если $x \geq y$ и $y \geq z$, то $x \geq z$.

Если при этом ни $x \geq y$, ни $y \leq x$, то говорят, что x и y несравнимы. Если одновременно $x \geq y$ и $y \geq x$, то получается рефлексивное отношение $x=y$. Отметим, что несравнимость не всегда то же самое, что безразличие. В случае, когда некоторые элементы перечня несравнимы по упорядочивающему отношению, а остальное подмножество допускает сравнение, имеет место так называемый частичный порядок.

Аксиомы упорядочения допускают те же статистические операции, что и аксиомы тождества, т. е. получение частот и мод. Ранговый порядок позволяет вычислять медианы, центили и коэффициенты ранговой корреляции, оценивающие степень близости упорядочений в сравниваемых множествах.

Арифметические и другие статистические операции, кроме

перечисленных выше, исключаются, так как объекты на шкалах порядка не обязательно располагаются равномерно.

С помощью операций измерения c_1 , c_2, ..., c_j можно получить на множестве чисел N множество шкал порядка, гомоморфно отображающих систему с отношениями

$$\overline{Y} = (Y, r_y^1, r_y^2)$$

в систему с отношениями

$$\overline{N} = \{N, r_n^1, r_n^2),$$

где r_n^2 — отношение слабого порядка (рис.4.15).

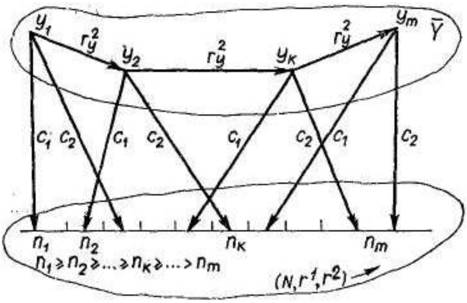

Рис.4.15.Отношения в шкале порядка

Определение 5. Шкала П : Y→N называется шкалой порядка, если она единственна с точностью до монотонности возрастающих непрерывных отображений множества $\gamma(Y)$ в множество N.

ШКАЛЫ ИНТЕРВАЛОВ

Шкала интервалов в отличие от шкалы порядка равномерная. При ее использовании численно равные разности шкалы измеряют численно равные разности измеряемого признака. Иными словами, интервалы между точками шкалы порядка сами могут быть упорядочены, отсюда еще одно название — дважды упорядоченные шкалы. Поскольку шкала интервалов представляет собой упорядоченную шкалу порядка, то ее логической основой являются все те же аксиомы, что и для шкалы порядка.

Примером шкал интервалов являются температурные шкалы. Помимо всех свойств шкалы порядка здесь применимо еще одно: разность между 35 и 36° и разность между 96 и 97°, измеренная в одних и тех же единицах, одна и та же. Шкала интервалов так же, как и шкала порядка, не допускает арифметические операции. Если, например, даны температуры 200 и 50F, то можно было бы неосторожно принять, что первая в 4 раза выше второй и что отношение останется тем же при переходе к другим шкалам (Цельсия, Кельвина), но те же температуры по шкале Цельсия составляют 88 и 18°, т. е. отношение изменилось. Это обусловлено тем, что различные температурные шкалы имеют различные нулевые точки.

Свойства шкал интервалов определяются аксиомами A_1—A_5 в случае, если упорядоченное множество представляет собой множество действительных чисел. При этом оказывается возможным упорядочить интервалы между точками шкалы порядка. Иногда шкалы интервалов называют дважды упорядоченными шкалами, т. е.

$\gamma(y_k)-\gamma(y_m)=\gamma(y_i)-\gamma(y_n)= \ldots = y_k — y_m = y_i — y_n= \ldots =n_k — n_m = n_i — n_n= \ldots$

Шкалы интервалов не обладают важным свойством аддитивности, которое определено аксиомами A_6—A_9. Это значит, что на шкалах интервалов нельзя применять ни одно из основных арифметических действий, поскольку *вычитание, умножение и деление есть частные случаи сложения.*

В ряде случаев на шкалах интервалов выбирают произвольный нуль, что позволяет рассматривать разности как абсолютные величины, обладающие свойством аддитивности. Календарное время и высота над уровнем моря суть шкалы интервалов, но с ними обычно обращаются как со шкалами отношений, так как общепринято соглашение о нуле (уровень моря и т. п.).

На шкалах интервалов можно выполнять те же операции, что и на шкалах наименований и порядка, а кроме того, процедуры вычисления математического ожидания, стандартного отклонения, коэффициента асимметрии и смешанных моментов. Не имеет смысла только одна операция: определение отношения стандартного отклонения к математическому ожиданию, так как эта величина зависит от положения нулевой точки.

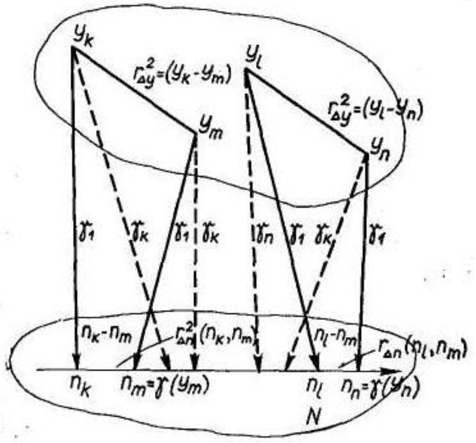

Рис. 4.16. Отношения в шкале интервалов.

На рис. 4.16 символом $r_{\Delta y^2}$ обозначено отношение порядка на множестве различий по некоторому качеству *y,* что соответствует случаю, когда

$$r_{\Delta n^2}\,(n_k, n_m) = r_{\Delta n}\,(n_l, n_m),$$

т. е. имеет место

$$r^1\,(r_{\Delta n^2}\,(n_k, n_m)\,, r_{\Delta n^2}\,(n_l, n_m))\,.$$

ШКАЛЫ ОТНОШЕНИЙ

Шкала отношений обладает всеми свойствами описанных шкал, а также важным свойством аддитивности, определяемым аксиомами A_6—A_9. Изменение шкалы не изменяет отношения результатов одного измерения к другому. Другими словами, на шкале отношений величина y подвергается преобразованию

$$y=cx,$$

где c — любой ненулевой скаляр.

Нуль шкалы отношений естествен. Вес, объем, давление, скорость и другие физические величины измеряются по шкалам отношений. На шкалах отношений можно выполнять все арифметические операции.

Определение 6. Группа Q_p положительных линейных преобразований из N на N состоит из всех преобразований вида

$$q_{\alpha,\beta}: y \to \alpha y+\beta,$$

где $\alpha \in N^+$, $\beta \in N$ (N^+ — множество всех положительных чисел).

Группа $Q=\{q_{\alpha,0} : \alpha \in N^+\}$ называется группой растяжений, а $Q_s=\{q_{1,\beta} : \beta \in N\}$ — группой сдвигов.

Определение 7. Шкала называется шкалой интервалов, если она единственна с точностью до положительных линейных преобразований (растяжений). Пример шкалы отношений приведен на рис. 4.17.

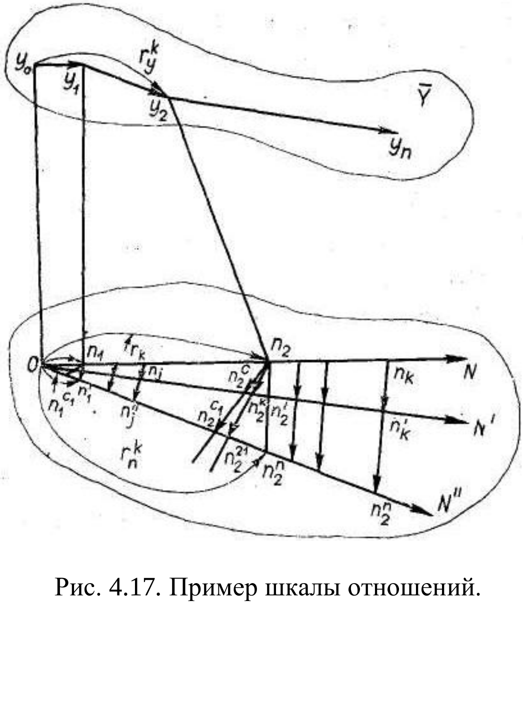

Рис. 4.17. Пример шкалы отношений.

Таким образом, важно отметить, что рассмотренные шкалы позволяют отобразить на множество чисел определенные отношения на множестве свойств и характеристик рассматриваемых рекомендаций. В этой связи особый интерес представляет способность ЛФР упорядочить множество рекомендаций по множеству показателей.

Все рассмотренные шкалы являются одномерными, позволяющими измерять простые свойства предметов и явлений. Во многих белее сложных случаях явления могут быть описаны рядом характеристик, каждая из которых требует своей шкалы измерения. При этом общую оценку явления на базе отдельных оценок его характеристик называют вектором. Векторы измеряют многомерными шкалами.

Полное электрическое сопротивление Z, состоящее из активного сопротивления R и реактивного X, является векторной величиной. В этом случае оба компонента измеряют по шкалам отношений. Вектор полного сопротивления записывают в виде $Z=R + jX$, где j — указывает на ортогональность осей направлений шкал компонент.

Величины компонент независимы. Ввиду свойств ортогональности и аддитивности полное сопротивление можно записать длиной вектора $Z = \sqrt{R^2 + X^2}$ и углом между вектором и осью активного сопротивления $\varphi = arctg\backslash(X/R)$. Заметим, что и в этом случае для задания вектора необходимы два числа.

Характерной особенностью многомерных шкал является нарушение аксиомы 4, говорящей о том, что сравнивать можно только сравнимое. Аксиома сравнимости не может быть удовлетворена для вектора, если только не наделить компоненты относительными весами, обеспечивающими транзитивное ранжирование. В качестве таких весов могут выступать маргинальные коэффициенты замещения между i -м критерием и фиксированным опорным локальным критерием.

Сложность определения относительных весов (коэффициентов весомости, важности), отсутствие их аналитического выражения затрудняют практическое использование многомерных шкал при формировании аналитической формы функции цели. Решение может быть упрощено, если воспользоваться единой мерой для оценки всех факторов и альтернативы в целом. Такой мерой может служить *полезность*.

4.4.3. Поиск схемы оценки формируемых и сформированных рекомендаций

Здесь мы рассмотрим вопросы, относящиеся к проблеме оценки формируемых и сформированных рекомендаций на основе математических методов. Построение моделей оценки рекомендаций, в наибольшей степени адекватных поведению ЛФР, позволит повысить эффективность решения задач консультируемых проблем.

Под *поведением ЛФР* будем понимать его реакцию на состояние консультационного процесса в смысле выбора (формирования) наилучшей альтернативы для достижения целей функционирования консультируемой проблемы .

Множество оценок Z выхода Y консультационного процесса *P* или состояний консультируемой проблемы состоит из величин, измеренных в различных шкалах. Это не позволяет без дополнительных преобразований выполнять адекватные арифметические и другие операции без искажения информации, заложенной при первичных измерениях.

Одной из эффективных процедур, позволяющих устранить разнородность шкал и разный масштаб измерения показателей *Z*, является нормализация. Пусть множество показателей *Z* конечно и фиксированно. Возможность добавления хотя бы одного элемента к *Z* не рассматривается. Тогда общая схема процедуры нормализации может быть представлена следующим выражением:

$$\overline{Z}_i = \frac{z_i - z_i^{\min}}{z_i^{\max} - z_i^{\min}},$$

где $\overline{Z}$ — нормированное значение *i*-го показателя *z;*

z_i — нормируемое значение *i*-го показателя *z;* $z_i^{\min}$ — минимальное значение *i*-го показателя *z;* $z_i^{\max}$ — максимальное значение *i*-го показателя *z*. При этом наибольшее значение *z* для *i*-го показателя отобразится на множестве чисел в единицу, а наименьшее — в нуль, независимо от масштаба и типа шкалы измерения.

На рис. 4.18 показано, как состояния процесса P_1 и P_2 с помощью операций измерения $c_1, c_2, ..., c_k$ отображаются на шкалы $N_1, N_2,..., N_k$ для соответствующих показателей $z_1, z_2, ... z_k$.

С помощью операции нормализации γ разнородные шкалы отображаются на числовой интервал [0, 1] множества действительных чисел *N*.

В результате операции нормализации для *m* возможных альтернатив $A_1, A_2,...,A_m$ получается *k* числовых оценок $z_1, z_2, ... z_k$, соответствующих каждой оцениваемой альтернативе.

Оценка z_{ij} соответствует i-й альтернативе по j-му показателю (табл. 4.5).

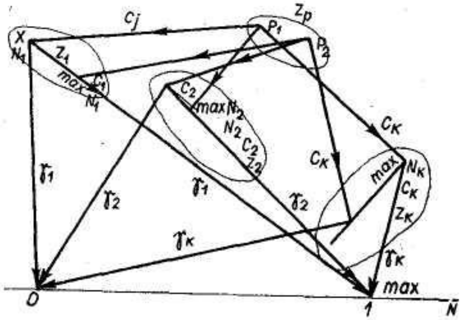

Рис. 4.18. Пример измерения состояния консультационного процесса.

Существует несколько критериев выбора (формирования) оптимальной рекомендации для альтернатив, характеризующихся матрицей оценок, аналогичных приведенным в табл. 4.5. Рассмотрим некоторые из них.

Максимальный критерий Вальда, или принцип гарантированного результата, в качестве оптимальной выбирает такую альтернативу *A,* при которой максимальное значение нормированного показателя максимально:

$$W = \max_{i} \min_{j} \bar{z}_{ij}\ \mathrm{j}.$$

Таблица 4.5

Матрица оценок альтернатив

Характеристика	Показатели			
	z_1	z_2…	z_j…	z_k
a_1	z_{11}	z_{12}…	z_{1j}…	z_{1k}
a_2	z_{21}	z_{22}…	z_{2j}…	z_{2k}
…	…	… …	… …	…
a_i	z_{i1}	z_{i2}…	z_{ij}…	z_{ik}
…	…	… …	… …	…
a_m	z_{m1}	z_{m2}…	z_{mj}…	z_{mk}

Если ЛФР действует согласно этому критерию, то при равном приоритете всех показателей выбирается такая альтернатива, для которой «наихудший» показатель является «наилучшим» по сравнению с другими.

Пользуясь критерием Вальда, ЛФР обеспечивает для себя наилучшие характеристики в случае пессимистической оценки обстановки и больше внимания уделяет слабым показателям и неуспехам, чем достоинствам и сильным сторонам характеризуемого процесса.

Рассмотрим пример оценки рекомендаций по критерию Вальда. В табл. 4.6 приведены оценки альтернатив по трем показателям.

Таблица 4.6

Пример для оценки альтернатив по трем показателям

Альтернативы	Показатели		
	z_1	z_2	z_3
a_1	1,0	0,3	0,6
a_2	0	0,2	0,4
a_3	0,6	0,5	0,4
a_4	0,8	1,0	0,2

На первом этапе находится минимальное значение показателя для каждой строки и выписываются в столбец значения, соответствующие min $\bar{z}_{ij}$. На втором этапе определяется максимальный элемент в столбце min $\bar{z}_{ij}$. Для первой альтернативы это значение показателя z_2 равное 0,3. Для второй альтернативы a_2 минимальное значение, равное нулю, будет у первого показателя z_1. Для альтернативы a_3 на втором этапе замечаем, что минимальное нормированное значение показателя z_3, равное 0,4, является наибольшим по сравнению с другими альтернативами. Согласно критерию Вальда, a_3 определяется как оптимальная альтернатива, которая лучше альтернативы a_1. В свою очередь, альтернатива a_1 лучше альтернативы a_4 Наконец, альтернатива a_4 лучше, чем альтернатива a_2 (табл. 4.7).

Таблица 4.7

Пример оценки альтернатив по критерию Вальда

Альтерна-тивы	Показатели			Критерий Вальда		
	z_1	z_2	z_3	min $\overline{z}_{ij}$	maxmin $\overline{z}_{ij}$	Ранг
a_1	1,0	0,3	0,6	0,3	-	2
a_2	0	0,2	4	0	-	4
a_3	0, 6	0,5	0,4	0,4	0,4	0,4
a_4	0, 8	1,0	0,2	0,2	-	3

Критерий Гурвица, иначе называемый критерием пессимизма-оптимизма, рекомендует при оценке альтернатив не руководствоваться ни крайним пессимизмом (всегда рассчитывать на худшее стечение обстоятельств и обращать внимание только на недостатки), ни крайним легкомысленным оптимизмом (считать что все обойдется лучшим образом, и обращать внимание только на успехи и достоинства альтернативы). Критерий Гурвица имеет вид

$$H = \max_i \{\alpha \min_j \overline{z}_{ij} + (1-\alpha) \max_j \overline{z}_{ij} \qquad (4.18)$$

где α — коэффициент, значение которого заключено между нулем и единицей. Анализ выражения критерия Гурвица показывает, что при α=1 критерий Гурвица превращается в пессимистический критерий Вальда, а при α = 0 — в критерий «крайнего оптимизма». Последний рекомендует выбирать альтернативу, для которой существует такая наилучшая оценка показателя, что ее «не могут превзойти характеристики других альтернатив». При 0<α<1 получаем что-то среднее между крайним пессимизмом и крайним оптимизмом. Можно заметить, что коэффициент α обозначает как бы степень пессимизма ЛФР. Этот коэффициент назначается ЛФР из субъективных соображений и получает значение тем ближе к единице, чем больше ЛФР хочет «подстраховаться» в неприятных и опасных ситуациях.

Для альтернатив, заданных табл. 4.6, при α = 0,5 в соответствии с критерием Гурвица принимается, что в равной степени рассматриваются как «сильные», так и «слабые» стороны альтернатив. При этом вычисленный по формуле критерий Гурвица принимает наибольшее значение, равное 0,65, для альтернативы a_1:

$$H(a_1) = \max \{0{,}5\min z_{1j} + (1—0{,}5)\ max\ z_{1j}\text{-}\} = \max_1 \{0{,}5•0{,}3+0{,}5•1\} = 0{,}65.$$

Значения критерия Гурвица и ранги для других альтернатив приведены в табл. 4.8.

Таблица 4.8

Пример оценки альтернатив по критерию Гурвица

Аль-терна-тивы	Показатели			Критерий Гурвица при α=0,5						
	z_1	z_1	z_1	min $\bar{z}_{ij}$	max $\bar{z}_{ij}$	αmin $\bar{z}_{ij}$	(1-α)× ×max $\bar{z}_{ij}$	H	maxH	Ранг
a_1	1,0	0,3	0,6	0,3	1	0,15	0,5	0,65	0,65	1
a_2	0	0,2	0,4	0	0,4	0	0,2	0,20	-	4
a_3	0, 6	0,5	0,4	0,4	0,6	0,20	0,3	0,50	-	3
a_4	0,8	1,0	0,2	0,2	1	0,10	0,5	0,60	-	2

При сравнении результатов оценки альтернатив по критериям Гурвица и Вальда отметим различие в рангах для альтернатив a_1 и a_3. По критерию Вальда альтернатива a_3 предпочтительнее, чем альтернатива a_1. По критерию Гурвица уже не только альтернатива a_1, но и a_4 лучше, чем a_3.

По критерию Лапласа учитываются не только «лучшие» и «худшие» значения показателей оцениваемых альтернатив, но и все остальные, промежуточные значения. При этом оценка каждой альтернативы приводится как бы к среднему значению, обобщающему значения всех показателей для каждой альтернативы. Это среднее значение альтернативы вычисляется по формуле

$$S_j = \frac{1}{n}\sum_{j=1}^{n} z_{ij}$$

где n — общее число показателей для i-й альтернативы.

В табл. 4.9 приведены оценки для альтернатив, вычисленные по критерию Лапласа.

Наибольшее значение критерия Лапласа (S = 0,67) получает альтернатива a_4:

$$S_j = \frac{1}{n}\sum_{j=1}^{n} z_{ij} = \frac{1}{3}(0{,}8 + 1 + 0{,}2) \approx 0{,}67.$$

Таблица 4.9

Пример оценки альтернатив по критерию Лапласа

Альтернативы	Показатели			Критерий Лапласа		
	z_1	z_2	z_3	$\sum z_{ij}$	s_i	Ранг
a_1	1,0	0,3	0,6	1,9	0,65	2
a_2	0	0,2	0,4	0,6	0,20	4
a_3	0, 6	0,5	0,4	1 ,5	0,50	3
a_4	0,8	1,0	0,2	2,0	0,67	1

Отметим при этом, что альтернатива a_4 лучше a_1, которая, в свою очередь, лучше, чем a_3. Таким образом, для каждого критерия выбора получается своя лучшая альтернатива, оказывающаяся не лучшей по другим критериям (табл. 4.10).

На практике имеет место случай, когда ЛФР предъявляется множество альтернатив и становится известен результат выбора лучшей. Для примера, приведенного в табл. 4.10, будем считать, что установлено только три вышеуказанных критерия оценки альтернатив, которыми может пользоваться ЛФР, но заведомо неизвестно, каким именно критерием оно пользовалось.

Таблица 4.10

Пример сравнения критериев оценки альтернатив

Альтер-нативы	Показатели			Ранги по критериям				
							ЛФР	
	z_1	z_2	z_3	Вальда V	Гурвица H	Лапласа S	Случай 1	Случай 2
a_1	1,0	0,3	0,6	2	1	2	2,4	1,2
a_2	0	0,2	0,4	4	4	4	2,4	4
a_3	0, 6	0,5	0,4	1	3	3	1	3
a_4	0,8	1,0	0,2	3	2	1	2,4	1,2

Если ЛФР выбрало одну альтернативу a_3, а остальные альтернативы остались неупорядоченными (случай 1 в табл. 4.10), то альтернативе a_3 присвоен ранг 1, а всем остальным — 2, 3 и 4, т. е.

средний ранг равен 3. В первом приближении можно утверждать, что критерий, по которому ЛФР выбрало альтернативу, ближе к критерию Вальда, чем к другим.

Когда ЛФР выбрало как наилучшие и равные между собой альтернативы a_1, и a_4, а затем определило, что a_3 лучше, чем a_2 (случай 2 в табл. 4.10), то в первом приближении можно утверждать, что при определении схемы оценки рекомендаций ЛФР пользовалось скорее критериями Гурвица и Лапласа, чем Вальда.

Задача оценки адекватности поведения ЛФР по упорядочению множества альтернатив и схем оценки рекомендаций реализуется с помощью методов ранговой корреляции. Это позволяет алгоритмическим путем определить схему оценки рекомендаций для конкретной ситуации выбора и конкретного ЛФР, используя методы векторной оптимизации.

4.4.4. Критерий качества в условаях неопределенности и риска

Современная теория полезности содержит совокупность непротиворечивых допущений (аксиом) о множестве решений и свойствах предпочтений лица, формирующего рекомендации, а также совокупность утверждений (теорем), которые можно вывести на основе этих допущений. С помощью теории полезности информация о предпочтениях лица, формирующего рекомендации, преобразуется в соответствующую численную информацию о полезности. Затем численная информация используется для вычисления полезностей реальных альтернатив. Результаты сравнения численных значений полезностей вновь преобразуются в суждении о предпочтении.

Основоположниками математической теории полезности являются Дж. фон Нейман и О. Моргенштерн, чьи аксиомы «рационального поведения» стали основой этой теории. Воспользовавшись этими аксиомами, можно показать, что существует мера ценности, позволяющая упорядочить предполагаемые результаты сформированных рекомендаций. *Эту меру ценности называют полезностью.* Доказано, что если лицо, формирующее рекомендации, строит свои действия на основе определенных допущений, то на множестве предполагаемых исходов возможных действий может быть определена функция полезности, связывающая это множество с множеством чисел.

Одной из наиболее важных концепций при использовании теории полезности в задачах формирования рекомендаций с векторным критерием является понятие независимости или аддитивности полезностей (критериев). При этом должно выполняться требование, состоящее в том, что лицо, формирующее рекомендации, может выразить согласованное суждение о величинах любого одного критерия, когда оценки по всем другим критериям фиксированы. При этом его суждения не должны зависеть от значений фиксированных оценок по другим критериям.

Существует *теория аддитивной полезности*. В большинстве случаев можно сформулировать локальные критерии таким образом, чтобы выполнялось условие аддитивности.

Аксиомы фон Неймана-Моргенштерна гласят, что $x \leq y$ тогда и только тогда, когда $u(x) \leq u(y)$, т. е. *отношение предпочтения — безразличия относительно х и у определяется соответствующим бинарным отношением, связывающим их полезности*.

Для формирования рекомендаций можно использовать игровую трактовку ситуации: *лицо, формирующее рекомендации, рассматривают как арбитра в игре, в которой каждый фактор выступает как игрок, стремящийся к максимальному выигрышу*. Если критерии количественно несоизмеримы, предполагают, что каждый из факторов является игроком, стремящимся максимизировать свою функцию полезности. Этому случаю соответствует некооперативная игра.

Так как рекомендация будет одобрена только тогда, когда все игроки выбирают одну и ту же стратегию $S_k \subset S$, то за выбор согласованной стратегии их следует поощрять. Выигрыш при исходе S_k для i-го игрока определяется выражением

$$u_i(S) = \begin{cases} u_i(s) & \text{при } S = S_k \text{ для всех } i; \\ \min\limits_{\{i\}} u_i(s) & \text{при } S \neq S_k \text{ для всех } i. \end{cases}$$

Эта игра решается на основании теоремы Нэша, но не всегда в чистых стратегиях, и является множеством неподчиненных равновесных ситуаций, получаемым из множества всех ситуаций равновесия удалением тех, которые хуже одновременно для всех игроков. К сожалению, выбор выигрыша з данном случае осуществляется волевым путем.

Для случая, когда критерии частично сравнимы, т. е. когда возможно их объединение в группы с последующим сравнением, пытаются описать ситуацию с позиций кооперативной игры игроков-факторов.

При формировании рекомендаций в условиях риска предполагают, чту каждое действие приводит к одному исходу из множества возможных, причем формирующему рекомендации известны вероятности этих исходов. Формирование рекомендаций в условиях риска еще не исследовано. Для исследования формирования рекомендаций в условиях риска можно воспользоваться теорией ожидаемой полезности фон Неймана-Моргенштерна. При определенных допущениях, форма критерия определяется выражением

$$u_j = \sum_i p(F_{ij})\, u(F_{ij}), \tag{4.19}$$

где F_i — i-е значение одного фактора; $p(F_i)$ — вероятность получения этого значения. Аналогичные результаты можно получить, используя теорию статистических решений, базирующуюся на теории статистических игр.

В ряде случаев вероятность наступления предполагаемых событий неизвестна и должна быть измерена. Мерой уверенности лица, формирующего рекомендации, в том, что наступит тот или иной исход, являются так называемые *субъективные вероятности*. При определении субъективных вероятностей используют два подхода. При интуитивном подходе к множеству событий или исходов применяют аксиомы сравнительного вероятностного отношения «не более вероятно, чем». Другой подход основан на аксиомах отношения предпочтения — безразличия («не предпочтительнее, чем»). Каждый из этих подходов дает возможность измерить вероятности и определить функцию полезности для получения субъективной вероятностной модели полезности, согласованной с отношением предпочтения — безразличия лица, формирующего рекомендации.

Однако во всех случаях речь идет в основном об однофакторных задачах. Это послужило причиной того, что используя теорию статистических игр для формирования рекомендаций, лица, формирующие рекомендации, вынуждены все сводить к одному фактору — затратам. Результат— чрезвычайное обеднение рекомендации в тех случаях, когда ряд факторов не имеет явного стоимостного эквивалента либо он трудно определим. Кроме того, выражение (4.19) имеет серьезный недостаток, общий для выражений подобного типа, заключающийся в снижении вероятности эффективности рекомендации при однократном применении.

Для облегчения измерения полезности американский статистик Харрингтон предложил приближенное аналитическое выражение зависимости полезности от уровня фактора в виде

$$u = \exp[\, - \exp(\, - F)]. \tag{4.20}$$

Это выражение получено как усреднение большого числа экспериментальных кривых полезности. Совмещение шкалы измерения конкретных факторов со шкалой F осуществляется экспертами.

При использовании выражения (4.20) $1 \geq u \geq 0$. При этом стандартные отметки на шкале полезности таковы: $1{,}00 \geq u \geq 0{,}80$ — очень хорошо; $0{,}80 > u \geq 0{,}65$ — хорошо; $0{,}63 > u \geq 0{,}37$ — удовлетворительно;

$0{,}37 > u \geq 0{,}20$ — плохо; $0{,}20 > u \geq 0$ — очень плохо.

Свертка полезностей факторов в обобщенную функцию полезности осуществляется выражением

$$u = \sqrt[k]{\prod_{i=1}^{n} u_i},$$

не дающим в большинстве случаев, как будет показано ниже, адекватной модели.

Указанные недостатки рассмотренных методов в значительной степени ограничивают возможности их применения, а во многих случаях приводят к ошибочным выводам. Попытаемся улучшить существующие решения.

Пусть задано множество альтернатив, каждая из которых характеризуется факторами $F_1, F_2, \ldots, F_n$, область значений которых определена. Совокупность значений факторов обеспечивает определенную полезность варианта u_j (F_{ij}, $i = 1, 2, \ldots, n$) на множестве F, допускающую измерение.

Задача формулируется следующим образом. *Из конечного множества альтернатив (рекомендаций) $A = \{A_j, i = 1, 2, \ldots\}$ выбрать наилучшую, обеспечивающую максимизацию (минимизацию) оценочного функционала u_j (F_{ij}, $i = 1, 2, \ldots, n$) при известном законе распределения вероятностей $p_{ij} = \{p_{ij}, r = 1, 2, \ldots\}$ взаимоисключающих значений факторов каждого варианта $F_{ij} = = \{F_{ij}, r = 1, 2, \ldots\}$*. При этом значения факторов составляют полную группу, т. е.

$$\sum_{r=1}^{N_i} p_{rij} = 1,$$

где N_i — число возможных значений r -го фактора j-го варианта.

В такой формулировке ситуация может быть представлена как *игровая*. Поскольку одним из «противников» является природа, речь идет о решении задачи на базе статистических игр (статистических решений). Однако в отличие от обычных статистических решений, рассматривающих простое множество состояний природы, в данном случае задача осложняется наличием множества факторов, принимающих множество значений. Поэтому сформулированную задачу будем называть *множественным (многофакторным) статистическим решением или множественной статистической игрой*. Сложность решения поставленной задачи заключается в отсутствии информации о корреляции полезностей факторов и трудности выбора критерия для решения игры.

Наиболее простым обходом первой трудности является принятие допущения об аддитивности полезностей факторов. Это допущение возможно при формировании рекомендаций, поскольку приводит к наиболее простому процессу формировании рекомендаций. В следующем разделе дано обоснование справедливости принятого допущения при формировании рекомендаций для решения определенного класса консультационных задач — выборе *предпочтительного варианта сформированой рекомендации*.

Другая трудность заключается в существовании достаточно большого разнообразия критериев решения игры, дающих различные результаты, и в отсутствии рекомендаций по их применению. Единственным основанием их применения является согласованность с практикой.

Как будет подтверждено в следующем разделе, при формировании рекомендаций для решения рассматриваемого класса консультационных задач хорошую согласованность с практикой дает критерий Т. Байеса. Однако критерий Т. Байеса, представляющий собой поиск наименьшего (наибольшего) значения математического ожидания, предложен для решения обычной однофакторной игры. Распространение этого подхода на многофакторные игры дает цену игры в виде суммы математических ожиданий оценочного функционала по всем факторам.

Таким образом, решение имеет вид

$$\max_{A_j \subset A}(\min) u_j(F_{ij}) = \max_{A_j \subset A}(\min) \sum_{i=1}^{n} \sum_{r=1}^{N_i} u_{rij}(F_{rij})\, p(F_{rij}). \tag{4.21}$$

Если одинаковое значение полезности u_j достигается в нескольких рекомендациях из A, то такие рекомендации являются равноценными и между ними выполняется отношение безразличия.

Достоинство критерия Т. Байеса по сравнению с другими заключается в тесной связи с аксиомами теории полезности. В этом случае u_{rij} — численное значение полезности i-го фактора при r-м уровне его значения для j-го варианта. Результатом игры может быть как выигрыш, так и проигрыш. Цена игры характеризуется не только величиной, но и знаком (направлением), следовательно, ее можно рассматривать как вектор. Все составляющие вектора цены игры также можно рассматривать как векторы, расположенные на одной прямой. Такое взаимное расположение составляющих определяется тем, что полезность возрастает (убывает) на полную величину составляющих. Благодаря этому суммирование обращается в алгебраическое с учетом знаков. Такой подход позволяет осуществлять формирование функции полезности в соответствии с природой и влиянием каждого фактора.

Пользуясь критерием Т. Байеса, следует иметь в виду, что возможно существование альтернатив, одинаковых по средней полезности, но с разными максимальными и минимальными (даже отрицательными) значениями полезности.

Если лицо, формирующее рекомендации, обладает значительным «богатством» по сравнению с возможным проигрышем, то оно может предпочесть вариант, обладающий возможностью большего проигрыша и выигрыша вопреки выражению (4.21). Отсюда следует, что выражение (4.21) не учитывает шкалу выигрышей и потерь относительно «богатства» консультанта. Поскольку при оценке всегда необходимо исходить из интересов консультируемых клиентов, то степень «богатства» консультанта во всех случаях принимают одинаковой. Это позволяет вводить единую меру оценки сформированных рекомендаций во всех подразделениях народного хозяйства.

Выражение (4.21), основанное на сумме математических ожиданий, следует применять при многократном использовании рекомендаций. Суммарный эффект его многократного применения будет наибольшим. Однако в неблагоприятном случае применение этого критерия может привести к значительным потерям. При этом возможные потери тем больше, чем больше дисперсия факторов. Правда, в отличие от обычной однофакторной статистической игры, в рассматриваемой множественной (многофакторной) игре очень высокая вероятность усреднения полезного эффекта по факторам благодаря тому, что одни факторы принимают меньшие значения, а другие — большие, чем их математические ожидания.

Несмотря на бóльшую вероятность достоверности рекомендации многофакторной игры по сравнению с обычной однофакторной, все же и в этом случае степень риска формирования неправильной рекомендации — проигрыша — достаточно высока. Уменьшения степени риска можно достичь за счет применения критерия пессимизма-оптимизма Гурвича. Однако отсутствие объективных методов определения меры осторожности затрудняет его применение. *В качестве меры осторожности может быть использована величина полезности дисперсии факторов*. В этом случае появляется единый подход к оценке альтернатив, определяемый выражением

$$u_i = \sum_{i=1}^{n} \left[\sum_{r=1}^{N_i} u_{rij}(F_{rij})\, P_{rij}(F_{rij}) - u_{ij}(\sigma_{ij}) \right]. \tag{4.22}$$

Такой подход обеспечивает компромиссное решение, гарантирующее достаточно высокие результаты при ограниченном риске проигрыша. Правда, при этом уменьшается возможность повышения выигрыша за счет выбора альтернатив с большим риском проигрыша.

Строго говоря, *выражение (4.22) позволяет осуществлять выбор альтернатив и с возможным высоким выигрышем при повышенном риске проигрыша.* Для этого достаточно изменить знак минус на плюс.

При консультировании крупных проблемных проектов потеря большого выигрыша (реализация лишь умеренного выигрыша) предпочтительнее большого проигрыша, так как большой проигрыш может лечь тяжелым бременем на экономику (в кризисных условиях возможно даже банкротство фирмы). При сравнительно мелких с точки зрения консультируемой проблемы, но крупных для клиента проблем потеря выигрыша или проигрыш для всего хозяйства почти равнозначны (все же потеря выигрыша несколько предпочтительнее проигрыша). Однако для клиента-предпринимателя проигрыш ведет к потере общественных фондов стимулирования и, естественно, менее предпочтителен, чем потеря выигрыша. Таким образом, в обоих случаях предпочтительность рекомендации пропорциональна уменьшению риска проигрыша. Вероятность большого проигрыша тем выше, чем больше дисперсия. Отсюда следует, что форма выражения (4.22) соответствует как интересам всего хозяйства, так и клиента-предпринимателя.

В тех случаях, когда вероятности неизвестны, мерой уверенности лица, формирующего рекомендации, в том, что наступит тот или иной исход, являются субъективные вероятности, определяемые, как показано выше. В условиях отсутствия информации о законе распределения вероятностей целесообразно исходить из обеспечения условия максимума информационной энтропии, поскольку такой подход гарантирует отсутствие субъективизма. В рассматриваемом случае максимому энтропии соответствует равномерное распределение вероятностей, поэтому

$$u_j = \sum_{i=1}^{n} \left[\frac{1}{N_i} \sum_{r=1}^{N_i} u_{rij}(F_{rij}) - u_{ij}(\sigma_{ij}) \right],$$

где $u_{ij}(\sigma_{ij})$—полезность среднеквадратичного отклонения i-го фактора j-й альтернативы, измеряемая по шкале полезностей факторов;

$$\sigma_{ij}=\sqrt{\sum_{r=1}^{N_i}\{F_{rij}-M[F_{rij}]\}^2 p_{rij}}. \tag{4.23}$$

Математическое ожидание значения i -го фактора j -й альтернативы

$$M[F_{rij}]=\sum_{r=1}^{N_i}F_{rij}p_{rij}. \tag{4.24}$$

Если для всех факторов принять единую шкалу полезности, т. е. обеспечить соответствие $v_{\max}$ и $F_{i_{\max}}$, $v_{\min}$ и $F_{i_{\min}}$, то при монотонном характере функции полезности и $F_{i_{\min}} \le F_i \le F_{i_{\max}}$ получим

$$v_i(F_{i_{\min}}) \le v_i(F_i) \le v_i(F_{i_{\max}})$$

Здесь $v(F)$ в отличие от $u(F)$ — значение полезности, измеренное по выбранной единой шкале. В этом случае выражение (4.22) можно записать в виде

$$u_j(F_{ij})=\sum_{i=1}^{n}\left[\sum_{r=1}^{N_i}w_i v_i(F_{rij})\,p_{rij}(F_{rij})-w_i v_i(\sigma_{ij})\right], \tag{4.25}$$

где

$$w_i=\frac{u_{i_{\max}}}{u_{k_{\max}}}; \tag{4.26}$$

здесь $u_{k_{\max}}$ — максимальное значение полезности фактора, шкала измерения которой принята за общую, т. е. $u_{k_{\max}} = v_{\max}$.

Выражение (4.25) не противоречит лемме о линейном преобразовании полезности и может быть получено непосредственно из нее.

Анализ выражений (4.25) и (4.26) показывает, что коэффициент w_i может быть интерпретирован как коэффициент весомости (важности) i-го фактора по сравнению с другими факторами. Он указывает, во сколько раз полезность одного фактора выше полезности другого фактора. Поскольку векторная оптимизация (по Перето) предполагает ухудшение одних факторов при улучшении других, то этот коэффициент указывает, какие из факторов следует улучшить в первую очередь, т. е. увеличивать при положительной и уменьшать при отрицательной полезности.

Изменение величины факторов, обладающих большим значением коэффициента весомости, оказывает большее влияние на величину полезности варианта.

Из выражения (4.26) следует, что

$$w_i = \frac{a_i F_{i_{\max}} \Delta u_k}{F_{k_{\max}} \Delta u_k} = \frac{a_i F_{i_{\max}}}{F_{k_{\max}}}, \tag{4.27}$$

где

$$\Delta u_k = \frac{u_{k_{\max}}}{F_{k_{\max}}}$$

— полезность единицы максимального значения k-го фактора; a_i — коэффициент перевода единиц i -го фактора в единицы k -го фактора (маргинальный коэффициент замещения между i -м и k-м факторами). Например, если k -й фактор представляет собой затраты, то a_i указывает, сколько «стоит» единица i-го фактора.

Физический смысл коэффициента a_i становится ясным, если вспомнить, как на практике одни физические величины измеряются в единицах других. Например, частота вращения измеряется в об/мин, в вольтах (при использовании тахогенератора), в градусах отклонения стрелки измерительного прибора и т. д.

Из выражения (4.27) следует, что чем выше максимальный уровень фактора, тем выше величина коэффициента важности. В общем случае полезность является нелинейной функцией фактора, и ее определение вызывает значительные трудности. Выражение (4.27) является более содержательным, чем известное выражение:

$$w_i = \frac{\partial u / \partial F_i}{\partial u / \partial F_k},$$

достаточно эффективное в человеко-машинных процедурах. При условии $w_k = 1$ это выражение превращается в тождество вида $w_i \equiv w_k$. Полагая полезность линейной функцией факторов, а коэффициенты весомости — нелинейной, получим

$$u_j(F_{ij}) = \sum_{i=1}^{n} \left[\sum_{r=1}^{N_i} w_i(F_{rij})\, v_i(F_{rij})\, p_i(F_{rij}) - w_i(F_{rij})\, v(\sigma_{ij}) \right]. \tag{4.28}$$

Нелинейность коэффициентов весомости обусловлена зависимостью a_i от уровня фактора для данной альтернативы, т. е.

$$w_i(F_{rij}) = \frac{a_i(F_{rij})\, F_{i_{\max}}}{F_{k_{\max}}}.$$

Полезность факторов может быть записана в виде

$$v_i(F_{rij}) = \frac{v(F_{i_{\max}}) - v(F_{i_{\min}})}{F_{i_{\max}} - F_{i_{\min}}} F_{rij} + v(0), \tag{4.29}$$

а полезность среднеквадратичных отклонений

$$v_i(\sigma_{ij}) = \frac{v(F_{i_{\max}}) - v(F_{i_{\min}})}{F_{i_{\max}} - F_{i_{\min}}} \sigma_{ij}, \tag{4.30}$$

поскольку $v_i(\sigma_{ij} = 0) = 0$. Знаменатель в выражении (4.30) аналогичен знаменателю в выражении (4.29), так как среднеквадратичное отклонение измеряется в единицах i-го фактора.

Так как для определения полезности факторов и среднеквадратичных отклонений при любых значениях $F_{i_{\min}}$ появляется необходимость учитывать v_i $(F_i = 0)$, то удобно принять во всех случаях $F_{i_{\min}} = 0$. Тогда

$$v_i(F_{rij}) = \frac{v(F_{i_{\max}}) - v(0)}{F_{i_{\max}}} F_{rij} + v(0);$$

$$v_i(\sigma_{ij}) = \frac{v(F_{i_{\max}}) - v(0)}{F_{i_{\max}}} \sigma_{ij}.$$

При этом выражение (4.28) примет вид

$$u_j = \sum_{i=1}^{n} \left[\sum_{r=1}^{N_i} w_i(F_{rij}) \frac{v(F_{i_{\max}}) - v(0)}{F_{i_{\max}}} F_{rij} p_i(F_{rij}) + \sum_{r=1}^{N_i} w_i(F_{rij})\, v(0)\, p_i(F_{rij}) - w_i(\sigma_{ij}) \frac{v(F_{i_{\max}}) - v(0)}{F_{i_{\max}}} \right].$$

При измерении полезности по шкале интервалов с условным нулем можно положить $v(0) = 0$, тогда

$$u_j = \sum_{i=1}^{n} \left[\sum_{r=1}^{N_i} w_i(F_{rij}) \frac{v(F_{i_{\max}})}{F_{i_{\max}}} F_{rij} p_i(F_{rij}) - w_i(\sigma_{ij}) \frac{v(F_{i_{\max}})}{F_{i_{\max}}} \sigma_{ij} \right],$$

откуда

$$K_j = \frac{u_j}{v(F_{i_{\max}})} = \sum_{i=1}^{n}\left[\sum_{r=1}^{N_i} w_i(F_{rij})\,p_i(F_{rij}) - w_i(\sigma_{ij})\frac{\sigma_{ij}}{F_{i_{\max}}}\right]. \tag{4.31}$$

Величина K_j представляет собой количественное выражение качества j-й альтернативы, измеренное в относительных единицах. Будем называть ее комплексным либо обобщенным показателем качества. В качестве базовых значений в выражении (4.31) можно выбирать не только максимальные значения, но и любые другие, удовлетворяющие условию

$$\frac{F_{1_б}}{F_{1_{\max}}} = \frac{F_{2_б}}{F_{2_{\max}}} = \ldots = \frac{F_{i_б}}{F_{i_{\max}}} = \ldots \tag{4.32}$$

При этом коэффициенты весомости не изменяют свою величину и определяются выражением (4.27).

Поскольку

$$\left.\begin{aligned} \frac{a_i F_{i_{\max}}}{F_{k_{\max}}}\,\frac{F_i}{F_{i_{\max}}} &= \frac{a_i F_{i_б}}{F_{k_{\max}}}\,\frac{F_i}{F_{i_б}}; \\ \frac{a_k F_{k_{\max}}}{F_{k_{\max}}}\,\frac{F_k}{F_{k_{\max}}} &= \frac{a_k F_{k_б}}{F_{k_б}}\,\frac{F_k}{F_{k_{\max}}}, \end{aligned}\right\}$$

то

$$\sum_{i=1}^{n}\frac{a_i F_{i_{\max}}}{F_{k_{\max}}}\,\frac{F_i}{F_{i_{\max}}} = \frac{F_{k_б}}{F_{k_{\max}}}\sum_{i=1}^{n}\frac{a_i F_{i_б}}{F_{k_б}}\,\frac{F_i}{F_{i_б}}.$$

Следовательно, отбрасывая сомножитель, общий для всех альтернатив, убеждаемся, что выбор базовых значений может быть произвольный, но при этом, если нарушается условие (4.32), изменяется величина коэффициентов весомости в зависимости от базовых значений факторов. В выражении (4.31) вместо максимальных значений будут фигурировать базовые, а коэффициент весомости будет определяться нелинейным выражением

$$w_i(F_i) = \frac{a_i(F_i)\,F_{i_б}}{F_{k_б}};$$

$$K_j = \sum_{i=1}^{n}\left[\sum_{r=1}^{N_i} w_i(F_i)\frac{F_{rij}}{F_{i_б}}\,p_i(F_{rij}) - w_i(\sigma_{ij})\frac{F_{rij}}{F_{i_б}}\right]. \tag{4.32}$$

Пример. Сформировать рекомендации для детальной разработки вариант проекта системы автоматического управления из двух возможных альтернатив, характеризующихся двумя показателями: затратами (первый фактор) и временем переходного процесса (второй фактор).

Значения факторов первого варианта следующие: F_{111}=10 000 грн. с вероятностью p_{111}=0,3; F_{211}=12 000 грн. с вероятностью p_{211} = 0,5;

F_{311}= 14 000 грн. с вероятностью p_{311}= 0,2; F_{121}= 0,5 с с вероятностью p_{121}=0,3; F_{221}= 0,6 с с вероятностью p_{221} = 0,5; F_{321}= 0,7 с с вероятностью p_{321}=0,2. Это расшифровывается следующим образом: затраты на изготовление возможны в сумме (грн.) 10 000, 12 000, 14 000; вероятность таких значений затрат равна 0,3; 0,5; 0,2 соответственно. Вовможные значения времени переходного процесса (с) 0,5; 0,6; 0,7 с вероятностью 0,3; 0,5; 0,2 соответственно.

Значения факторов второго варианта следующие F_{112}=10 000 грн. с вероятностью p_{112} = 0,2; F_{212}= 12 000 грн. с вероятностью p_{212} = 0,5; F_{312}=14000 грн. с вероятностью p_{312} = 0,3; F_{122}= 0,6 с с с вероятностью p_{122} = 0,5; F_{222}= 0,7 с с вероятностью p_{222}=0,5.

Коэффициенты весомости факторов не зависят от их уровня и равны

$$w_1 = 0{,}4; \text{ и } w_2 = 0{,}6.$$

На основании выражений (4.23) и (4.24) определяем математическое ожидание факторов, а затем их среднеквадратичное отклонение σ_{11}=1400 грн.; σ_{21}= 0,07 с; σ_{12}=l400 грн.; σ_{12}= 0,15 с. На основании выражения (4.33) определяем значения

$$K_1 = -\frac{0{,}4}{14\,000}(10\,000\cdot 0{,}3 + 12\,000\cdot 0{,}5 + 14\,000\cdot 0{,}2 - 1400) + \\ + \frac{0{,}6}{0{,}7}(0{,}5\cdot 0{,}3 + 0{,}6\cdot 0{,}5 + 0{,}7\cdot 0{,}2 - 0{,}07) = -1{,}138;$$

$$K_2 = -\frac{0{,}4}{14\,000}(10\,000\cdot 0{,}2 + 12\,000\cdot 0{,}5 + 14\,000\cdot 0{,}3 - 1400) + \\ + \frac{0{,}6}{0{,}7}(0{,}6\cdot 0{,}5 + 0{,}7\cdot 0{,}5 - 0{,}15) = -1{,}332.$$

Поскольку K_2<K_1, предпочтение следует отдать первому варианту.

4.4.5. Критерий качества в условиях определенности

Рассмотренный подход к решению задачи выбора предпочтительного варианта рекомендации с позиций статистических игр имеет смысл в самом общем случае, когда отсутствует детерминированная информация о локальных показателях качества альтернатив. Такая ситуация характерна при выборе предпочтительных вариантов на начальной стадии консультирования, когда известны только априорные распределения вероятностей уровней отдельных показателей. В тех случаях, когда число вариантов рекомендаций велико, а их реализация требует значительных затрат ресурсов (материальных и времени), задачу решают изложенным методом. Однако очень часто решение принимают в условиях наличия достоверной информации об уровнях факторов, т. е. $p(F_{rij}) = 1$.

При этом соотношение (4.31) примет вид

$$K_j = \sum_{i=1}^{n} w_i (F_{ij}) F_{ij}^{\Lambda} = \sum_{i=1}^{n} K_{ij}, \tag{4.34}$$

а решением задачи выбора предпочтительного варианта рекомендации будет альтернатива, удовлетворяющая условию

$$\max_{\{j\}} (\min) K_j = \max_{\{j\}} (\min) \sum_{i=1}^{n} K_{ij}.$$

При небольших диапазонах изменения факторов w_i = const и функция (4.33) является линейной.

Сумма коэффициентов весомости

$$\sum_{i=1}^{n} w_i > 1,$$

поскольку

$$w_k = \frac{a_k F_k}{F_k} = 1.$$

Однако при определении величины коэффициентов весомости, как это будет показано ниже, иногда удобно их нормировать таким образом, чтобы выполнялось условие

$$\sum_{i=1}^{n} w_i = 1. \tag{4.35}$$

При этом

$$w_i = \frac{a_i F_{i_б}}{F_{k_б} \sum_{i=1}^{n} \frac{a_i F_{i_б}}{F_{k_б}}},$$

откуда

$$w_i = \frac{a_i F_{i_б}}{\sum_{i=1}^{n} a_i F_{i_б}}.$$

Теперь перейдем к выяснению условия объединения нескольких факторов. Пусть необходимо объединить q факторов. Тогда

$$\sum_{i=1}^{q} w_i \frac{F_i}{F_{i_б}} = \sum_{i=1}^{q} \frac{a_i F_{i_б}}{F_{k_б}} \frac{F_i}{F_{i_б}} = \sum_{i=1}^{q} \frac{a_i F_i}{F_{k_б}}.$$

Если факторы одной природы ($a_i = a$ = const), то

$$\sum_{i=1}^{q} \frac{a_i F_i}{F_{k_б}} = \frac{a \sum_{i=1}^{q} F_{i_б}}{F_{k_б}} \frac{\sum_{i=1}^{q} F_i}{\sum_{i=1}^{q} F_{i_б}} = \sum_{i=1}^{q} w_i \frac{\sum_{i=1}^{q} F_i}{\sum_{i=1}^{q} F_{i_б}}. \tag{4.36}$$

Отсюда следует вывод, что при объединении нескольких однородных факторов (например, затрат разного вида) в один фактор коэффициент весомости этого объединенного фактора равен сумме коэффициентов весомости объединяемых факторов (сочетательный закон). Выражение (4.36) соответствует принципам измерения векторов по многомерным шкалам, рассмотренным выше.

4.4.6. Экономическая форма критерия качества

Важным достоинством формы (4.34) является то, что она отражает структуру общественных потребностей, выраженную через показатели качества, в которой отдельные составляющие вектора общественных потребностей ранжированы по степени их важности.

К выводу о том, что качество продукции «само выступает как экономическая категория, отражающая меру общественной полезности», можно подойти и исходя из экономического анализа.

Поскольку всесторонняя характеристика результатов труда включает и величину затрат, то одними из факторов являются затраты. Полагая, что n-й фактор из выражения (4.34) представляет собой затраты, получим

$$K_j = \frac{w_з}{З_б} \left(\sum_{i=1}^{n-1} \frac{w_i}{w_з} З_б F_{ij} + З_j \right),$$

где $w_n = w_з$, $F_n = З_j$, $w_з$ — коэффициент весомости затрат; $З_j$— затраты по j-му варианту; $З_б$ — затраты одного из вариантов, являющиеся максимальными по сравнению с затратами других вариантов. Отсюда следует, что стоимостное выражение качества варианта определяется выражением

$$C_j = K_j \frac{З_б}{w_з} = \sum_{i=1}^{n-1} \frac{w_i}{w_з} З_б F^{д}_{ij} + З_j \tag{4.37}$$

или

$$C_j = \sum_{i=1}^{n} C_{ij}. \qquad (4.38)$$

Действительно, в выражениях (4.37) и (4.38) каждое слагаемое выражено в гривнах.

Стоимостное выражение i-го фактора j-го варианта

$$C_{ij} = \frac{w_i}{w_з} З_б F^д_{ij}.$$

С учетом выражения (4.27)

$$C_{ij} = a_i \frac{F_{iб}}{З_б} З_б F^д_{ij} = a_i F_{ij},$$

поскольку $w_з$=1. Выражение (4.39) полностью подтверждает утверждение о том, что C_{ij} является стоимостным выражением F_{ij} Экономическая эффективность реализации j-го варианта рекомендации

$$Э_j = C_j - C_з,$$

где $C_з$ — стоимостное выражение качества заменяемого варианта. На основании выражения (4.34)

$$Э_j = \sum_{i=1}^{n-1} \frac{w_i}{w_з} З_б (F^д_{ij} - F^д_{iз}) + З_j - З_з \qquad (4.40)$$

или

$$Э_j = \frac{З_б}{w_з} \Delta K_j,$$

где $\Delta K_j = K_j - K_з$.

Экономическая эффективность изменения одного i-го фактора

$$Э_{ij} = \pm \frac{w_i}{w_з} З_б (F^д_{ij} - F^д_{iз}),$$

где $F^д_{iз}$ —долевое значение фактора заменяемого варианта.

В случае, если все факторы могут быть выражены стоимостным эквивалентом, имеющим вид затрат, на основании выражения (4.40) экономическая эффективность определится известным выражением $Э_j = З_j - З_з$, которое может быть в частном случае сведено к разности приведенных затрат.

4.4.7. Сравнение форм критериев качества

Анализ существующих форм критерия качестна показывает, что все они могут быть разбиты на *три группы*: средневзвешенная (среднеарифметическая) относительных значений факторов, средневзвешенная абсолютных значении факторов, мультипликативная.

Первая группа совпадает с формой (4.34), однако в литературе нет обоснования выбора базовых значений и их связи с коэффициентами весомости, не учтена нелинейность коэффициентов весомости и их зависимость от уровней факторов.

Вторая группа форм имеет вид

$$K_j = \sum_{i=1}^{n} w_i F_{ij}. \tag{4.41}$$

Поскольку, как это будет показано ниже, при определении коэффициентов весомости не учитывают единицы измерения факторов, то ранжировка альтернатив при применении этой формы зависит от единиц измерения факторов. Размерность коэффициентов весомости в выражении (4.41) устанавливают обратной относительно размерности факторов волевым методом. Кроме того, этой форме присущи все недостатки предыдущей формы в том виде, в котором она встречается в литературе. Пример применения этой формы рассмотрен в п. 4.5.

Третья группа форм имеет вид

$$K_j = \prod_{i=1}^{n} F_{ij}^{w_i}. \tag{4.42}$$

Она противоречит аддитивности полезностей в задачах рассматриваемого класса. Применение формы (4.42), представляющей логическое объединение целей, рационально лишь в том случае, когда факторы являются булевыми функциями, принимающими только значения 0 и 1. Основной мотивировкой в пользу этой формы является выполнение условия K=0 при хоть одном из факторов, равном нулю, тогда как в форме (4.34) это условие не выполняется. Слабость такой мотивировки заключается в том, что далеко не всегда равенство нулю одного из факторов означает нулевой уровень обобщенного показателя. Примером тому может служить следующая ситуация.

Пусть в магазине покупателю, желающему приобрести стиральную машину, предложены на выбор две модели. Одна, более дорогая, оснащена программным управлением, другая не имеет такого управления. Во многих случаях, как показывает опыт, покупатель может предпочесть вторую модель, поскольку при этом появляется возможность сэкономить денежные ресурсы.

В некоторых случаях создается ложное впечатление равенства нулю полезности при нулевом уровне одного из показателей. Пусть необходимо выбрать средство для перемещения грузов по водным маршрутам, а множество альтернатив состоит из автомобилей обычного использования, позволяющих осуществлять движение только по сухопутным маршрутам. Даже удовлетворение условиям грузоподъемности, стоимости и т. д. не приведет к решению задачи — выбору плавсредств, поскольку все элементы множества альтернатив не удовлетворяют поставленной цели — движению по воде. Это случай кажущейся бесполезности, обусловленный нарушением важнейшего требования, заключающегося в том, что набор альтернатив должен прежде всего удовлетворять поставленной цели, являющейся одним из элементов реализации сформированных рекомендаций.

Все варианты рекомендаций с недопустимо низким уровнем показателей должны исключаться из рассмотрения и сравнения.

Следует отметить, что в ряде случаев ранжированию по увеличению предпочтения соответствует уменьшение величин факторов и обобщенного показателя.

Таким образом, *наибольшей обоснованностью обладает критерий вида (4.31) и вытекающий из него в условиях определенности критерий вида (4.34). Это позволяет рекомендовать указанные критерии к использованию в практике формирования рекомендаций по решению консультационных задач консультируемой проблемы.*

4.5. Методы ранжирования факторов по их важности

4.5.1. Аналитические методы

Решение задачи выбора предпочтительного варианта рекомендации из множества альтернатив возможно лишь при известных значениях коэффициентов весомости факторов, входящих в комплексный критерий. Существует несколько методов решения этой задачи.

Так, критерием выбора значений весовых коэффициентов служит минимум дисперсии оценочного функционала комплексного критерия вида (4.41).

Очевидно, что этому условию соответствуют наименьшие коэффициенты весомости факторов, имеющих наибольшую дисперсию. При этом скорее оценивается дисперсия, чем важность факторов, что, естественно, не может быть положено в основу решения задачи. Коэффициенты весомости можно определять с помощью соотношения

$$w_i = \frac{\lambda}{F_{i_{\text{ср}}} - F_{i_{\text{пред}}}},\qquad (4.43)$$

где $F_{i_{ср}}$ — средние значения уровней факторов, удовлетворяющие требованиям нормативно-технической документации;

$F_{i_{пред}}$ — предельно допустимое значение i-го фактора; λ— постоянный множитель.

Это выражение основано на допущении одинаковой полезности вариантов, имеющих один из факторов на предельном уровне. Совершенно очевидно, что такое допущение в общем случае не выполняется, поскольку полезность варианта зависит от важности фактора, имеющего предельно допустимый уровень.

Результатом необоснованного вывода является зависимость значения коэффициента весомости, определяемого по выражению (4.43), от единиц измерения. Главный недостаток выражения (4.43) в том, что оно не отражает смысла коэффициента весомости, представляющего собой меру относительной важности факторов. Выражение же (4.43) утверждает возможность определения величины коэффициента весомости безотносительно других факторов.

Аналитическое определение коэффициента весомости возможно на основании (4.33). Пусть все факторы имеют стоимостное

выражение, тогда $a_i F_{i_б} = C_{i_б}$, откуда $w_i = C_{i_б} / C_{k_б}$, где C_i — стоимостное выражение i-го фактора.

Как уже было указано, большинство факторов имеют стоимостное выражение. Например, коэффициент полезного действия может быть выражен через величину потерь энергии, стоимость которой может быть легко определена. *Описанный подход к решению задачи может быть рекомендован при невозможности аналитического определения коэффициентов весомости всех факторов*. В противном случае удобнее использовать экономическую форму критерия качества, обходя определения коэффициентов весомости в явном виде.

Основной информацией для аналитического определения коэффициентов весомости в консультировании являются корреляционные зависимости коэффициента a_i для разных уровней факторов различных видов консультируемых проблем, на базе которых возможно построение соответствующих нормативов по формированию рекомендаций. Нормативы могут быть *отраслевыми, региональными и ведомственными.*

В тех случаях, когда невозможно аналитическое определение коэффициентов весомости, следует использовать эвристические методы, одним из которых является метод экспертных оценок.

4.5.2. Определение коэффициентов важности факторов однородной группой экспертов

Отсутствие информации о важности факторов приводит к необходимости решения задачи в условиях неопределенности и риска.

Пусть $w_{ik} \subset W_i$ (здесь W_i— множество возможных значений коэффициентов весомости i-го фактора), а p_{ik} — вероятность того, что коэффициент весомости i-го фактора примет k-е из m значений w_{ik}. Тогда

$$w_i = M\{w_{ik}\} = \sum_{k=1}^{m} w_{ik} p_{ik}. \qquad (4.44)$$

При полной неопределенности все значения равновероятны. Добиться уменьшения неопределенности можно за счет получения дополнительной информации. Одним из возможных источников такой информации является коллективный опыт специалистов. Процедура получения информации основана на формировании группы экспертов с последующим их опросом по определенной методике. В результате экспертизы распределение вероятностей изменится. При формировании группы из m экспертов равной квалификации коэффициент весомости i-го фактора на основании соотношения (4.44) будет

$$w_i = \sum_{k=1}^{l} w_{ik} \frac{m_{ik}}{m},$$

где m_{ik} — число экспертов, оценивших важность i-го фактора коэффициентом w_{ik} (k-м значением w_i); l — число подмножества значений коэффициентов весомости l-го фактора, состоящих из одинаковых значений коэффициентов весомости w_{ik}. Запишем это выражение иначе:

$$w_i = \frac{1}{m} \sum_{k=1}^{l} w_{ik} m_{ik},$$

но поскольку

$$m = \sum_{k=1}^{l} m_{ik},$$

то

$$w_i = \frac{1}{m}\sum_{k=1}^{m} w_{ik},$$

а при нормировании в соответствии с условием (4.35)

$$w_i = \frac{\frac{1}{m}\sum\limits_{k=1}^{m} w_{ik}}{\frac{1}{m}\sum\limits_{m=1}^{n}\sum\limits_{k=1}^{m} w_{ik}} = \frac{\sum\limits_{k=1}^{m} w_{ik}}{\sum\limits_{k=1}^{m}\sum\limits_{i=1}^{n} w_{ik}}. \qquad (4.45)$$

Полагая группу экспертов репрезентативной выборкой, результаты экспертизы можно распространить на любое число экспертов.

Учитывая результаты, полученные в первом разделе, не следует стремиться к полному исключению неопределенности, поскольку приращение эффективности накопления дополнительной информации снижается по мере ее роста.

Существует *шесть методов экспертных оценок*: ранжирование, оценка в баллах, частичное парное сравнение I, частичное парное сравнение II, полное парное сравнение, последовательное сравнение.

Наиболее простыми методами, получившими широкое распространение, являются первые два метода — *ранжирование и оценка в баллах.* Серьезным достоинством этих методов является невысокая трудоемкость. Это тем более важно, что в качестве экспертов привлекают высококвалифицированных специалистов, загруженных своей основной работой. Попытка навязать им дополнительную нагрузку, требующую значительной затраты времени и труда, обычно оканчивается неудачей.

Метод *ранжирования* заключается в расположении всех факторов в порядке возрастания либо убывания важности с приписыванием соответствующего порядкового номера — ранга. Обычно ранжируют в порядке убывания важности. Наиболее важному показателю присваивают рант *n,* а наименее важному— ранг 1. Ранг, присвоенный фактору, называют прямым рангом.

Метод *балльной оценки* заключается в оценке важности факторов по шкале баллов. При этом наиболее важный показатель, оценивают наивысшим баллом, а остальные в порядке убывания — меньшим числом баллов. Равноценные факторы оценивают одним рангом либо одинаковым числом баллов.

Метод балльной оценки основан на использовании шкалы интервалов с соглашением о нуле либо шкалы отношений.

Метод ранжирования основан на использовании шкалы порядка. Аксиомы упорядочения и тождества, положенные в ее основу, допускают статистические операции получения частот и мод. Кроме того, ранговый порядок позволяет вычислять медианы, центили и коэффициенты ранговой корреляции. Арифметические действия недопустимы.

Использование экспертного метода включает следующие этапы:

1) формулирование задачи; 2) организация экспертизы; 3) обработка результатов; 4) оценка эффективности.

Формулирование задачи. Прежде чем начинать экспертизу, необходимо четко сформулировать вопрос, на который должен быть получен ответ. Это возможно только в том случае, если известна аналитическая форма функций цели.

Выше было показано, что наиболее обоснованной формой функции цели является выражение вида (4.34). В качестве k-го фактора наиболее удобно выбирать затраты, тогда a_i будет стоимостным выражением единицы i-го фактора.

Анализ показывает, что a_i является нелинейной функцией уровня фактора F_i. Например, уменьшение габаритных размеров электродвигателя привода станка при прочих равных условиях приводит к уменьшению затрат на конструктивные элементы,, связанные с его установкой. Однако при значительном уменьшении габаритных размеров удельное снижение затрат уменьшается и даже может стать равным нулю, так как в качестве несущих элементов могут быть использованы уже имеющиеся детали конструкции в их основном виде. Например, экономический эффект от увеличения скорости самолета на каждые 100 км/ч различен в диапазоне скоростей, близких к 200 и 2000 км/ч. Аналогичных примеров можно приводить бесконечное множество. Таким образом, в общем случае величина коэффициентов важности w_i зависит как от уровня факторов [a_i (F_i)], так и от выбора базовых значений.

Здесь уместно отметить чрезвычайно важное свойство коэффициента важности, заключающееся в независимости его величины от единиц измерения факторов. Изменение единиц измерения факторов приводит к изменению численных значений базовых величин и одновременно коэффициента соответствия a_i таким образом, что величина коэффициента важности не изменяется.

Итак, задача экспертизы заключается в получении ответа на вопрос о величине коэффициента соответствия или, при выборе затрат в качестве k-го фактора, в получении ответа на вопрос: «Сколько «стоит» единица i-го фактора при j-м уровне» (первая информационная ситуация).

Обычно эксперты не обладают столь высокой степенью информативности, особенно в тех случаях, когда речь идет о факторах, стоимостная оценка которых затруднительна. С другой стороны, как следует из выражения (4.42),

$$w_i = \frac{u_{i_б}}{u_{k_б}}, \tag{4.46}$$

где $u_{i_б}, u_{k_б}$ — полезность (выигрыш) i-го и k-го факторов при базовых значениях соответственно. Таким образом, w_i представляет собой относительную полезность i-го фактора при определенном его уровне. *Подчеркиваем, что речь идет об относительной полезности значения фактора, а не о полезности его единицы.*

Отсюда вытекает вторая разновидность формулировки вопроса к экспертам: «Какова относительная важность факторов при их заданных уровнях?» (вторая информационная ситуация). Как в первом, так и во втором случае предполагается набор ответов, по количеству равный набору альтернатив.

В тех случаях, когда уровни факторов, характеризующие различные альтернативы, отличаются незначительно, эксперты не могут оценить изменения их относительной важности. Таким образом, возникает третья информационная ситуация, в которой перед экспертами ставится вопрос: «Какова относительная важность факторов?». При этом для ориентировки указывают один набор уровней, выбираемый в качестве базовых значений.

В особых случаях возникает четвертая информационная ситуация, в которой эксперты могут только ранжировать факторы по степени их важности.

Помимо рассмотренной формы функции цели (4.34), как показано выше, предлагается ряд других форм. Рассмотрим форму (4.41), отличающуюся от формы (4.34) тем, что коэффициенты важности являются постоянной величиной, не зависящей от уровня факторов, имеющей размерность, обратную размерности факторов, и форму (4.42).

Отсутствие строгого вывода этих форм в отличие от формы (4.34) не позволяет сформулировать физический смысл коэффициентов w_i, называемых в литературе коэффициентами важности, и выяснить, от чего зависит их величина. Проделаем это для формы (4.41). Если в форме (4.34) положить $F_{i_б} = F_{k_б} = 1$, то получим выражение (4.41).

Таким образом, форма (4.41) является частным случаем формы (4.34) при единичных базовых значениях факторов. Это приводит к тому, что значение w_i не зависит от выбора базовых значений, но очень сильно зависит от выбора единиц измерения факторов. Последнее в значительной мере усложняет работу экспертов, поскольку практически затруднение учета уровня базового значения приводит к меньшей погрешности результатов экспертизы, чем затруднение учета единиц измерения. Действительно, если в качестве базового значения фактора выбрать среднее из возможных, то при диапазоне изменения

$$D = \frac{F_{i_{\max}}}{F_{i_{\min}}}$$

максимальная ошибка

$$\Delta w\,\% = \frac{F_{i_{\max}} - \dfrac{(F_{i_{\max}} - F_{i_{\min}})}{2}}{\dfrac{(F_{i_{\max}} + F_{i_{\min}})}{2}}\,100\,\% = \frac{D-1}{D+1}\,100\,\%.$$

Обычно диапазон изменения факторов не превышает трех, а в третьей информационной ситуации и того меньше. При этом Δw %=50%. Максимальное значение ошибки при диапазоне изменения фактора, стремящемся к бесконечности, стремится к 100%.

Минимальное значение ошибки вследствие изменения единиц измерения в метрической системе мер равно 100%, а максимальное значение принципиально не ограничено. Отсюда следует вывод о значительном преимуществе формы (4.34) но сравнению с формой (4.41).

Подойдем к выводу формы (4.41) с другой стороны. При выполнении аксиомы аддитивности полезностей факторов можно записать

$$u = \sum_{i=1}^{n} u_i.$$

Полезность i-го фактора $u_i = \Delta u_i\, F_i$, где Δu_i — средняя полезность единицы i-го фактора, в общем случае зависящая от уровня фактора. Отсюда следует выражение (4.41) с учетом равенства

$$w_i = \frac{\Delta u_i}{\Delta u_k} = \frac{a_i \Delta F_i}{\Delta F_k}. \tag{4.47}$$

Таким образом, w_i представляет собой относительную полезность единицы i-го фактора по отношению к единице полезности k-го фактора и зависит от единиц измерения.

Подобная интерпретация коэффициентов весомости не облегчает задачу экспертов даже для линейной системы. Здесь уместно отметить, что определить коэффициенты весомости для формы (4.41) аналитически можно при предварительном использовании выражения (4.46), а не непосредственно на базе выражения (4.47).

Организация экспертизы. Вопросы организации экспертизы рассмотрены во многих работах. Поэтому остановимся только на некоторых аспектах проблемы.

Прежде всего необходима высокая точность постановки вопросов и определение содержания применяемых понятий даже тех, которые являются широко распространенными. Как показывает опыт, очень часто организаторы экспертизы сталкиваются с разночтением даже таких понятий. Это требует оперативного уточнения и разъяснения. Отсюда следует вывод о низкой эффективности заочной экспертизы. Последнее связано еще и с невозможностью применения метода Дельфы.

Одним из важнейших вопросов является количественный состав списка факторов. Обычно коэффициенты важности многих факторов поддаются аналитическому определению, и на долю экспертизы остается небольшое их число. Однако в тех случаях, когда число факторов, входящих в анкету, велико, следует учитывать психофизиологические возможности экспертов. К сожалению, до настоящего времени не существует единого мнения о предельном и оптимальном их числе. По существующим рекомендациям оно колеблется от 6—8 до 20—25.

Для обхода этого ограничения следует полное множество факторов разбивать на пересекающиеся подмножества и проводить экспертизу так, как это осуществляется неоднородной группой экспертов. В зависимости от информационной ситуации по приведенной выше классификации оценивают либо величину a_i в соответствующих единицах, либо w_i в баллах или ранжируют. Обязательным условием является указание уровней факторов, для которых проводят экспертизу. При этом эксперты указывают различные значения a_i или w_i для каждого уровня.

В третьей информационной ситуации в качестве базовых значений факторов (по ним ведут экспертизу) следует выбирать значения факторов современных образцов либо близких к ним, опыт работы с которыми есть у экспертов. При этом необходимо ориентироваться на те из них, уровень факторов которых соответствует среднему значению уровней сравниваемых вариантов. Это обеспечивает повышение точности экспертизы. В третьей информационной ситуации эксперты не могут указать зависимость коэффициентов важности от уровней, и в этом случае информация об уровнях факторов является ориентирующей.

Опыт работы с экспертами показывает, что для получения надежных результатов экспертизы необходимо принять ряд мер. Прежде всего следует иметь в виду, что экспертиза в какой-то мере отражает прошлый опыт. Необходимо обратить внимание экспертов на этот психологический фактор. Очень полезно сообщить экспертам результаты прогноза совершенствования сформированных рекомендаций, для которых предназначены формируемые рекомендации по решению задач консультируемой проблемы, а также результаты прогноза реализации новых сформированных рекомендаций. Указанную информацию лучше всего сообщить не сразу, а после первого тура оценки важности на базе метода Дельфы. В этом случае дополнительная информация позволит экспертам внести соответствующие коррективы в свои оценки и не окажет чрезмерного давления на них. Для исключения взаимного влияния экспертов обмен информацией должен быть анонимным.

Обработка результатов. В условиях первой информационной ситуации, когда эксперты отвечают на вопрос о «стоимости» единицы *i*-фактора, для каждого уровня определяют среднее значение величины *i*-го фактора для *j*-го уровня.

В условиях второй информационной ситуации, когда эксперты измеряют важность в баллах по шкале интервалов с условным нулем, либо по шкале отношений, они отвечают на вопрос: во сколько раз полезность одного фактора выше полезности другого фактора. При этом выполняется свойство аддитивности и допустимы все арифметические и статистические операции Отсюда следует, что коэффициент важности *i*-го фактора *j* -го уровня

$$w_{ij}=\frac{\sum_{r=1}^{m} B_{ijr}}{\sum_{r=1}^{m} B_{kjr}},$$

где B_{ijr}, B_{kjr} — балльная оценка *i*-го и *k*-го факторов *j*-го уровня *r*-м экспертом. При этом функция цели (4.34) принимает вид

$$K_j=\sum_{i=1}^{n} w_{ij} \neq 1.$$

В условиях третьей информационной ситуации, когда эксперты измеряют важность в баллах по шкале отношений независимо от уровня факторов, величины коэффициентов важности определяются выражением

$$w_i = \frac{\sum_{r=1}^{m} B_{ir}}{\sum_{i=1}^{n} \sum_{r=1}^{m} B_{ir}}, \qquad (4.48)$$

где m — число экспертов; B_{ir} — балл, присвоенный r-м экспертом i-му фактору. В этом случае выполняется известное условие

$$\sum_{i=1}^{n} w_i = 1.$$

В табл. 4.11, в качестве примера, приведены результаты экспертизы, проведенной применительно к серии автоматизированного электропривода малой мощности для станкостроения в условиях третьей информационной ситуации. В экспертизе принимали участие крупные специалисты в области станочного электропривода — ведущие сотрудники одиннадцати научно-исследовательских и проектных организаций.

Таблица 4.11

Значение коэффициентов важности показателей, характеризующих автоматизированный станочный электропривод малой мощности (0,7—10 кВт)

Показатель	Вариант привода	
	реверсивный	нереверсивный
Быстродействие	0,177	0,125
Надежность	0,187	0,196
Габариты привода	0,132	0,156
Габариты двигателя	0,151	0,161
Стоимость привода	0,089	0,106
Потери электроэнергии	0,072	0,070
Затраты на обслуживание	0,126	0,140
Реактивная мощность и мощность искажения	0,064	0,046

Почти все участники выступали в качестве очных экспертов. Организации были подобраны таким образом, чтобы свое мнение могли высказать три группы специалистов: научные работники, разработчики, изготовители, выступающие в ряде случаев и как потребители. В экспертизе принимали участие как электрики, так и механики.

Наименее информативной является четвертая ситуация, в которой важность факторов оценивают по шкале порядка. Аксиомы упорядочения и тождества, положенные в основу этой шкалы, допускают только некоторые статистические операции и исключают арифметические действия. Ранг фактора в этом случае можно определить, если представить оценки экспертов в виде интервального ряда с интервалами, равными единице. Ранг каждого фактора определяют как медиану интервального ряда:

$$R_i = Mc\{R_{ik}\} = R_{i0} + c\,\frac{\sum_p/2 - \sum_{m-1}}{p_m},$$

где R_{i0} — нижняя граница медианного интервала; c — величина класса; $\sum_p$ — сумма частот (число членов заряда); $\sum_{m-1}$- сумма частот интервалов, предшествующих медианному; p_m — частота медианного класса.

Поскольку для рассматриваемого интервального ряда величина класса равна единице, а сумма частот равна числу экспертов, то ранг i-го фактора

$$R_i = R_{i0} + \frac{m/2 - \sum_{m-1}}{p_m}. \tag{4.49}$$

Полученные значения рангов позволяют ранжировать факторы и частные цели для решения задачи выбора по локальным критериям, например, на базе метода последовательных уступок.

Использование же ранжирования для определения коэффициентов весомости факторов принципиально недопустимо, поскольку оно требует проведения недозволенных арифметических операций. При этом не следует принимать во внимание утверждение о том, что метод ранжирования и метод балльной оценки по шкале интервалов дают приблизительно одинаковые результаты.

Изложенное выше легко проиллюстрировать на примере. Пусть комплексный показатель качества включает два фактора. Предположим, все эксперты присвоили одному показателю первый ранг, а второму — второй. Тогда на основании выражения (4.48)

$$w_1 = 0{,}667;\ w_2 = 0{,}333.$$

Пусть показатель качества включает три фактора. Тогда при полной согласованности мнений всех экспертов получим w_1= 0,5; w_2=0,333; w_3=0,167. При четырех факторах и полной согласованности мнений экспертов получим w_1 = 0,4; w_2=0.3; w_3=0,2; w_4 = 0,1.

Предположим, что два первых важных фактора во всех случаях одни и те же. Второй вариант отличается от первого учетом одного дополнительного второстепенного фактора. Третий вариант отличается от второго дополнительным учетом еще одного второстепенного фактора. Таким образом, учет второстепенных факторов изменяет величину соотношения коэффициентов весомости двух главных факторов. Если в первом случае $w_1/w_2=2$, то во втором случае 1,5, а в третьем 1,33, т. е. если в первом случае первый фактор в 2 раза важнее второго, то во втором только в 1,5, а в третьем в 1,33 раза, несмотря на то, что речь идет об одних и тех же факторах.

В общем случае при полной согласованности мнений экспертов на основании выражения (4.48)

$$w_i = \frac{2R_i}{n(1+n)}. \tag{4.50}$$

Поскольку принято условие согласованности мнений всех экспертов, можно записать, что $r_{ik}=i$. Тогда на основании формулы (4.50) коэффициент весомости i-го фактора определится соотношением

$$w_i = \frac{2(n-i+1)}{n(1+n)}. \tag{4.51}$$

При этом

$$\frac{w_e}{w_i} = \frac{n-e+1}{n-i+1},$$

т. е. отношение коэффициентов весомости e-го и i-го факторов зависит от числа факторов n, что является следствием применения шкалы порядка. В частном случае

$$\frac{w_1}{w_i} = \frac{n}{n-i+1}.$$

При увеличении числа n отношение $\frac{w_1}{w_i}$ (i=const) асимптотически стремится к единице.

Изложенные рассуждения приведены в предположении отсутствия равнозначных факторов, т. е. факторов, получивших одинаковый ранг, и при условии полной согласованности мнений экспертов. При нарушении этих условий знаменатель в выражении (4.51) для важных факторов несколько уменьшается, а для маловажных возрастает, что приводит к некоторому увеличению дроби в первом случае и уменьшению во втором. Эти изменения определяются степенью несогласованности мнений экспертов.

Пример. Определить ранг трех факторов, если четыре эксперта произвели их ранжировку в порядке убывания предпочтения (табл. 4.12).

Таблица 4.12

Результаты экспертизы

Ранг					№ фактора
5	4	3	2	1	—
3	1	—	—	—	1
—	1	2	1	—	2
—	—	1	2	1	3
—	—	—	1	3	4
1	2	1	—	—	5

Для определения медианы рангов первого фактора составляем табл. 4.13.

Таблица 4.13

Вычисление медианы рангов первого фактора

Ранг	Частота класса	Накопленная частота
5	3	3
4	1	4
	$\Sigma_p = 4 = m$	

Срединная частота, равная двум (сумма частот, равная числу экспертов, равна 4), лежит в медианном интервале. Следовательно, медианным интервалом является первый интервал. Накопленная частота определяется путем сложения частот каждого класса. Тогда на основании выражения (4.49) получим

$$R_1 = 5 + \frac{\frac{4}{2} - 0}{3} = 5{,}66.$$

Аналогичным образом строим таблицы для определения медианы рангов всех факторов (табл. 4.14—4.17).

Таблица 4.14

Вычисление медианы рангов второго фактора

Ранг	Частота класса	Накопленная частота
4	1	1
3	2	3
2	1	4

Таблица 4.15

Вычисление медианы рангов третьего фактора

Ранг	Частота класса	Накопленная частота
3	1	1
2	2	3
1	1	4

Таблица 4.16

Вычисление медианы рангов четвертого фактора

Ранг	Частота класса	Накопленная частота
2	1	1
1	3	4

Таблица 4.17

Вычисление медианы рангов пятого фактора

Ранг	Частота класса	Накопленная частота
5	1	1
4	2	3
3	1	4

На основании выражения (4.49)

$$R_2 = 3 + \frac{2-1}{2} = 3,5; \quad R_3 = 2 + \frac{2-1}{2} = 2,5;$$

$$R_4 = 1 + \frac{2-1}{4} = 1,25; \quad R_5 = 4 + \frac{2-1}{2} = 4,5.$$

Отсюда ранжировка факторов и порядке убывания предпочтения следующая: 1, 5, 2, 3, 4.

Существование различных информационных ситуаций обусловлено многообразием задач, решаемых в процессе консультирования. Поэтому консультант в своей деятельности может столкнуться с любой из этих ситуаций.

Оценка эффективности экспертизы. Если до проведения экспертизы вероятность любой оценки

$$p_{i0} = \frac{1}{A}, \tag{4.52}$$

где A — число возможных значений оценки важности, данной экспертом, то после ее проведения вероятность равна p_{ik}. Это обусловлено тем, что экспертиза, увеличивая информативность, не делает систему детерминированной, а только уменьшает степень неопределенности на величину

$$\Delta H = \Delta I = \sum_{i=1}^{n} \log \frac{p_{ik}}{p_{i0}} = \sum_{i=1}^{n} \log (p_{ik} A).$$

Полагая результаты экспертизы по всем факторам одинаковыми, получаем величину дополнительной информации

$$\Delta I = n \log (A p_{ik}).$$

Максимально возможный объем полученной информации при полностью согласованном мнении экспертов

$$\Delta I_{\max} = n \log A.$$

На основании приведенных рассуждений создается впечатление, что необходимо увеличение A. Это справедливо до достижения предела «чувствительности» экспертов. Так, например, если при стобалльной шкале (A = 100**)** эксперты оценили важность факторов с разрешающей способностью, равной десяти (все оценки в баллах кратны десяти), то переход к тысячебалльной шкале (A=1000) скорее всего не приведет к увеличению информативности экспертизы. Если же разрешающая способность выше, то увеличение A может привести к увеличению информативности экспертизы только в том случае, если эксперты могут повышать разрешающую способность измерения важности факторов. Напоминаем, что *А* целое число, как это следует из его определения. Кроме того, эксперты обычно десятыми долями не пользуются.

Информативность групповой экспертизы определяет ее эффективность и зависит от численности и качественного состава экспертной группы. Максимальная эффективность экспертизы определяется выражением

$$Э_{э_{\max}} = Э_{э_{и}} \left[1 - B_0 \exp \left(\frac{\log A}{\log \frac{1}{A}} \right) \right],$$

где $Э_э$ — идеальное значение эффективности экспертизы, соответствующее полному детерминизму; B_o — начальная неупорядоченность системы. Отсюда следует, что

$$Э_{э_{\max}} = Э_{э_{и}} \left[1 - \frac{B_0}{e} \right].$$

Поскольку максимальная эффективность практически недостижима, очевидно, можно удовлетвориться некоторым приближенным значением $\beta Э_{э_{\max}}$, где β <1. Тогда на основании выражения (4.52) получим

$$\beta \left[1 - \frac{B_0}{e} \right] = 1 - B_0 \exp \left(\frac{\log p_{ik} A}{\log 1/A} \right).$$

После ряда преобразований получим

$$p_{ik} \geqslant A^{\left\{ \ln \left[\frac{\beta}{e} + \frac{1}{B_0}(1-\beta) + 1 \right] \right\}}. \qquad (4.53)$$

Выражение (4.53) позволяет производить проверку численного состава группы экспертов и обеспечения требуемой вероятности. Например, если p_{i0}=0,l (A =10); β = 0,8, то p_{ik}=0.475.

Для определения величины p_{ik}, полученной в результате экспертизы, необходимо построить полигон распределения мнений экспертов, а затем найти закон распределения вероятностей различных оценок экспертов. Гипотезу проверяют одним из известных критериев.

Проверка выполнения условия (4.53) позволяет косвенно оценивать требуемую численность группы при данной согласованности мнений.

4.5.3. Согласованность мнений экспертов

Согласованность мнений экспертов имеет большое значение при оценке степени важности различных единичных факторов.

Существует несколько методов оценки меры согласованности. Наиболее грубый метод основан на расчете так называемых коэффициентов ассоциации. В зависимости от степени согласованности коэффициент ассоциации изменяется от нуля до единицы. Кроме этого метода, существует ряд разновидностей метода ранговой корреляции (например, по Кендэллу, по Спирмену). Коэффициент согласованности мнений m экспертов по q-му признаку в этом случае определяется выражением

$$\omega_q = \frac{S(d^2)}{\frac{1}{12} m^2 (n^3 - n) + m \sum_i T_j},$$

где

$$s(d^2) = \sum_{i=1}^{n} \left\{ \sum_{j=1}^{m} (x_{qi}^{j}) - \frac{1}{2} m (n+1) \right\}^2;$$

$$T_j = \frac{1}{12} \sum_j (t_j^3 - t_j);$$

здесь t_j — число повторений каждого ранга в ранжировке j-м экспертом; x^{j}_{qi} — ранг i-го альтернативной рекомендации по q-*му* признаку, приписанный j -м экспертом.

Согласованность мнений экспертов может быть определена также методом конкордации, методом однородности множества, коэффициентом согласия

$$\omega = 1 - \frac{H}{H_{\max}},$$

где H, $H_{\max}$ — неопределенность, полученная после экспертизы, и максимальная неопределенность соответственно.

Во всех случаях согласованность определяется некоторым числом, в лучшем случае позволяющим оценивать процент полностью согласованных мнений. Такой подход к решению задачи обладает существенным недостатком — отсутствием критерия удовлетворительной согласованности, т. е. удовлетворительных значений коэффициентов ассоциации, рантовой корреляции, конкордации и т. д.

Для исключения этого недостатка рассмотрим степень влияния отклонения коэффициента весомости одного фактора, вносимого несогласованностью мнений экспертов, на величину обобщенного показателя качества (4.34).

Пусть в результате экспертизы m — 1 экспертов дали значения коэффициентов весомости w_1, w_2,..., w_n, а m экспертов, включая всю предыдущую группу из m — 1 экспертов, — w'_1, $w'_{2, \ldots,}$ w'_n.

Если при этом обобщенный показатель качества j-го альтернативного варианта, рассчитанный на основании результатов экспертизы первой группы, состоящей из m — 1 экспертов, определяется выражением (4.34), то обобщенный показатель качества того же альтернативного варианта, полученный на основании результатов экспертизы второй группы, состоящей из m экспертов, определяется выражением

$$K'_j=\sum_{i=1}^{n} w_i F^{д}_{ij}+\sum_{i=1}^{n}\Delta w_i F^{д}_{ij},$$

где

$$\Delta w_i = w_i - w'_i \,(\Delta w'_i \gtrless 0).$$

Отсюда следует, что изменение величины показателя качества j-го альтернативного варианта

$$\Delta K_j = K'_j - K_j = \sum_{i=1}^{n}\Delta w_i F^{д}_{ij}. \tag{4.54}$$

Если величина ΔK_j меньше разности показателей качества любых двух альтернативных вариантов, то полученная последовательность ряда альтернативных вариантов, ранжированная в порядке возрастания (убывания) обобщенного показателя качества, не нарушается и, следовательно, колебания значений w_i не влияют на принятие решения — выбор предпочтительного варианта. Из соотношения (4.54) следует, что ΔK_j зависит от колебания коэффициентов важности, а также от числа факторов и их величины.

Пусть отклонение в баллах одного i-го фактора, вызванное одним m-м экспертом,

$$\Delta B_{im} = B_{i(m-1)} - B_{im},$$

где $B_{i(m-1)}$ — средневзвешенный балл i-го фактора, полученный при опросе m—1 экспертов; B_{im}—средневзвешенный балл i-го фактора, полученный при опросе m экспертов.

Тогда

$$\Delta w_i = \frac{\sum_{k=1}^{m} B_{ik} + \Delta B_{im}}{\sum_{i=1}^{n}\sum_{k=1}^{m} B_{ik} + \sum_{i=1}^{n} \Delta B_{ik}} - \frac{\sum_{k=1}^{m} B_{ik}}{\sum_{i=1}^{n}\sum_{k=1}^{m} B_{ik}}.$$

Учитывая, что

$$\sum_{k=1}^{m} B_{ik} \gg \Delta B_{im};\ \sum_{i=1}^{n}\sum_{k=1}^{m} B_{ik} \gg \sum_{i=1}^{n} \Delta B_{im},$$

можно считать, что несогласованность мнений экспертов значительно меньше влияет на коэффициент весомости, чем на оценку в баллах.

При оценке величины

$$\sum_{i=1}^{n} \Delta B_{im}$$

следует иметь в виду, что обычно имеют место отклонения обоих знаков. Для оценки согласованности мнений и влияния каждого эксперта на общую согласованность мнений группы последовательно исключают из рассмотрения одного очередного эксперта и ранжируют альтернативные варианты в порядке возрастания (убывания) показателя качества. Если ранжировка не изменяется, мнения следует считать согласованными. Число экспертов следует выбирать, таким, чтобы несогласованность их мнений начинала оказывать влияние на ранжировку альтернативных вариантов. При этом создается возможность учета оригинальных мнений экспертов. Если, используя метод Дельфы, не удается согласовать мнения экспертов так, чтобы последовательность ряда альтернативных вариантов не изменялась, то следует увеличить численность экспертов до числа, не оказывающего влияние на эту последовательность вследствие несогласованности мнений.

Предлагаемая методика оценки согласованности мнений экспертов в отличие от существующих базируется на четком критерии — неизменности последовательности ранжирования альтернативных вариантов. Это справедливо, тем более, что для лица, формирующего рекомендации, важна не столько величина комплексного показателя качества, сколько последовательность ряда альтернативных вариантов, ранжирования в порядке их предпочтения.

Изложенные соображения показывают, что некоторая остаточная неопределенность, обусловленная недостаточным объемом информации, получаемой при применении метода экспертных оценок, не снижает точность решения задачи, если уровень неопределенности не превышает значения, соответствующего разности значений обобщенного показателя качества альтернативных вариантов.

4.5.4. Определение коэффициентов важности факторов неоднородной группой экспертов

Обычно не удается создать однородную группу экспертов, обладающую компетентностью, достаточной для оценки важности всех факторов. Поэтому приходится формировать неоднородную группу, включающую несколько подгрупп. В каждую подгруппу входят специалисты одной специальности (электрики, механики, гидравлики, специалисты по пневмооборудованию и т. д.) либо одной профессии (разработчики, представители заводов-изготовителей, представители потребления).

Такая неоднородная группа способна более объективно и всесторонне оценивать важность факторов, обеспечивать более полную информацию, чем однородная группа.

Таким образом, группа экспертов в общем случае представляет собой некоторое множество экспертов *M,* состоящее из непересекающихся подмножеств $M_1, M_2, \ldots, M_i, \ldots, M_k$ $(M_1 \subset M)$.

Одновременно множество факторов *F* разбивают на несколько пересекающихся подмножеств $\mathcal{F}_1$ так, что выполняются условия

$$\mathcal{F}_1 \subset F,\ \mathcal{F}_1 \cap \mathcal{F}_2 \cap \ldots \cap \mathcal{F}_1 \cap \ldots \cap \mathcal{F}_d \neq \varnothing.$$

Коэффициенты весомости факторов F_i, входящих во все подмножества

$$\mathcal{F}_l (F_i \subset \mathcal{F}_l, l = \overline{1, d}),$$

определяют соотношением

$$w_i^* = \frac{1}{d} \sum_{l=1}^{d} w_{il}, \tag{4.55}$$

где l — номер подмножества факторов; w_{il} — коэффициент важности i-го фактора в l- м подмножестве факторов, экспертиза по которому осуществляется l-м подмножеством экспертов; d — число подмножеств.

Для определения коэффициента важности w^*_k фактора, не входящего во все подмножества, введем предположение о сохранности соотношений

$$\frac{w_i^*}{w_k^*} = \frac{w_{il}}{w_{kl}} \qquad (4.56)$$

для всех r факторов, общих для всех подмножеств F_1 и всех q подмножеств, содержащих k-й фактор, не являющийся общим для всех подмножеств. Здесь w_{k1} — коэффициент важности k-го фактора в l-м подмножестве, экспертиза по которому осуществляется l-м подмножеством экспертов.

Вероятность выполнения условия (4.56) для всех указанных значений i и l одновременно меньше единицы. Полагая, что вероятность выполнения этого соотношения для каждого из указанных i и l равна p_{il}, получим

$$w_k^* = \sum_l \sum_i p_{il} \frac{w_{kl}}{w_{il}} w_i^*. \qquad (4.57)$$

Если k-й фактор входит только в одно подмножество, то выражение (4.57) примет вид

$$w_k^* = \sum_i p_i \frac{w_k}{w_i} w_i^*$$

или

$$w_k^* = w_k \sum_i p_i \frac{w_i^*}{w_i}. \qquad (4.58)$$

Полагая выполнение условия (4.56) равновероятным для r факторов в каждом подмножестве (i=1, 2, …, r), входящих во все подмножества, и q подмножеств факторов (l=1, 2, …, q), содержащих данный фактор, из соотношений (4.57) и (4.58) получим соответственно

$$w_k^* = \frac{1}{rq} \sum_{l=1}^{q} w_{lk} \sum_{i=1}^{r} \frac{w_i^*}{w_{li}};$$

$$w_k^* = \frac{w_k}{r} \sum_{i=1}^{r} \frac{w_i^*}{w_i}. \qquad (4.59)$$

Коэффициент важности i-го фактора (w_{li}) и k-го фактора (w_{lk}) в l-м подмножестве определяют на основании проведенной экспертизы l-м подмножеством экспертов с использованием выражения (4.45).

В ряде случаев консультируемая проблема функционирует в разных областях. В этих случаях, проводя экспертизу, необходимо учитывать особенности всех клиентов консультируемой проблемы. С этой целью экспертную группу формируют из специалистов, представляющих все области. Экспертизу и определение коэффициентов важности осуществляют по группам. Затем определяют межгрупповые коэффициенты важности

$$w_i = \sum_{l=1}^{q} w_l p_{li},$$

где l — номер группы; q — число групп; w_{li} — коэффициент весомости i-го фактора для l-й группы, представляющей интересы l-й области;

$p_{li} = n_l/n$ — вероятность (частость) использование сформированных рекомендаций консультируемой проблемы в l-й области (здесь n_l — представитель l-й области; n — представитель всех областей).

Пример. Пусть имеется множество факторов F_1, F_2, F_3, F_4, F_5, коэффициенты важности которых определяются тремя группами экспертов. Для проведения экспертизы множество факторов разбито на три подмножества: первое F_1, F_2, F_4,; второе F_2, F_3, F_5 и третье F_2, F_4.

Необходимо определить коэффициенты важности факторов, если эксперты первого подмножества указали коэффициенты важности факторов w_{11}, w_{12}, w_{14}, эксперты второго — w_{22}, w_{23}, w_{25}, и эксперты третьего подмножества — w_{32} и w_{34}. Определим коэффициент важности фактора F_2, входящего во все подмножества. На основании выражения (4.55)

$$w_2 = \frac{1}{3}(w_{12} + w_{22} + w_{32}).$$

W_2 = — (W_{i2} + $W_2$2 + W32>-**o**

Коэффициенты важности остальных факторов, не входящих во все подмножества, определяем на основании выражения (4.59):

$$w_1 = w_{11}\frac{w_2}{w_{12}}; \quad w_3 = w_{23}\frac{w_2}{w_{22}};$$

$$w_4 = \frac{1}{2}\left(w_{14}\frac{w_2}{w_{12}} + w_{31}\frac{w_2}{w_{32}}\right);$$

$$w_5 = w_{25}\frac{w_2}{w_{22}}.$$

4.5.5. Учет компетентности экспертов

В качестве экспертов необходимо выбирать квалифицирован-ных специалистов. Однако зачастую не удается сформировать группу, состоящую из экспертов равной компетентности. В этих случаях необходимо учитывать влияние компетентности каждого эксперта.

Если вероятность того, что оценки k-го эксперта соответствуют действительности, равна p_k, то коэффициент весомости

$$w_i = \frac{\sum_{k=1}^{m} p_k B_{ik}}{\sum_{i=1}^{n} \sum_{k=1}^{m} p_k B_{ik}} \tag{4.60}$$

либо

$$w_i = \frac{\sum_{k=1}^{m} p_k w_{ik}}{\sum_{i=1}^{n} \sum_{k=1}^{m} p_k w_{ik}}, \tag{4.61}$$

где B_{ik} — оценка в баллах i-го фактора k-м экспертом; m — число экспертов; n — число факторов. Величина p_k характеризует компетентность эксперта, ее можно назначать в соответствии с занимаемой должностью, ученой степенью либо на основании квалификационного теста. При равнокомпетентных экспертах выражение (4.61) превращается в выражение (4.45).

Оценивая компетентность эксперта, уточним, что под этим термином подразумевается. Существует три мнения. Одни считают, что компетентность определяется оригинальностью мышления, развитой интуицией. При этом указывают на то, что чрезмерное профессиональное развитие вырабатывает склонность придерживаться сложившихся представлений и не позволяет своевременно замечать намечающихся скачков в развитии наки и техники. Другие считают, что компетентность определяется надежностью суждений, оцениваемой процентом угадываемых свершений. Третьи считают, что компетентность определяется степенью профессионализма.

Очевидно, все три точки зрения правомочны. Первая справедлива при решении задач методом «мозгового штурма», вторая — при прогнозировании и третья — при формировании рекомендаций в области проектирования. При этом необходимо иметь в виду, что речь идет о важнейшей черте эксперта, наряду с которой эксперт должен обладать и двумя другими в той или иной степени в зависимости от консультируемой проблемы. Отсюда следует вывод, что требования к эксперту должны быть аналогичными требованиям к творчески активному научному работнику.

Определение вероятности правильного ответа эксперта в большинстве случаев вызывает затруднение. Вместо вероятности в формулу (4.61) можно ввести коэффициент достоверности k_{∂_k}. При этом полагаем

$$p_k = k_{\text{д}_k}.$$

Коэффициент достоверности показывает степень приближения результатов экспертизы к истинному значению.

При

$$w_{ik} > \| w \| \quad k_{\text{д}_k} = \frac{\| w_i \|}{w_{ik}} < 1;$$

при

$$w_{ik} < \| w_i \| \quad k_{\text{д}_k} = \frac{w_{ik}}{\| w_i \|} < 1,$$

где $\| w_i \|$ —истинное значение коэффициента весомости i-го фактора.

Таблица 4.18

Результаты оценки весомости факторов

Факторы	Эксперты			
	1	2	3	4
1	30	30	20	30
2	20	10	30	20
3	10	20	10	10

Для определения коэффициента достоверности, характеризующего экспертов, в число факторов вводят один или несколько факторов, коэффициенты весомости которых могут быть определены аналитически. Обычно это стоимостные факторы. Если число таких факторов r, то

$$k_{\text{д}_k} = \frac{1}{r} \sum_{s=1}^{r} k_{\text{д}_{ks}}.$$

Пример. Определить коэффициенты весомости экспертным методом с учетом компетентности экспертов. Вероятности получения правильного ответа следующие:

Эксперты...............1 2 3 4

Вероятности ……..0,8 0,2 0,7 0,9

Результаты оценки весомости факторов по шкале интервалов в баллах приведены в табл. 4.18.

Воспользовавшись соотношением (4.60), определим коэффициенты весомости каждого фактора с учетом компетентности экспертов:

$$w_1 = \frac{0,8 \cdot 30 + 0,2 \cdot 30 + 0,7 \cdot 20 + 0,9 \cdot 30}{0,8(30+20+10)+0,2(30+20+10)+0,7(30+20+10)+0,9(30+20+10)} = 0,454;$$

$$w2 = \frac{0,8 \cdot 20 + 0,2 \cdot 10 + 0,7 \cdot 30 + 0,9 \cdot 20}{0,8(30+20+10)+0,2(30+20+10)+0,7(30+20+10)+0,9(30+20+10)} = 0,366;$$

$$w_3 = \frac{0,8 \cdot 10 + 0,2 \cdot 20 + 0,7 \cdot 10 + 0,9 \cdot 10}{0,8(30+20+10)+0,2(30+20+10)+0,7(30+20+10)+0,9(30+20+10)} = 0,18.$$

Воспользовавшись соотношением (4.45), определим те же коэффициенты, полагая одинаковую компетентность экспертов:

w_1 = 0,46; w_2= 0,334; w_3 = 0,206.

Как видно, отличие существенное.

4.6. Структура комплекса моделей для оценки рекомендаций

4.6.1. Задачи, возникающие при построении моделей оценки рекомендаций

Здесь рассматриваются задачи, возникающие при построении комплекса моделей для оценки рекомендаций по решению задач консультируемой проблемы с помощью САК.

Составными элементами консультационной задачи в консультируемой проблеме являются консультационный процесс, цели консультирования, критериальные оценки и ЛФР. Одно из направлений исследований консультирования связано с изучением проблемы выбора или моделированием поведения ЛФР с позиций теории полезности. При рассмотрении этой проблемы будем уделять внимания таким вопросам, как моделирование целей консультационного процесса и оценивание его состояний.

При построении комплекса математических моделей, функционально согласованных между собой относительно фиксированных целей консультационного процесса и консультируемой проблемы, возникает ряд задач. Рассмотрим основные из них.

1. Формализация структуры модели, отражающей оценку степени достижения целей консультационным процессом, возможна при построении дерева целей процесса. Для этого будем использовать экспертные методы. Применяя теорию полезности и методы ранговой корреляции, можно исследовать и оценивать различные структуры моделей целей поведения ЛФР.

Определение 8. Отношение ε называется адекватным для заданных систем с отношениями

$$(Y,\ R_{Y^i},\ i\in I),\ (N,\ R_{N^i},\ i\in I),$$

если для всех γ, $\gamma' \in \Gamma$ и $y_1, y_2, \ldots, y_k \in Y$ имеет место

$$\varepsilon(\gamma(y_1), \ldots, \gamma(y_k)) = \varepsilon(\gamma'(y_1), \ldots, \gamma'(y_k)).$$

Согласно определению 8, адекватные отношения можно записать с помощью следующего критерия.

Критерий адекватности. Отношение ε для систем (Y, R_{Y^i}, $i\in I$) и (N, R_{N^i}, $i\in I$) адекватно тогда и только тогда, когда для фиксированного отображения γ справедливо

$$\Gamma(\bar{Y}, \bar{N}) = \Gamma\{(Y, R_{Y^i}, i\in I, \gamma_k^{-1}(\varepsilon)), (N, R_{N^i}, i\in I, \varepsilon)\}.$$

Таким образом, каждый гомоморфизм системы (Y, R_{Y^i}, $i=1, 2,\ldots, k$) в систему (N_Y, R_{N^i}, $i=1, 2,\ldots, k$) есть гомоморфизм системы с отношениями (Y, R_{Y^i}, $i\in I$), пополненной отношением $\gamma_k^{-1}(\varepsilon)$, в систему с отношениями (N_Y, R_{N^i}, $i\in I$), пополненную отношением ε.

Физический смысл адекватного отношения удачно иллюстрируется примером измерения температуры тела двумя идеальными ртутными термометрами со шкалами Цельсия и

Фаренгейта. Сравнение двух температур T_1 и T_2, полученных в экспериментах, всегда позволит двум экспериментаторам прийти к соглашению относительно истинности отношения $T_1 < T_2$, но никогда — относительно равенства $T_2=2T_1$. Здесь из равенства $T_2=2T_1$ в градусах Цельсия следует, что $T_2 = 2T_1 — 32$ в градусах Фаренгейта, при условии, что температуры по Цельсию и по Фаренгейту связаны зависимостью ^{o}F =1,8°С + 32. Иначе говоря, в этом примере отношение $T_2 = 2T_1$ между двумя действительными числами T_1 и T_2 не имеет смысла без указания шкалы.

Определение 9. Моделью консультационного процесса *P*, адекватной отношению ε из некоторой системы с отношениями

(Y, R_{Y^i}, $i \in I$, γ_k^{-1}(ε)), называется гомоморфизм $\gamma \in \Gamma$ в систему с отношениями (N_Y, R_{N^i}, $i \in I$, ε).

Определение 10. Моделью целей консультирования *G* консультируемой проблемы, адекватной отношению ε из некоторой системы с отношениями (H, R_{H^i}, $i \in I$, λ_k^{-1}(ε)), называется гомоморфизм $\lambda \in \Lambda$ в числовую систему с отношениями

(N_G, R_{N^i}, $i \in I$, ε).

Определение 11. Моделью критериальных оценок состояния *S* консультационного процесса, адекватной отношению ε из некоторой системы с отношениями (F, R_{F^i}, $i \in I$, δ^{-1}(ε)), называется гомоморфизм $\delta \in \Delta$ в систему с отношениями (N_F, R_{N^i}, $i \in I$, ε), причем $S \cap Y \neq \varnothing$.

Определение 12. Моделью поведения ЛФР в ходе управления консультационным процессом *P* при заданных множествах целей *G* и альтернатив *A*, адекватной отношению ε, называется гомоморфизм пересечения $\Xi = \Gamma \cap \Lambda \cap \Delta$ заданной системы с отношениями

(A, R_A^i, $i \in I$, ξ^{-1}(ε)), в систему с отношениями (N, R_{N^i}, $i \in I$, ε).

Определение 13. Повторяющимися рекомендациями в САК называется множество альтернатив

$$A = (a_1, a_2, ..., a_k)$$

с заданным на нем унарным отношением r^A — «иметь место два и более раз в процессе достижения цели консультирования *G*».

Если множество *Y* состоит из эмпирических объектов и отношения R_i на *Y* определены эмпирически, то система (Y, R_{Y^i}, $i \in I$) будет называться эмпирической системой с отношениями.

Определение 14. К-мерными шкалами называются гомоморфизмы

Г, Ξ , Λ, Δ эмпирических систем с отношениями

$$(Y, R_Y{}^i, i\in I, \gamma^{-1}(\varepsilon)), \quad (G, R_G{}^i, i\in I, \lambda^{-1}(\varepsilon)),$$

$$(S, R_S{}^i, i\in I, \delta^{-1}(\varepsilon)), \quad (A, R_A{}^i, i\in I, \xi^{-1}(\varepsilon))$$

в *К*-мерные числовые системы с отношениями

$$(N_Y{}^K, R_N{}^i, i\in I, \varepsilon), \quad (N_G, R_N{}^i, i\in I, \varepsilon),$$

$$(N_S{}^K, R_N{}^i, i\in I, \varepsilon), \quad (N_A, R_N{}^i, i\in I, \varepsilon)$$

соответственно, где N — множество действительных чисел.

Решение задачи поиска такого минимального числа элементов множества выходных показателей консультационного процесса

$$Y=\{y_1, y_2, \ \dots, \ y_n\}, \ n\rightarrow n_{\min},$$

при котором модель целей консультации $\vec{G}$ была бы адекватна некоторому отношению ε в числовой системе с отношениями $\vec{N}$, возможно в случае применения методов последовательного перебора различных вариантов структур моделей целей консультирования. Если при этом на шкале порядка с помощью ранговой корреляции сравнивать выявленные у ЛФР отображения различных состояний процесса P по степени достижения цели консультирования G с отображениями, реализуемыми с помощью моделей оценки целей, то для каждого варианта структуры получаются численные оценки, позволяющие выбрать оптимальную (в некотором смысле) модель. В условиях практической реализации и построения САК это сравнение позволяет повысить быстродействие системы и уменьшить себестимость ее функционирования.

Применение ранговой корреляции делает возможным определение такой шкалы интервалов

$$\lambda=\{\lambda_{\alpha,0}:\lambda\in N^+\}:y\rightarrow\alpha y$$

и таких $\alpha^*\in N^+$, при которых оценка степени адекватности модели G отношению ε достигала бы максимального значения.

2. Формализация процедуры критериального оценивания состояния $s\in S$ **консультационного процесса** осуществляется путем использования конечного числа методов Δ построения функций полезности. При этом с помощью ранговой корреляции получаются числовые оценки степени адекватности отношению ε для конечного множества $\Delta_0\subset\Delta$ методов построения функций полезности:

$$\delta_i:(F, R_F{}^i, i\in I, \delta^{-1}(\varepsilon))-(N_F, R_N{}^i, i\in I, \varepsilon),$$

где $\delta_i \in \Delta$, $\Delta=\{\delta_1, \delta_2,..., \delta_n\}$. Это позволяет более строго обосновать процедуру построения отображения предпочтений и опыта ЛФР.

3. Формализация процедуры построения моделей консультационного процесса возможна при использовании методов сетевого планирования, имитационного моделирования и теории полезности. Таким путем формализуется отношение γ:

$$\gamma: (F, R_F{}^i, i \in I, \gamma^{-1}(\varepsilon)) \rightarrow (N_Y, R_N{}^i, i \in I, \varepsilon) .$$

Проведение исследования влияния множества $S=\{s_1 .. ., s_k\}$ на числовую оценку степени адекватности отношению ε позволяет определить такое значение $k=\{1, 2,...,N\}$, при котором

$$\varepsilon(Y) \rightarrow \max.$$

4. Формализация процедуры упорядочения множества векторных оценок и получение с помощью методов ранговой корреляции числовых оценок степени адекватности различных схем оценок рекомендаций отношению ε

$$\Xi_0 \subset \Xi = \Gamma \cap \Lambda \cap \Delta \text{ для } \xi \in \Xi ,$$

где

$$\xi_0: (A, R_A{}^i, i \in I, \xi_0(\varepsilon)) \rightarrow (N_A, R_N{}^i, i \in I, \varepsilon).$$

5. Исследование различных значений $\alpha \in N^+$, $\sum_i \alpha_i = 1$ и их влияния на числовые оценки степени адекватности отношению ε в отображениях вида ξ_0: $y \rightarrow R$, где R — множество операций свертки,

а ξ_0 — множество алгебраических операций. Это дает возможность найти такую структуру решающего правила, при которой выходы модели, оценивающей рекомендации в шкале порядка, и поведение ЛФР в наибольшей степени совпадают.

6. Исследование методов адаптации и самонастройки комплекса моделей при изменении отношения адекватности ε путем нахождения такого правила коррекции $\Psi_0 \in \Phi$, что

$$\max \rho[\varepsilon\{(N(\bar{A}(\Psi)) \cap \bar{G}(\Psi) \bar{Y}(\Psi)),$$
$$(N(\bar{A} \cap \bar{G} \cap \bar{Y}), R(D))\}] = 1 ,$$

где $\rho(\varepsilon)$ — числовые значения степени адекватности отношению ε на множестве действительных чисел. Такое исследование позволяет алгоритмическим путем изменять структуру и параметры комплекса моделей для оценки рекомендаций без дополнительного вмешательства оператора (консультанта) и перестраиваться на изменение как поведения консультационного процесса, так и предпочтений ЛФР, а также схем оценки рекомендаций органа консультирования.

4.6.2. Функциональные взаимосвязи комплекса моделей оценки рекомендаций

Для построения и исследования моделей повторяющихся рекомендаций необходим комплекс четырех согласованных моделей:

а) модель оценки степени достижения цели консультирования G;

б) модель критериального оценивания C;

в) модель консультационного процесса P;

г) модель поведения ЛФР D.

В терминах теории множеств отношения между моделями могут быть представлены следующей системой:

$P{:}A\times\rightarrow U\times X\times T\rightarrow Y\times Y'\times T;$ (4.62)

$D{:}A\times W\times H\times T\rightarrow A\times A\times T;$ (4.63)

$C{:}X\times A\times Y\times T\rightarrow F\times T;$ (4.64)

$G{:}K\times F\times T\rightarrow H\times T;$ (4.65)

$N\subset U\times U\times T;$ (4.66)

$J\subset K\times K\times T;$ (4.67)

$M\subset W\times W\times T;$ (4.68)

$R\subset H\times H\times T.$ (4.69)

Здесь отношения в формальной записи описывают: P — модель консультационного процесса; A, U, X — множества входов для модели консультационного процесса, где A — множество альтернатив (консультационных воздействий), U — подмножество изменяемых отношений множества R_Y^i, $i\in I$, X — множество нормативных (технологических) входов, не пересекающееся с множеством A, $U\in R_Y^i$, $i\in I$; Y и Y' — множества выходов консультационного процесса; Y — множество контролируемых выходов модели P; Y' — множество неконтролируемых выходов модели; D — модель поведения ЛФР со множествами входов A, H, W, где W — подмножество изменяемых отношений множества R_H^i, $i\in I$, $W\in R_H^i$, H — множество интегральных критериальных оценок, служащих для оценки состояния P по достижению G в момент времени $t\in T$. В данном случае H представляет оценки достижения цели G; F есть декомпозиция H и представляет оценки свойств цели G, т. е. $H\supseteq F$.

Множество входов $A=\{a_1, \ldots, a_n)$ модели D представляет собой множества альтернатив с известными отношениями между его элементами $(A, R_A^i, i\in I)$.

C — модель критериальных оценок со множествами входов A, Y, T, X и множествами выходов F, T, где F — множество критериальных оценок состояния консультируемой проблемы по достижению цели G, т. е.

$$(F,\ R_F{}^i,\ i\in I)\rightarrow(N_F,\ R_N{}^i,\ i\in I).$$

G —модели цели консультирования проблемы со множествами входов K, F, T, где K — подмножество изменяемых отношений множества R_F^i, $i \in I$.

N, M, J — спецификация подмножеств изменяемых отношений R_Y^i R_H^i, R_F^i, $i \in I$ соответственно; R — отношение глобальной целенаправленности в целом.

Систему отношений (4.62) —(4.69) можно представить графически (рис. 4.19).

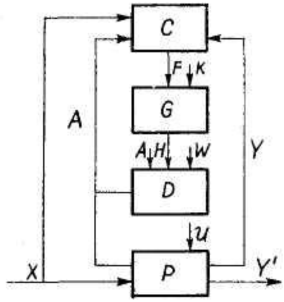

Рис. 4.19. Структура комплекса моделей САК

Рассмотрим более подробно на теоретико-множественном уровне вопросы, которые необходимо решить при построении оптимальных в некотором смысле САК.

4.6.3. Построении моделей оценки степени достижения цели консультирования и критериальных оценок и их исследование

При построении модели G оценки степени достижения цели консультирования реальным процессом P в САК необходимы, как это уже отмечалось в п. 4.6.2, модель поведения ЛФР D и система измерения выходов процесеса Y.

Обозначим через $\bar{H}$ систему с отношениями

$$\bar{H} = (H, R_H),$$

сформированную в сознании ЛФР и отображающую категории и понятия, соответствующие целям консультирования

$$h: \{h_1, h_2, \ldots, h_k\} = H.$$

Отношения между целями консультирования h_j и h_k обозначим символами r_h^j, причем пока не будем накладывать никаких ограничений на тип отношений r_h^i. Это могут быть любые логические и другие высказывания, отображенные на шкалы наименований.

При построении комплекса моделей для оценки рекомендаций на первом этапе необходимо с помощью отображений $\lambda_h \in \Lambda$ перейти на шкалы наименований, описанные в п. 4.4.2, и получить систему с отношениями

$$\bar{H}_H=(N_H,R_{NH}),$$

которая будет гомоморфизмом системы H с отношениями R. Необходимо отметить, что λ_n практически представляют собой процедуры экспертного анкетирования и в простейшем случае сводятся к процедуре задания ЛФР спецификаций целей с помощью символов. В теоретико-множественном описании это можно представить в виде

$$\lambda_n:(H,R_H)+(N_H,R_{NH}),$$

где R_{NH} — множество отношений на шкале наименований между образами на N_H, которые соответствуют элементам N.

При построении моделей целей консультирования в ряде задач оказывается возможной декомпозиция наименования λ_j на составляющие элементы

$$f_j=\{f_{1j},\ f_{2j},\ \ldots,\ f_{kj}\},$$

где f_{kj} — k-я составляющая j-й цели консультирования в сознании ЛФР.

Очевидно, что с помощью отображений $\delta \in \Delta$на шкале наименований получается числовая система $\bar{N}_F$, которая будет образом системы с отношениями $\bar{F}=\{F,\ R_F)$. Иначе говоря, можно записать, что

$$\delta:(F;R_F)\rightarrow(N_F;R_{NF})\ ;$$

$$(F;R_F)\rightleftarrows(H;R_H)\ ;$$

$$(N_F;R_{NF})\rightleftarrows(N_H;R_{NH})\ .$$

Шкалы наименований практически не позволяют выполнять арифметические операции, необходимые для оценки рекомендаций. Чтобы оценить состояние консультируемого объекта, необходимо с помощью отображений λ_π и δ_π получить на шкалах порядка образы

$$\bar{N}_{FY}=(N_{FY};R_{NF})$$

и

$$\bar{N}_{HY}=(N_{HY};R_{NH})$$

соответственно.

Если на предыдущих этапах процедуры измерения h представляли собой способы получения наименований целей консультирования (δ —

информационный процесс выявления составляющих целей консультирования), то λ_π и δ_π — процессы передачи информации (отображения) об отношениях порядка соответственно R_{NF}^{π} и R_{NH}^{π} систем с отношениями $\overline{N}_F$ и $\overline{N}_H$ в числовые системы $\overline{N}_{FZ}=(N_{Fz}; R_{NF})$ и $\overline{N}_{HZ} = (N_{HZ}; R_{NH})$, где $R_{NH}{}^{\pi}$ — множество отношений на шкале порядка П для числовых наименований компонент N_H целей консультирования; $R_{NF}{}^{\pi}$ — множество отношений на шкале порядка П для числовых наименований компонент N_F целей консультирования.

Отношениям порядка и эквивалентности в числовых системах соответствуют отношения предпочтения и безразличия в САК. В ряде случаев ЛФР имеет возможность применить для консультирования множество альтернатив A с отношениями R_A. Множество альтернатив, отраженное в сознании ЛФР, обозначим через систему с отношениями

$$\overline{A} = (A, R_A).$$

Ранее термин «гомоморфизм» употреблялся для описания процесса отображения эмпирической системы с отношениями в числовую систему с отношениями. В САК этому процессу может соответствовать *процесс передачи информации от человека к системе, т. е. информационный процесс.*

Пусть существуют информационные процессы $d\in D$, позволяющие отобразить в шкалах наименований систему с отношениями

$$\overline{A} = (A, R_A).$$

$$d:(A,R_A)\rightarrow(N_A;R_{NA}).$$

При отображении системы с отношениями $\overline{N}_A$ на шкалу порядка для целей консультирования, представленных моделью $\overline{N}_{HY}$, получается числовая система $\overline{N}_{HA}$, на которой множество альтернатив A упорядочено по какому-либо признаку, например по степени достижения целей G.

Воспользуемся методами теории полезности, позволяющими практически осуществить отображение λ'_π , δ'_π для предпочтений ЛФР по достижению целей консультирования H при заданном консультационном процессе P и множестве альтернатив A:

$$\lambda'_\pi: (N_{HZ}; R_{NH})\rightarrow(N_{HA}; R_N);$$

$$\delta'_\pi: (N_{FY}; R_{NF})\rightarrow(N_{FA}; R_N),$$

где $\lambda_{\pi'}$ — методы теории полезности, реализованные в САК для отображения отношений между $\overline{N}_H$ и $\overline{N}_{HA}$ на шкале порядка; δ'_π —

методы теории полезности, реализованные в САК для отображения отношений между $\overline{N}_F$ и $\overline{N}_{FA}$ на шкале порядка.

При этом система с отношениями $\overline{N}_{FA}$ есть конечный этап декомпозиции системы с отношениями $\overline{N}_{HA}$. Реализация этого этапа в ходе построения моделей САК обусловливается неповторяющимися ответами ЛФР. Значит, можно записать

$$(N_{HA}; R_N) \rightleftarrows (N_{FA}; R_N)\ .$$

Взаимосвязи между гомоморфизмами и шкалами при построении модели оценки степени достижения цели консультирования и модели критериальных оценок состояния консультационного процесса показаны на рис. 4.20.

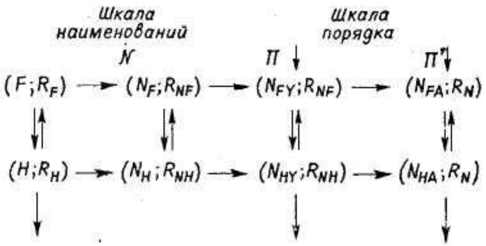

Рис. 4.20. Взаимосвязь шкал *N* и П с моделями *G* и *C*.

Под <u>*моделью целей консультирования*</u> *G САК, адекватной отношению ε заданной системы с отношениями* $(H;\ R_H{}^i),\ i \in I,\ \lambda^{-1}(\varepsilon))$, *согласно определению 10, понимается гомоморфизм* $\lambda \in \Lambda$ *в числовую систему с отношениями* $(N_H,\ R_N{}^i,\ i \in I,\ \varepsilon)$. Для исследования свойств модели оценки степени достижения цели числовая система с отношениями N_H отображается на множества N_{HY} и N_{HA} такие, что

$$(N_H; R_N{}^i, i \in I, \varepsilon) \to (N_{HY}; R_N{}^i, i \in I, h(\varepsilon)) \wedge (N_{HA}; R_N{}^i, i \in I, h(\varepsilon)),$$

где N_{HY} есть композиция $\gamma_Y \bigcirc \gamma_A \bigcirc \gamma_Y{}^{-1} \bigcirc \delta^2$ отображений γ_Y: $Y \to A$; δ_Y: $Y \to H$, λ_0:$H \to H$. Определяется такое подмножество $Y_0 \subset Y$, для которого $\rho(\varepsilon) \to \max=1$ или $\max\ \rho(\varepsilon\ (Y_0,\ \lambda_0)) \Longrightarrow 1$, где $\rho(\varepsilon)$ — числовое значение степени адекватности отношению ε на множестве действительных чисел..

Другими словами, *находятся такое подмножество* $Y_0 \subset Y$ *и такое отображение* λ_0*:H→H, при которых числовая оценка степени адекватности модели G будет* <u>*максимальной*</u>.

Под <u>*моделью критериальных оценок состояния консультационного*</u> <u>*процесса*</u>, *адекватной отношению ε заданной* *системы с отношениями (Н; R_H^i ,$i\in I$, $\delta(\varepsilon)$), будем понимать, согласно определению 11, гомоморфизм $\delta\in\Delta$ в систему с отношениями (N_F, R_N^i ,$i\in I$, ε)=* $\bar{N}_F$. Исследуя эту модель, отобразим числовую систему с отношениями $\bar{N}_F$ на множества N_{FA} и N_{FY} такие, что

$$(N_F; R_N{}^i, i\in I, \varepsilon)\rightarrow(N_{FA}(U(A)); N_{FY}(U(Y)); R_N{}^i, i\in I, \delta(\varepsilon)),$$

где *U(A)* есть образ, отображения δ_A:$A\rightarrow U$; *U(Y)* есть образ отображения δ_Y: $Y\rightarrow U$.

Исследование δ_A и δ_Y предполагает поиск таких отображений δ_{OA} и δ_{OY}, при которых числовая оценка степени адекватности принимает *максимальное* значение:

$$\max\rho(\varepsilon(\delta_{OA}, \delta_{OY}, R(\bar{N}_F)))=1,$$

где $R(N_F)$ — совокупность отношений на множествах N_{FA} и N_{FY}.

4.6.4. Построение модели консультационного процесса для формирования рекомендации и ее исследование

Под моделью, адекватной отношению ε для консультационного процесса P и заданной системы с отношениями

$$\bar{Y}=(Y; R_Y{}^i, i\in I, \gamma^{-1}(\varepsilon)),$$

будем понимать, согласно определению 5, гомоморфизм $\gamma\in\Gamma$ в числовую систему с отношениями

$$\bar{N}_Y=(N_Y; R_N{}^i, i\in I, \varepsilon).$$

Для практического изучения свойств моделей в САК отобразим $\bar{N}_Y$ в следующую систему с отношениями:

$$N_{HYA}=d_Y\circ\gamma_N\circ\gamma_F\circ\gamma_H; \quad d_Y: A\rightarrow\bar{Y}_A; \quad \gamma_N:\bar{Y}_A\rightarrow\bar{N}_{YA};$$

$$\gamma_F=\bar{N}_{YA}\rightarrow\bar{N}_{FYA}; \quad \gamma_H:\bar{N}_{FYA}\rightarrow\bar{N}_{HYA}.$$

В этой модели консультационного процесса наиболее важным является отображение d_Y: $A\rightarrow Y_A$, которое можно исследовать косвенно с помощью системы с отношениями N_{YA}. Для этого необходимо найти такое отображение $\gamma_{ON}(\bar{Y}_A)\in\Gamma$, что

$$\gamma_{ON}(\bar{Y}_A)=d_{OY}\circ\gamma_{ON}=N_{ONA};$$

$$\max\rho(\varepsilon(\gamma_H(\bar{Y},\bar{A}), \gamma_H(\bar{N}_{OYA},\bar{N}_A)))\Rightarrow 1.$$

Взаимосвязи между моделью консультационного процесса, моделью оценки степени достижения целей и моделью критериального оценивания показаны на рис. 4.21, где *N* обозначает шкалу наименований, а П — шкалу порядка.

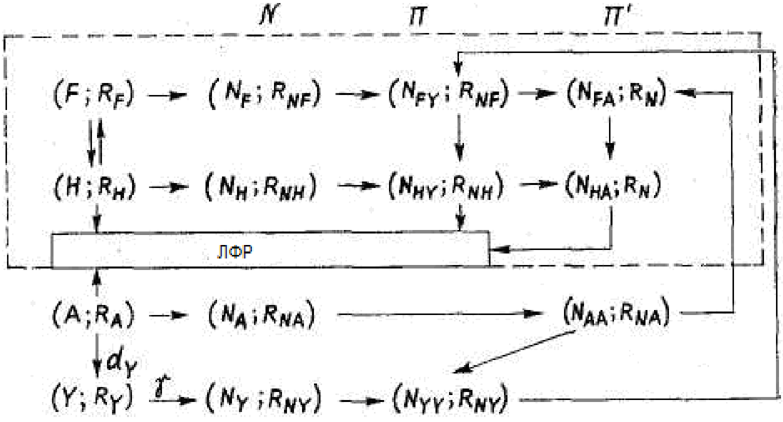

Рис. 4.21. Взаимосвязь шкал *N* и П с моделями *P, G* и *C*.

4.6.5. Построение модели правила выбора ЛФР и ее исследование

Под моделью, адекватной отношению ε, правила выбора ЛФР при управлении процессом P, заданным на множестве целей H и множестве альтернатив A, понимается, согласно определению 12, гомоморфизм заданных систем с отношениями

$$(A, H, Y, D;\ R_A{}^i, R_D{}^i, R_Y{}^i, R_H{}^i, i\in I, \xi^{-1}(\varepsilon))$$

в системы с отношениями

$$\bar{N}_{AY}{}^{\Xi} = \bar{N}_A{}^{\Xi} = (N_A; \xi_A(R_{NA}{}^i), i\in I, \xi_A\in\Xi, \varepsilon);$$

$$\bar{N}_Y{}^{\Xi} = (N_Y; \xi_Y(R_{NY}{}^i), i\in I, \xi_Y\in\Xi, \varepsilon);$$

$$\bar{\Xi} = (\Xi, R_{\Xi}{}^i, i\in I).$$

Для исследования свойств моделей при их построении выбираются множество решающих правил Ξ и отношений на нем $R_{\Xi}{}^i$, $i\in I$, а также отображения ξ_A и ξ_Y с отношениями $R_{\xi A}{}^i$, $R_{\xi Y}{}^i$, $i\in I$. Определяются такие ξ_{OA}, $\xi_{OY}\in\bar{\bar{\Xi}}$, $R_{O\xi A}{}^i$, $R_{O\xi Y}{}^i$, $i\in I$, при которых числовая оценка ρ степени адекватности ε достигает максимума:

$$\max \rho(\varepsilon(\xi_{OA}(\bar{A}, \bar{H}, \bar{Y}), \xi_{OA}(\bar{N}_A, \bar{N}_H, \bar{N}_Y)))\to 1.$$

Взаимосвязи моделей правила выбора с другими моделями показаны на рис. 4.22.

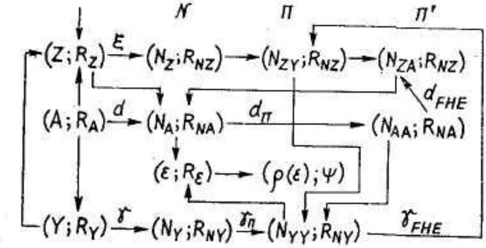

Рис. 4.22. Взаимосвязь шкал *N* и П с моделями *D, P, G* и *C*.

На этом рисунке *N* обозначает шкалу наименований, П — шкалу порядка для отображения параметров консультируемой проблемы, П' — шкалу порядка для отображения множества альтернатив. Символы, заключенные в круглые скобки, обозначают эмпирические системы с отношениями. При этом $(z;\ R_z)$ обозначает моделированную систему целей консультирования ЛФР, $(A;\ R_A)$ — моделируемое множество альтернатив ЛФР, $(Y,\ R_Y)$ — моделируемая консультируемая проблема. $(N_z;\ R_{NZ})$, $(N_A;\ R_{NA})$, $(R_{NY},\ N_Y)$ — гомоморфизмы (модели) целей консультирования, альтернатив и консультируемой проблемы в шкале наименований, полученных с помощью процедур ξ, *d,* γ соответственно, реализация которых на практике может осуществляться в диалоговом режиме ЛФР и САК.

На рис. 4.23 представлены взаимосвязи между задачами исследований при построении комплекса моделей для оценки рекомендаций. Здесь система с отношениями $(Z,\ R_z)$ декомпозирована на две системы $(H;\ R_H)$ и $(F;\ R_F)$, причем первая соответствует моделируемому множеству целей консультирования, а вторая — моделируемому множеству свойств целей консультирования.

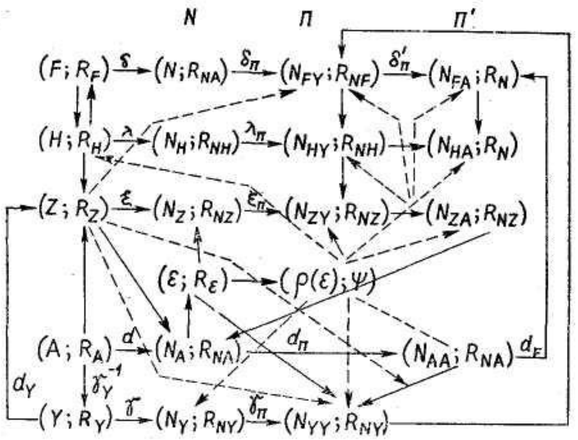

Рис. 4.23 Взаимосвязи между задачами исследований при построении моделей *D, P, G* и *C*.

Система с отношениями (Ξ; R_Ξ) соответствует моделируемому множеству стратегий или правил выбора ЛФР альтернативы $a^0 \in A$ в ситуации *(H, A, Y*; R_H, R_A, R_Y*)*. При этом пара ($\rho(\varepsilon)$; Ψ) обозначает множество методов настройки и коррекции построенных

моделей по выбранному критерию адекватности ε и его числовому значению $\rho(\varepsilon)$.

В связи с поставленными задачами исследований возникает проблема определения отношения ε адекватности моделей и числовой оценки степени адекватности в случае, если модели не строго адекватны.

4.6.6. Критерии адекватности моделей использования повторяющихся рекомендаций поведению системы

Выбор критерия адекватности моделей использования повторяющихся рекомендаций поведению системы до сих пор не рассматривался в теории консалтинга. Между тем задача сравнения построенных моделей с реальным объектом требует введения и обоснования *числового критерия*.

По определению 8, *отношение ε называется адекватным для заданных систем с отношениями (Y; R_Y^i, $i \in I$, $I = 1, 2, ..., n$),* $\overline{N}$ *, если для всех $\gamma, \gamma' \in \Gamma$ и $y_1, y_2, ..., y_k \in Y$,*

$$\varepsilon(\gamma(y_1)\gamma(y_2), \ldots, \gamma(y_k)) = \varepsilon(\gamma'(y_1), \gamma'(y_2), \ldots, \gamma'(y_k)).$$

Из этого следует, что каждый гомоморфизм системы (Y; R_Y^i, $i \in I$) в систему (N_Y, R_{NY}^i, $i \in I$) есть гомоморфизм системы с отношениями

(N_Y, R_{NY}^i, $i \in I$), пополненной отношением $\gamma^{-1}(\varepsilon)$, в систему с отношениями (N_Y, R_{NY}^i, $i \in I$), пополненную отношением ε.

Наиболее общим отношением, которое объединяет как все модели, перечисленные в п. 4.6.2, так и реальные объекты, является отношение порядка, или предпочтения, r^2 (подробнее см. в п. 4.7.5). В теории статистических решений для сравнения степени адекватности порядков двух или более систем упорядоченных элементов применяют методы ранговой корреляции, которые позволяют определить степень и направление связи. Действительно, при самом строгом подходе отношение порядка на множестве альтернатив, назначенное ЛФР для достижения заданной цели G при исходном состоянии консультационного процесса Y^*, является существенным результатом, определяющим как функцию выбора $\xi_A(A)$ в модели, так и поведение ЛФР. В дальнейшем при переходе к шкалам интервалов множество упорядоченных элементов отображается в метрическое пространство с расстоянием $d(a_1, a_2)$ между ними. Для вычисления коэффициентов ранговой корреляции множество элементов упорядочивается. При этом самому предпочтительному исходу или альтернативе присваивается ранг 1, менее предпочтительному — 2 и т. д. В случае, когда элементы множества эквивалентны, им присваивается усредненный ранг, который вычисляется по формуле:

$$r_j = \overline{n, k} = \frac{\sum_{j=n}^{k} r_j}{n-k+1}.$$

Для вычисления коэффициентов корреляции рангов по Спирмену без учета одинаковых (усредненных) рангов используется выражение

$$\rho = 1 - \frac{6\sum_i d_i}{n(n^2-1)},$$

где ρ — коэффициент ранговой корреляции Спирмена; $d_i = r_i - r_i'$ — разность i-х рангов переменной A и A' для комплекса моделей и ЛФР соответственно; n — число сопоставленных пар: i=l, 2,..., n. Если среди рангов имеются одинаковые (усредненные), то используется выражение

$$\rho=1-\frac{6(\Sigma_i d_i^2+T_A+T_{A'})}{n(n-1)}, \tag{4.70}$$

где

$$T_A=\frac{1}{12}\Sigma_j(t_A{}^3-t_A)_j; \quad T_{A'}=\frac{1}{12}\Sigma_k(t_{A'}{}^3-t_{A'})_k;$$

t_A — количество одинаковых (усредненных) рангов множества A; $t_{A'}$ — количество таких рангов в переменной A' в k-й и j-й группах с одинаковыми рангами; k, j — номера групп с одинаковыми рангами.

Коэффициент ранговой корреляции по Кендаллу можно вычислить несколькими способами. Наиболее простой способ заключается в том, что для упорядоченной группы элементов и присвоенных им рангов записывают рядом ранги другой группы аналогичных элементов.

Коэффициент Кендалла определяется по формуле

$$\tau=\frac{2\Sigma_i z_i}{0{,}5n(n-1)}-1,$$

где τ — коэффициент ранговой корреляции Кендалла; z_i — количество рангов второй переменной, начиная с i+l, значение которых превышает значение i-го ранга этой переменной; i — номер (ранг) первой переменной, i=l, 2,...,n.

Оба коэффициента ранговой корреляции изменяются в пределах — 1≤ ρ ≤1, — 1≤ τ ≤1. По модулю они характеризуют степень связи, а по знаку — направление. Коэффициент Кендалла τ имеет некоторые недостатки по сравнению с коэффициентом Спирмена ρ: во-первых, τ не учитывает усредненные ранги; во-вторых, более сложна техническая реализация вычислений τ.

Для вычисления степени совпадения полученного с помощью модели P_i предсказания с наблюдавшимися значениями A_i, предлагается использовать коэффициент Тейла:

$$K=\frac{\sqrt{1/n\Sigma_i(P_i-A_i)^2}}{\sqrt{1/n\Sigma_i P_i^2}+\sqrt{1/n\Sigma_i A_i^2}},$$

который изменяется в пределах 0≤ K ≤1. Если предсказание, полученное на модели, полностью осуществилось, то K =0; при неоправдавшемся предсказании K =1. В случае ранжированных переменных P и A коэффициент Тейла менее информативен относительно степени совпадения результатов моделирования и реального процесса, чем коэффициент Спирмена, ввиду невозможности сравнивать элементы, перенумерованные в обратном порядке. В связи с этим целесообразно для сравнения степени адекватности моделей отношению порядка использовать коэффициент ранговой корреляции Спирмена.

4.7. Построение моделей целей консультирования и критериального оценивания консультируемой проблемы

4.7.1. Интерактивные процедуры и построение моделей для оценки рекомендаций

В настоящее время достигнуты определенные успехи при использовании человеко-машинных процедур в САК. Под такими процедурами будем понимать диалог между человеком и САК. В общем случае диалог является сложным ступенчатым процессом взаимного обучения ЛФР и комплекса математических моделей, заложенных в САК. Схематично процесс такого диалога представлен на рис. 4.24.

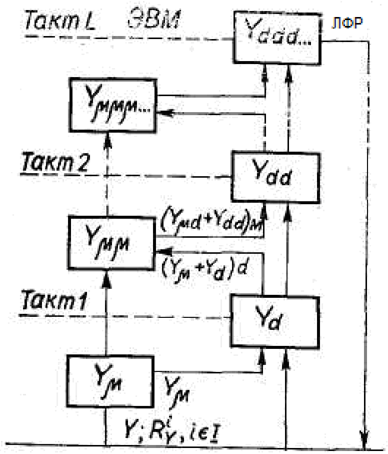

Рис. 4.24. Схема процесса диалога САК и ЛФР.

Очевидно, что, подобно ЛФР, САК в определенном смысле отражает ситуацию, возникшую в данный момент в ходе формирования рекомендации. Простейший диалог при получении справки формально описывается полиномом:

$$B^l=(\bar{Y}+B^{l-1}\mu)d; \quad l=\overline{1, L}; \quad B^0=Y,$$

где μ — состояние САК; d — состояние ЛФР; l — номер такта диалога; член B^l описывает сведения, полученные ЛФР от САК; $B^{l-1}\mu$ описывает сведения ЛФР об управлении консультационным процессом

$\bar{Y}=(Y;\ R_Y^i,\ i\in I)$. В случае, если ЛФР принимает во внимание как состояние окружающей среды, учтенное комплексом моделей, так и собственное восприятие ситуации, диалог может быть описан полиномом

$$B^l=[(\bar{Y}, \bar{H}, \bar{A})+\{(\bar{Y}, \bar{H}, \bar{A})+$$

$$(B^{l-1}\Xi)z\}\mu]d,$$

$$B^0=(\bar{Y}, \bar{H}, \bar{A});$$

$$l=\overline{1, L}, \ \Xi=(\Xi; R_\Xi^i, i\in I),$$

где z — комплекс математических моделей; H — множество целей; A — множество альтернатив; Y — множество векторов оценок состояний консультационного процесса; Ξ — множество правил выбора на множестве числовых систем с отношениями $\bar{N}_A$, $\bar{N}_Y$, $\bar{N}_H$, являющихся взаимно-однозначным отображением эмпирических систем $\bar{A}$, $\bar{Y}$, $\bar{H}$.

Применение диалоговых процедур, реализованных в моделях САК, сокращает время получения от ЛФР экспертной информации об отношении порядка.

4.7.2. Процедуры интерактивного отображения дерева целей консультирования и его структуры

Воспользуемся отображением λ^{-1} эмпирической системы с отношениями $(H,\ R_H^i,\ i\in I,\ \lambda^{-1}(\varepsilon))$, адекватной отношению r^2, для определения области значений в эмпирической системе с отношениями $(Y;\ R_Y^i,\ i\in I,\ \lambda^{-2}(\varepsilon))$. Здесь $\lambda^{-2}(\varepsilon)$ обозначает гомоморфизм $\lambda^{-1}(\lambda^{-1}(\varepsilon))$, который, согласно определению 9, является областью определения модели консультационного процесса. В этом случае находится такое

$\lambda_0\in\Lambda$, при котором численная мера адекватности отношению ε модели оценки степени достижения целей $\bar{N}_{HY}$ $(N_{HY};\ R_{NH}^i,\ i\in I,\ \varepsilon)$ принимает максимальное значение, т. е.

$$\rho(\varepsilon(\lambda_0(\bar{Y}, \bar{H}), \lambda_0(\bar{N}_Y, \bar{N}_H)))=\max.$$

Одновременно определяется такое $Y_0 \in Y$, при котором образ множества Y_0 в случае отображения λ_0 принадлежит цели консультирования $\overline{H}$, а значение численной меры адекватности отношению ε принимает максимальное значение, т. е. соответствует определению 8 и положениям п. 4.6.6.

Отображение $\lambda \in \Lambda$ эмпирической системы с отношениями реализуется на САК, состояние *l*-го такта которой в интерактивном режиме можно представить полиномом

$$B^l = [(\overline{Y}, \overline{H}) + \{(\overline{Y}, \overline{H}) + (B^{l-1}\Xi)z\}\mu]d,$$

$$B^0 = (\overline{Y}, \overline{H}), \quad l = \overline{1, L},$$

где μ — состояние САК; *z* — выход комплекса математических моделей; *d* — состояние ЛФР; *Y* — консультационный процесс; $\overline{H}$ — цель консультируемой проблемы.

Моделью цели консультирования G, согласно определению 10, будем называть гомоморфизм $\lambda \in \Lambda$ для эмпирической системы с отношениями $\overline{H}$ = (H; R_H^i, $i \in I$, $\lambda^{-1}(\varepsilon)$), адекватной отношению ε, в эмпирическую систему с отношениями Y=(Y; R_Y^i, $i \in I$, $\lambda^{-2}(\varepsilon)$), отображенной в числовую систему с отношениями

$$(N_{NY}; R_{NH}{}^i, i \in I, \varepsilon) = \overline{N}_{HY}.$$

Определение 15. Условием неопределенности построения модели цели консультирования будем называть гомоморфизм $\lambda^{-1} \in \Lambda$ для эмпирической системы (*H*; $\lambda^{-1}(\varepsilon)$), адекватной отношению ε, в числовую систему (N_{NY}; R_{NH}^i, $i \in I$, ε), причем множество

$$(R_{NH}^i)\ i \in I = \varnothing.$$

Определение 16. Моделью оценки степени достижения цели консультирования *G* будем называть гомоморфизм $\lambda^{-1} \in \Lambda$ для эмпирической системы с отношениями (*H*, R_H^i, $i \in I$, $\lambda^{-1}(\varepsilon)$), адекватной отношению ε, в эмпирическую систему с отношениями

(*Y*; $R_Y{}^i$, $i \in I$, $\lambda^{-2}(\varepsilon)$), отображенной в числовую систему с отношениями (N_{HY}; $R_{NH}{}^i$, $i \in I$, ε), такую, что

$$\mathsf{E}\ (R_{ONH}{}^i, i \in I) : R_{ONH}{}^i \in R_{NH}{}^i, i \in I, R_{NH}{}^i = \{r^1, r^2\},$$

т. е. композицию

$$\gamma_n \bigcirc \gamma_F \bigcirc \gamma_H \bigcirc \zeta_{NH}.$$

Введем определение композиции.

Определение 17. Пусть *H, Y, N* — множества, λ^{-2} — отображение из *H* в *Y*, γ_n — отображение из *Y* в N_Y. Композицией называется отображение из $H \rightarrow N_Y = \lambda_0^{-1}\gamma_N$, определенное равенством

$$\gamma_N \bigcirc \lambda^{-1}(h) = \gamma_N(\lambda^{-1}(h)).$$

Теорема 1. Пусть *(H; $R_H{}^i$, $i \in I$); (Y; $R_Y{}^i$, $i \in I$); (N; $R_N{}^i$, $i \in I$)* — системы с отношениями. Если λ^{-1} — гомоморфизм из *H* в *Y*, λ_n — гомоморфизм из *Y* в *N*, то $\gamma^{-1} \bigcirc \gamma$ — гомоморфизм из *N* в *H*.

Доказательство. Утверждение теоремы непосредственно следует из определений 3 и 17.

Определение 18. Для заданной системы с отношениями *(Y, $R_Y{}^i$, $i \in I$)* отношение R_Y^i называется произведением отношений R_Y^2 и R_Y^3, если выполняется следующее условие:

$$\forall y_p, y_q, y_l \in Y; \quad (y_p R^1 y_q) \leftrightarrow (y_p R^2 y_l) \wedge (y_l R^2 y_q).$$

Теорема 2. Систему с отношениями *n*-го порядка можно разложить на *n* — 2 трехместных отношения.

Доказательство. Представим отношение R^0 системы в виде произведения двух других отношений R^1 и R^2, т. е. $R^0 = R^1 R^2$.

$$R^1(y_{k1}, y_2, y_3) \wedge R^2(y_1, y_{k1}, y_4, \ldots, y_n).$$

Поскольку вместе с новыми отношениями R^1 и R^2 введен новый терм y_{k1}, на выбранные отношения можно не накладывать никаких ограничений, и, следовательно, такая декомпозиция возможна. Представим теперь R^2 как произведение отношений $R^3 R^4$, тогда

$$R^1(y_{k1}, y_2, y_3) \wedge R^3(y_{k2}, y_{k1}, y_4) \wedge R^4(y_{k2}, y_1, y_5, \ldots, y_n).$$

Продолжая этот процесс, получим

$$R^{2(k-1)} = R^{2(k-1)+1} R^{2(k-1)+2}, \quad k = 1, \ldots, (n-3),$$

или в более общем случае:

$$R^1(y_{k1}, y_2, y_3) \wedge R^3(y_{k2}, y_1, y_4) \wedge \ldots$$
$$\ldots \wedge R^{2(n-4)+1}(y_{n-3}, y_{n-4}, y_{n-2}) \wedge R^{2(n-4)+2}(y_{k(n-3)}, y_1, y_2). \tag{4.71}$$

На вводимые отношения не накладывается никаких ограничений, поэтому разложение возможно и выражение (4.71) содержит ровно *n* — 2 трехместных отношения, что и требовалось доказать.

Теорема 3. Систему *n*-го порядка можно разложить на двухместные отношения тогда и только тогда, когда любое трехместное отношение, полученное из теоремы 2, удовлетворяет условию

$$[Y_i R_j (Y_{i+1}, Y_{i+2})] \leftrightarrow \{(Y_i R_j^1 Z_j) \wedge [Z_j R_j^2 (Y_{i+1}, Y_{i+2})]\};$$
$$Z_j = Y_{i+1} \cup Y_{i+2}.$$

Доказательство. Рассмотрим одну из подсистем отношения

$$[R_1(Z_1, Y_2, Y_3)] \wedge [R_3(Z_2, Z_1, Y_4)] \wedge \dots$$

$$\dots \wedge [R_{2(n-1)+1}(Z_{n-3}, Z_{n-4}, Y_{n-2})] \wedge [R_{2(n-4)+2}(Z_{n-3}, Y_1, Y_n)]$$

и введем для удобства новые обозначения для термов:

$$R_j=(K_j^1, K_j^2, K_j^3)$$

где K_j^1, K_j^2, K_j^3 — соответствующие термы. Запишем R_j в виде следующего произведения отношений:

$$R_j=R_j^1R_j^2; \quad [R_j^1(Z^1, K_j^3)] \wedge [R_j^2(Z^1, Y_j^1, Y_j^2)].$$

Предположим, что условие теоремы, состоящее в том, что новый терм Z^1 совпадает с одним из термов K_j^1 или K_j^2 выполнено. Обозначим промежуточный терм Z через K_j^2. Тогда

$$[R_j^1(K_j^2, K_j^3)] \wedge [R_j^2(K_j^2, K_j^1, K_j^2)] =$$

$$= [R_j^1(K_j^2, K_j^3)] \wedge [R_j^2(K_j^1, K_j^2)].$$

Следовательно, подсистему R_j удалось разложить на два двухместных отношения. Поскольку такое разложение возможно для всех j, вся система в целом разлагается на $2(n—2)$ подсистемы двухместных отношений. Это доказывает достаточность условий теоремы. Покажем их необходимость. Предположим, что существует трехместное отношение, для которого ни один промежуточный терм не совпадает ни с одним из трех термов исходного отношения, тогда

$$R_j(K_j^1, K_j^2, K_j^3) = R_j = R_j^1R_j^2;$$

$$[R_j^1(K_j^1, Z^1)] \wedge [R_j^2(Z^1, K_j^2, K_j^2)].$$

Здесь Z^1 — новый терм, и, следовательно, второе отношение трехместно.

Интерактивные процедуры, используемые для формального перехода системы с отношениями *(Н,* R_H^i, $i \in I$*)* в систему с отношениями *(N,* R_{HN}^i, $i \in I$*),* практически позволяют реализовать отображение

$$\lambda: H \to N_H, R_H^i \to R_{NH}^i,$$

где H — отображение целей консультирования в сознании ЛФР; R_H — отображение множества отношений на H в сознании ЛФР; N_H — образ множества целей консультирования в модели; R_{NH} — отображение множества отношений на множестве N_H в модели.

Отображая систему с отношениями *(Н,* R_H^i, $i \in I$*)* в шкалу наименований в соответствии с отношением эквивалентности r^1, получаем гомоморфизм на множестве всех наименований. Обозначим множество всех отображений в шкалу наименований символом *v*.

Получение на множестве *N* с помощью отображения *v* множеств N_H и R_{NH} позволяет на естественном для ЛФР языке получить наименование цели консультирования. На первом такте диалога имеет место следующее соотношение:

$$\nu: \bar{B}^1 = [\bar{H} + (\bar{H} + \bar{H}z)\mu]d = [\bar{H} + \bar{H}(1+z)\mu]d,$$

где $\bar{H}(1+z)\mu$ обозначает первый вопрос к ЛФР относительно цели консультирования *H* процессом $\bar{Y}$; B^1 — ответ *d* на запрос САК.

При построении модели цели консультирования $\bar{H}(1+z)\mu$, первый вопрос, который формулирует САК, ставится так, чтобы ответом на него было наименование цели консультирования N_{HB}, т. е.

$$N_{HB} = \bar{B}^1 = [\bar{H} + \bar{H}(1+z)\mu]d.$$

На втором такте на экране монитора формируется вопрос, позволяющий получить наименование компонент N_{HB}, т. е.

$$n_1, n_2, \ldots, n_k = \bar{B}^2 = [\bar{H} + (\bar{H} + B^1 Z)\mu]d,$$

$$n_i \in N_{HB}, \forall i = \overline{1, k}.$$

На третьем такте происходит дальнейшая декомпозиция структуры цели консультирования $\bar{H}_2$ на шкале наименований *N* с помощью вопросов $(\bar{H} + B^2 z)\mu$:

$$n_{11}, n_{12}, \ldots, n_{1q}, \ldots, n_{mk} = \bar{B}^3 = [\bar{H} + (\bar{H} + \bar{B}^2 z)\mu]d.$$

На (*k*+1)-м такте диалога получаем наименования целей *k*-го уровня

$$\bar{B}^{k+1} = [\bar{H} + (\bar{H} + \bar{B}^k z)\mu]d.$$

Декомпозиция цели консультирования $\bar{H}$ продолжается до тех пор, пока на очередной *t*-й вопрос множество наименований $\tilde{B}^l$ не станет пустым, т. е. ЛФР будет не в состоянии отобразить понятие в шкале наименований через два и более других наименования, не эквивалентных друг другу.

4.7.3. Цели консультирования и предпочтения ЛФР

Здесь рассмотрим, как учесть при построении моделей оценки рекомендаций предпочтения ЛФР, которые сказываются на его ответах.

На рис. 4.25 представлено дерево целей консультирования $\bar{H}$, декомпозиция которых осуществлена в интерактивном режиме.

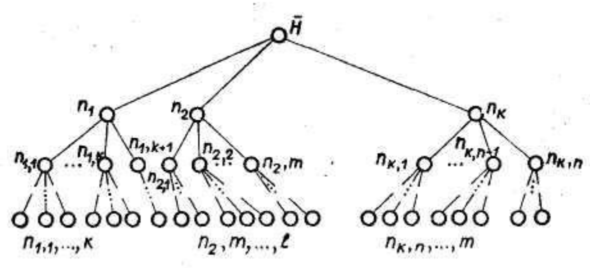

Рис. 4.25. Пример дерева целей консультирования, получаемого в интерактивном режиме.

Таким образом, система с отношениями $\bar{H}$ отображена на первом этапе с помощью v в систему с отношениями $(N_H,\ r^1)'$, которая, согласно определению 10 и приведенным теоремам 1—3, является гомоморфизмом системы $(\bar{H};\ R_H^i,\ i\in I)$. Ввиду того что условия определения не выполняются, гомоморфизм v:

$(H;\ R_H^i,\ i\in I)\rightarrow(N_H;\ r^l)$ не соответствует определению модели целей консультирования. Для того чтобы гомоморфизм v_1 :

$(H;\ R_H^i,\ i\in I)\rightarrow(N_H;\ r^l)$ удовлетворял критерию адекватности, предложенному в п. 4.6, отобразим отношение предпочтения $r^2\in R_H^i$ на систему с отношениями $(N_{HB},\ r^1)$.

Для двух заданных наименований n_k и n_d при выявлении отношения предпочтения возможны четыре случая:

1) $n_k r^2 n_d;$ 3) $n_d r^1 n_k;$

2) $n_d r^2 n_k;$ 4) $n_d r^0 n_k,$

где r^1 — отношение эквивалентности (безразличия); r^0 — отношение несравнимости двух элементов. Если отношение R условно изобразить в виде дуги ориентированного графа, то случаи 1—4 можно представить, как показано на рис. 4.26.

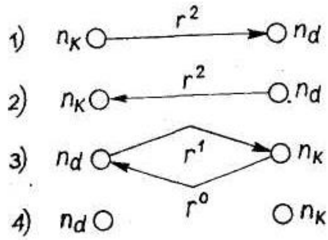

Рис. 4.26. Виды выявленного отношения предпочтения.

Неопределенность одного из четырех случаев отношения предпочтения может быть оценена в предположении равновероятности исходов

$$H = \log_2 m = \log_2 4 = 2 \text{ бит/отношение.}$$

Этот результат можно интерпретировать следующим образом. Для того чтобы выявить отношение предпочтения r^2, необходимы минимум два вопроса, ответы на которые несут по одному биту информации. Такие ответы ЛФР соответствуют словам «да» и «нет». Для случаев, представленных на рис. 4.26, первый вопрос будет, например, следующего содержания:. «Можете ли вы назвать из двух предлагаемых имен n_k и n_d наиболее предпочтительное для достижения цели $N_{НВ}$?» Возможные ответы: «да» и «нет». Если ЛФР отвечает «нет», то это соответствует случаям 3 и 4 на рис. 4.26. Второй вопрос может быть следующего содержания: «Имена n_k и n_d эквивалентны (равнозначны) для достижения цели $N_{НВ}$?» В результате положительного ответа может быть выявлено отношение $n_k r^1 n_d$, а в результате отрицательного — $n_k r^0 n_d$. Задавая в интерактивном режиме вопросы для выявления предпочтений ЛФР, получаем от него информацию, которая отображает отношение $R(n_k, n_d, k, t, N_{НВ})$ с учетом ситуации, цели консультирования $N_{НВ}$, действия помехи и относится к k-му шагу ЛФР. В настоящее время ЛФР в большинстве задач располагает такой информацией об отношениях на множестве $N_Н$, которая делает его наиболее компетентным экспертом по данному классу задач консультирования, учитывающим специфику отношений на системе $(H;\ R_H^i,\ i \in I)$. Поэтому экспертное оценивание ЛФР параметров решаемой задачи является единственно возможным для слабо структуризованных проблем, когда наименования и их оценки имеют в основном качественные параметры и шкалы. *Формализация информации, полученной от ЛФР, в некоторых случаях является единственным способом, позволяющим в дальнейшем применить* *математические методы для решения различных задач консультирования в САК.*

Известны *четыре шкалы*, позволяющие отобразить отношения R в систему с отношениями $(H;\ R_H^i,\ i\in I)$. Простейшей шкалой является шкала *наименований*, определение которой дано в п.4.4.7, с единственным заданным на ней отношением эквивалентности r^1. Шкала наименований используется только для классификации объектов. Система с отношениями $(N,\ r^1)$ будет в дальнейшем обозначать шкалу наименований.

Шкала *порядка* является следующим усилением измерения объектов $(H;\ R_H^i,\ i\in I)$, так как дополнительно к отношению эквивалентности r^1 выявляет отношение порядка r^2, т. е. использует отображение π, устанавливающее соответствие между шкалой наименований и шкалой порядка согласно определению 5:

$$\pi\colon (N,\, r^1)\to(N;\, r^1,\, r^2)=\Pi\,.$$

Шкала *интервалов* является отображением φ шкалы порядка П в метрическое пространство Ф с известным расстоянием $q(n_k,\ n_d)$ между любыми элементами $n\in N$

$$\varphi\colon \Pi\to\Phi=(N;\, r^1,\, r^2,\, q\,(n_k,\, n_d))\,,$$

причем известно

$$r^k\in R:\forall q\,(n_k,\ n_d)\to(r^1,\ r^k\in R),$$

т. е. известно отношение любых двух расстояний (интервалов) между элементами n_k, n_d по определению 7.

Согласно теореме 1, композиция

$$\nu \bigcirc \pi \bigcirc \varphi$$

позволяет для системы с отношениями $(H;\ R_H^i,\ i\in I)$ получить гомоморфизм $(N_H,\ r^1,\ r^2\ q(n_k,\ n_d),\ \varepsilon)$, адекватный отношению порядка ε на шкале интервалов Ф, шкале порядка П и шкале наименований N. Иначе говоря, можно получить модель целей консультирования согласно определению 9, которая отображает систему с изменяемыми отношениями $(H;\ R_H^i,\ i\in I)$ в метрическое пространство с определенными отношениями. Предлагается несколько методов получения от ЛФР информации об отношении r^2 на множестве H, которые могут быть использованы при реализации интерактивного режима. Ранее было показано, как на l-м такте диалога ЛФР с САК в моделях отображается дерево целей в шкале наименований $(N_H,\ r^1)$ с дальнейшим отображением отношения $r^2\in R_H$ в шкалу порядка П:

$$\pi\colon (N_H;\, r^1)\to(N_H;\, r^1,\, r^2)=\Pi\,.$$

Для этого на (t+1)-м такте диалога на экране монитора появляются вопросы, позволяющие определить отношение r^2 на первом уровне дерева целей, т. е.

$$\bar{B}^{t+1} = [\bar{H} + (\bar{H} + (\bar{H}; r^1)^t z)\mu]d,$$

где B^{i+1} — ответ ЛФР на запрос САК μ, требующий нахождения $n_0 \in N_H$, удовлетворяющего следующему отношению:

$$n_0 r^2 n_1 \wedge n_0 r^2 n_2 \wedge \ldots \wedge n_0 r^2 n_{(k-1)} = \bigwedge_{i=1}^{k-1} n_0 r^2 n_i .$$

На (t + 2)-м такте диалога элемент n_0 исключается из множества

$$N_H^{l+2} \subset N_H^{l+1}, \quad n_0 \notin N_H^{l+2}$$

и определяется такой элемент $n_0^1 \in N_H^{l+2}$, для которого

$$\bar{B}^{t+2} = [\bar{H} + (\bar{H} + (H; r^1, r^2)^t z)\mu]d,$$

удовлетворяющий следующему отношению r^2:

$$n_0^1 r^2 n_1 \wedge n_0^1 r^2 n_2 \wedge \ldots \wedge n_0^1 r^2 n_{(k-2)} = \bigwedge_{i=1}^{k-2} n_0^1 r^2 n_i .$$

Число тактов для выявления отношения r^2 на первом уровне дерева целей консультирования $\bar{H} = (H, R_H)$ находится в интервале от 1 до $(n^2 — n)/2$, где n — число элементов на первом уровне.
Общее число тактов диалога для построения модели оценки степени достижения цели определяется выражением

$$L = \sum_{i=1}^{n} \sum_{j=1}^{k} \frac{n_{ij}(n_{ij}-1)}{2} K,$$

где j=1, 2,…, K — число уровней декомпозиции, а i=1,2,…, n — число элементов на одном уровне. Исследование различных отображений $\pi \in \Pi$ для шкалы порядка рассмотрено в п. 4.7.8.

4.7.4. Примеры интерактивной процедуры построения модели цели консультирования

Интерактивные процедуры в их практической реализации представляют собой последовательность выведенных на видеотерминальное устройство вопросов и введенных с него ответов ЛФР. Приведем один из возможных вариантов такого диалога между ЛФР и САК.

Вопросы, выведенные на видеотерминальное устройство, отмечены символом *.

* СФОРМУЛИРУЙТЕ И ВВЕДИТЕ НАИМЕНОВАНИЕ ВАШЕЙ ПРОБЛЕМЫ (СИТУАЦИЯ ВЫБОРА)

— Н.

* ВВЕДИТЕ ФАКТОРЫ, ХАРАКТЕРИЗУЮЩИЕ (Н) БОЛЕЕ ПОДРОБНО, РАЗДЕЛИВ ИХ ЗАПЯТОЙ.

— Г, М, Е.

* МОЖНО ЛИ ФАКТОР (Г) ХАРАКТЕРИЗОВАТЬ БОЛЕЕ ПОДРОБНО ЧЕРЕЗ ДРУГИЕ (ДА, НЕТ)?

— ДА. .

* ВВЕДИТЕ ФАКТОРЫ, ХАРАКТЕРИЗУЮЩИЕ (Г), РАЗДЕЛИВ ЗАПЯТОЙ.

— А, В.

* МОЖНО ЛИ ФАКТОР (М) ХАРАКТЕРИЗОВАТЬ БОЛЕЕ ПОДРОБНО ЧЕРЕЗ ДРУГИЕ (ДА, НЕТ)?

— ДА.

* ВВЕДИТЕ ФАКТОРЫ, ХАРАКТЕРИЗУЮЩИЕ (М), РАЗДЕЛИВ ИХ ЗАПЯТОЙ.

— С.

* МОЖНО ЛИ ФАКТОР (Е) ХАРАКТЕРИЗОВАТЬ БОЛЕЕ ПОДРОБНО ЧЕРЕЗ ДРУГИЕ (ДА, НЕТ)?

— НЕТ.

* МОЖЕТЕ ЛИ ВЫ ИЗ ФАКТОРОВ (А, В, С, Д) ВВЕСТИ НАИМЕНОВАНИЯ ТЕХ, КОТОРЫЕ МОЖНО ВЫРАЗИТЬ ЧЕРЕЗ ДРУГИЕ ФАКТОРЫ?

— НЕТ.

* ВВЕДИТЕ ФАКТОРЫ ИЗ СПИСКА А, В, С, Д, Е, ЗНАЧЕНИЯ КОТОРЫХ НАДО МИНИМИЗИРОВАТЬ ПРИ ДОСТИЖЕНИИ ЦЕЛИ Н.

— Е.

* ВВЕДИТЕ НАИМЕНОВАНИЕ ФАКТОРА, НАИБОЛЕЕ ВАЖНОГО ПРИ ДОСТИЖЕНИИ ЦЕЛИ Н ИЗ СПИСКА Г, М, Е.

— Г.

* ВВЕДИТЕ НАИМЕНОВАНИЕ ФАКТОРА, НАИБОЛЕЕ ВАЖНОГО ПРИ ДОСТИЖЕНИИ ЦЕЛИ Н ИЗ СПИСКА М, Е.

— М.

* ВВЕДИТЕ НАИМЕНОВАНИЕ ФАКТОРА, НАИБОЛЕЕ ВАЖНОГО ПРИ ДОСТИЖЕНИИ ЦЕЛИ Н ИЗ СПИСКА А, В.

— А.

* ВВЕДИТЕ НАИМЕНОВАНИЕ ФАКТОРА, НАИБОЛЕЕ ВАЖНОГО ПРИ ДОСТИЖЕНИИ ЦЕЛИ Н ИЗ СПИСКА С, Д.

— С.

* ПРЕДЛАГАЕМЫЙ СПИСОК СОСТОЯНИЙ УПОРЯДОЧИТЕ В СООТВЕТСТВИИ СО СТЕПЕНЬЮ ДОСТИЖЕНИЯ ЦЕЛИ Н:

К1 = (Д); К2 = (А, В); К3=(А, С); К4= (А, Д); К5=(А, С, Д);

К6= (В, С, Е); К7 =(В, Д, Е); К8=(В, С, Д, Е).

— К5, К2, К3, К8, К4, К6, К7, К1.

В результате интерактивных процедур моделью формализуется дерево целей консультирования $H = (H;\ R_H^i,\ i \in I)$. В табл. 4.19 представлены факторы, характеризующие ряд состояний при определении весовых коэффициентов вершин различных уровней дерева целей, где 1 обозначает максимальную оценку по данному показателю; 0 — минимальную.

Таблица 4.19

Пример характеристики состояний консультируемой проблемы

Состояние	Факторы				
	A	*B*	*C*	*D*	*E*
K_1	0	0	0	1	0
K_2	1	1	0	0	0
K_3	1	0	1	0	0
K_4	1	0	0	1	0
K_5	1	0	1	1	0
K_6	0	1	1	0	1
K_7	0	1	0	1	1
K_8	0	1	1	1	1

4.7.5. Основные положения теории полезности

Теория полезности делает возможным отображение информации о предпочтениях ЛФР в числовую информацию, используемую для оценки полезностей (предпочтений) реальных альтернатив, которые в дальнейшем преобразуются в суждения о предпочтениях на множестве последних.

Одним из основных понятий теории полезности является понятие отношения предпочтения.

Отношением предпочтения $r^2 \in R$ на множестве элементов Y называется бинарное рефлексивное отношение, такое, что yr^2z интерпретируется как «у не хуже (не менее предпочтительно), чем z».

Условие рефлексивности этого отношения можно записать в виде $\forall\, y\in Y{:}yr^2y$. Симметричную часть этого отношения

$$r^1\{yr^1z\leftrightarrow yr^1z\wedge zr^1y\}$$

называют *отношением неразличимости или безразличия*. Асимметричную часть отношения r^2 обозначим r^3 и назовем *отношением строгого предпочтения*:

$$r^3\{yr^3z\leftrightarrow yr^3z,\ \bar{z}r^3y,\ z,\ y\in Y\},$$

причем yr^3z читается так: «y строго (заведомо, наверняка) лучше (предпочитается) z».

Отношение несравнимости r^0 удовлетворяет следующему условию:

$$r^0\{yr^0z\leftrightarrow[(\urcorner\, yr^2z)\wedge(\urcorner\, zr^2y)]\},\ y,\ z\in Y.$$

Можно отметить, что $r^1\bigcup r^3=r^2$.

Отношением строгой неразличимости (эквивалентности), порожденным отношением предпочтения r^2, называется бинарное отношение r^1, удовлетворяющее следующим условиям:

$$\text{а)}\ \ (yr^1z\leftrightarrow zr^1y),\quad z,\ y\in Y;$$

$$\text{б)}\ \ (yr^2z\wedge yr^1\omega\wedge zr^1k)\leftrightarrow\omega r^2k.$$

Обозначим через $\overline{Y}=Y/r^1$ фактор-множество Y по отношению эквивалентности r^1 $[y_i]_r{}^1$, где $i\in I$ — множество элементов Y, эквивалентных y_i по отношению r^1.

Можно отметить, что на классы эквивалентности фактор-множества $\overline{Y}=Y/r^1$ переносится отношение предпочтения между его элементами r^2, т. е.

$$[y_a]_{r^1}\bar{r}^2[y_k]_{r^1}\leftrightarrow(\{\forall y_n\in[y_a]_{r^1},\ y_q\in[y_k]_{r^1}:$$

$$:y_nr^2y_q\}\rightarrow y_ar^2y_k);\quad a,\ k,\ n,\ q\in I,$$

где r^2 называют отношением, индуцированным r^2 на Y/r^1.

В случае, когда множество $\overline{Y}$ не более чем счетно, а отношение r^3 транзитивно, т. е.

$$(y_ar^3y_k)\wedge(y_kr^3y^n)\rightarrow y_ar^3y_n,$$

существуют такие вещественные функции $U{:}\ Y\rightarrow N$ и положительные функции $U{:}\ Y\rightarrow N^+\in\Phi$, что

$$\forall y_a,\ y_k\in Y{:}\,y_ar^3y_k\rightarrow U(y_a)>U(y_k).$$

Если отношение r^3 — полупорядок, т. е.

$$y_a r^3 y_k \wedge y_k r^1 y_n \wedge y_n r^3 y_q$$

и

$$y_a r^3 y_k \wedge y_k r^3 y_n \rightarrow \{y_n, y_q \in Y: y_q r^1 y_a, y_q r^1 y_n\} = \varnothing ,$$

то существует такая вещественная функция $U: Y \rightarrow N$ и положительная функция $v:Y \rightarrow N$, где $N^+ = \{a{:}a \in N, a>0\}$, что

$$\forall y_a, y_k \in Y: y_a r^3 y_k \leftrightarrow u(y_a) > u(y_k) + v(y_a).$$

Когда отношение r^2 транзитивно, симметрично и рефлексивно, т. е.

$$y_a r^2 y_k \wedge y_k r^2 y_n \rightarrow y_a r^2 y_n ;$$

$$y_a, y_k, y_n \in Y; \quad y_a r^2 y_k \leftrightarrow y_k r^2 y_a ,$$

существует такая вещественная функция u: $Y \rightarrow N$, что

$$\forall y_a, y_k \in Y: y_a r^2 y_k \leftrightarrow u(y_a) \geqslant u(y_k).$$

Одним из наиболее важных в теории полезности является понятие независимости, или аддитивности, факторов, которое широко используется для решения многокритериальных задач.

Если $s = (y_1,..., y_n)$, то, в соответствии с теорией аддитивной полезности,

$$u(s) = u(y_1, \ldots, y_n) = \sum_{i=1}^{n} u_i(y_i) .$$

В некоторых случаях это выражение удобно представить в виде

$$u(s) = u(y_1, \ldots, y_n) = \sum_{i=1}^{n} v_i w_i(y_i), \tag{4.72}$$

где

$$v_i > 0,\ \Sigma_i v_i = 1,\ \max w_i(y_i) = 1,\ \min w_i(y_i) = 0.$$

Полученное выражение (4.72) нередко помогает разрешать проблемы многокритериальности и разнородности единиц измерения показателей.

Переменную v_i иногда называют весом, важностью, весовым коэффициентом, степенью важности и т. п., а $w_i(y_i)$ — значением (числовым) функции полезности для оценки y_i показателя.

В случае, когда $\forall a_i, a_j \in A$,

$$(p(y_1)_i, p(y_2)_i, \ldots, p(y_n)_i)$$

и

$$(p(y_1)_j, p(y_2)_j, \ldots, p(y_n)_j) : \Sigma_i p(y_k)_i = \Sigma_m p(y_m)_j ,$$

где $p(y_k)_i$ — числовая оценка уверенности ЛФР в условиях неопределенности в том, что при выборе альтернативы a_i процесс на выходе будет иметь максимальное значение y_k показателя $p(y_m)_j$, что соответствует числовой оценке j-й альтернативы. Величины $p(y_k)_i$ и $p(y_m)_j$ называют *субъективными вероятностями*. В этом случае выражение для оценки альтернатив можно представить в следующем виде:

$$u(a_j)=u(y_1, p(y_1)_j, \ldots, y_n, p(y_n)_j)=\sum_{i=1}^{n} p(y_i)v_i w_i(y_i).$$

Следует отметить важное свойство отношений предпочтения. Бинарное отношение предпочтения есть сокращенная запись четырехместного отношения

$$r^2(y_i; y_j, k, t),$$

где y_i и y_j — объекты сравнения, k — индивидуум, у которого выявлено это отношение, а t — момент времени, к которому это отношение относится. Тогда последнее выражение может быть прочитано так:

«k-й индивидуум в момент времени t не предпочитает y_i элементу y_j». В связи с этим возникает вопрос о стационарности предпочтений, т. е. о том, насколько они изменяются во времени. Однако и при фиксированных k и t, y_i и y_j отношение r^2 может измениться. Это имеет место, если бинарное отношение предпочтения r^2 есть пятиместное отношение $r^2(y_i, y_j, k, t, G_j)$, где первые четыре элемента означают то же, что и в предыдущем случае, а G_j означает j-ю цель, достигаемую выбором y_i или y_j. Отсюда следует, что отношение предпочтения при некоторых условиях может оказаться нестационарным и модели САК должны своевременно учитывать его изменение. При этом отношение предпочтения позволяет представить в числовой форме отображение реальных отношений на моделируемых процессах САК.

4.7.6. Методы оценки аддитивных ценностей (рекомендаций) и построения функции полезности рекомендаций

Здесь рассматриваются методы, позволяющие численно оценить предпочтения ЛФР. Приведенные примеры и числовые значения имеют чисто иллюстративный характер.

Метод ранжировки используется при оценке дискретных факторов и основан на суждениях ЛФР о предпочтении или на «прямых» суждениях о неравенстве ценностей (рекомендаций).

Уровни оцениваемого фактора ранжируются от наименее предпочтительного до наиболее предпочтительного или наоборот. Получить ранжировку можно многими способами, которые основаны на парных сравнениях. При этом можно использовать суждения как о безразличии (r^0), так и о строгом предпочтении (r^2). Информацию относительно числовых значений ценностей для разных уровней получают на основе ранжировки, например:

$$u\ (y_i) < u\ (y_k) <...< u\ (y_l).$$

Прежде чем приписать определенные числовые значения ценностям уровней, желательно получить дополнительную информацию о предпочтениях ЛФР. Пусть дискретный фактор — число реализованных рекомендаций, сформированных консультантом (ЛФР,ом) консультационного центра за последний год. Для консультанта, который реализовал, например, семь рекомендаций, y_i= 7, для консультанта, который реализовал только одну рекомендацию, y_k =l, если консультант не реализовал ни одной рекомендации, то y_i = 0.

Допустим, что чем больше рекомендаций реализует консультант консультационного центра, тем выше оценивается эффективность его консультационной работы, т. е.

$$u(0)<u(1)<...<u(7).$$

ЛФР, исходя из своего опыта и целей, может присвоить следующие числовые значения оценкам для каждого уровня (табл. 4.20).

Таблица 4.20

Пример оценки уровней

Уровень фактора	Полезность
7	100
6	99
5	88
4	60
3	40
2	30
1	20
0	0

На рис. 4.27 показан один из вариантов графика функции полезности для рассматриваемого примера.

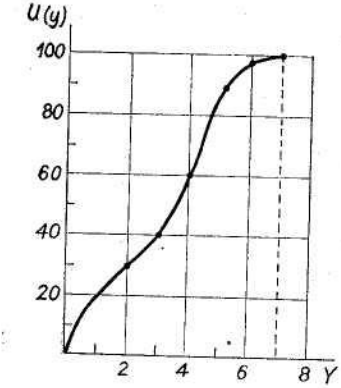

Рис. 4.27. График функции полезности, полученный методом ранжировки.

Метод стандартной игры. Этот метод, используемый для шкалирования ценностей вероятности, основан на суждениях о безразличии для оценки одного фактора, как в случае его непрерывных, так и дискретных значений.

Методом ранжировки определяют наиболее желательный уровень y_1 и наименее желательный уровень y_i. Затем для каждого уровня из совокупности других уровней $\{y_k, ..., y_l\}$ определяется вероятность $p(y_k)$, при которой y_k безразлично к распределению вероятности $p(y_k)$ для y_l и вероятности $1 — p(y_k)$ для y_l. В этом случае

$$u(y_k)=p(y_k)u(y_l) + [1-p(y_k)]u(y_l).$$

Если фактор дискретен, это можно сделать для каждого уровня. Если он непрерывен, можно воспользоваться выборкой оценок для получения предварительного вида кривой $u(y_k)$.

Пусть, например, $y_l = 5$, $y_k = 4$, $y_i = 3$ есть средние баллы трех выпускников института. Определим значения полезности для ЛФР — руководителя отдела консультационной организации, который имеет возможность выбрать, исходя из своих предпочтений и опыта, одного из выпускников и принять его на работу.

Назначим базовую вероятность $p(y_k)$ =0,9. Предлагаем ЛФР либо сразу получить выпускника со средним баллом $y_k = 4$, либо участвовать в следующей лотерее. В урну кладут 9 белых шаров и 1 черный. Если ЛФР вынимает белый шар, то получает выпускника со средним баллом $y_l = 5$, если черный шар — выпускника со средним баллом $y_i = 3$. Пусть ЛФР согласилось на участие в лотерее. (Разумеется, лотерею можно проводить и мысленно.) Это означает, что

$$u(y_k = 4) < 0{,}9u(y_k = 5) + (1\text{-}0{,}9)u(y_k = 3).$$

В методе стандартной игры необходимо стремиться к состоянию, когда ЛФР будет безразлично к выбору или сразу среднего по предпочтительности уровня или участия в лотерее с наиболее и наименее предпочтительными уровнями. В рассматриваемом примере для этого необходимо уменьшить базовую вероятность $p(y_k)$.

Пусть при $p(y_k)$=0,8 ЛФР отказалось участвовать в описанной лотерее и согласилось сразу получить выпускника со средним баллом $y_k = 4$. Это означает, что

$$u(y_k = 4) < 0{,}8u(y_k = 5) + (1\text{-}0{,}8)u(y_k = 3).$$

Пусть, наконец, при $p(y_k) = 0{,}85$ ЛФР высказалось, что ему безразлично, получить ли сразу выпускника со средним баллом $y_k = 4$ или участвовать в лотерее. В этом случае

$$u(y_k = 4) < 0{,}85u(y_k = 5) + (1\text{-}0{,}85)u(y_k = 3).$$

Допустим, что $u(y_k = 5) = 1$, а $u(y_k = 3) = 0$. Тогда

$$u(y_k = 4) = 0{,}85\cdot 1 + 0{,}15\cdot 0 = 0{,}85.$$

Если наименее предпочтительный уровень $u(y_k = 3) \neq 0$, а $u(y_k = 3) = 0{,}4$, то

$$u'(y_k = 4) = 0{,}85\cdot 1 + 0{,}15\cdot 0{,}4 = 0{,}91.$$

Графики функций полезности $u(y)$ и $u'(y)$ для рассматриваемого примера показаны на рис. 4.28.

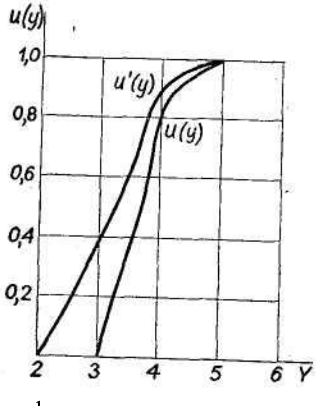

Рис. 4.28. График функции полезности, полученный методом стандартной игры

Метод вероятностной оценки основан на суждении о безразличии альтернатив при оценке весов двухуровневых факторов или параметров преобразования шкал.

Пусть фактор y^n предпочтительнее, чем фактор y^i. Необходимо получить такое r^i, при котором безразличны следующие две альтернативы:

1) получить лучший уровень фактора *i* и худший уровень фактора *n;*

2) получить худший уровень фактора *i* и лучший уровень *n* с вероятностью p_i или худший уровень фактора *i* и худший — фактора *n* с вероятностью $1—p_i$.

На рис. 4.29 показан пример структуры альтернатив для иллюстрации рассматриваемого метода.

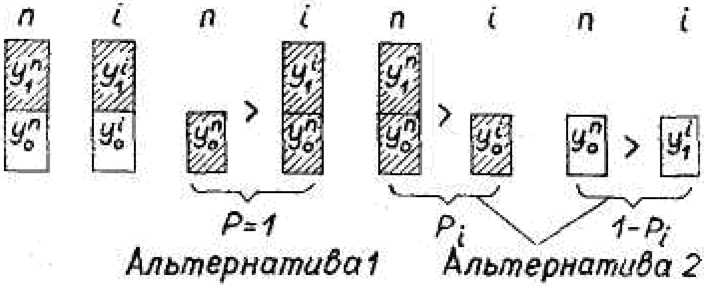

Рис. 4.29. Схема формирования альтернатив при использовании метода вероятностной оценки.

При этом фактор y^n предпочтительнее фактора y^i т. е. $y^n > y^i$. Далее, уровень y_1^n предпочтительнее уровня y_0^n для фактора y^n, а y_1^i предпочтительнее y_0^i для фактора y^1. Если будет выполнено условие, при котором имеет место безразличие между первой и второй альтернативами, то

$$u(y^i)=p_i u(y^n)$$

при условии, что выполняется

$$u(y^1, y^2, ..., y^n)=u(y^1)\, y^1+ + u(y^2)\, y^2+ ... +u(y^n)y^n,$$

где $(y^1, y^2, ..., y^n)$, $y^i \in \{0,1\}$, $\forall\, i = 1, 2,\ldots, n$. Необходимо также отметить, что в теории ожидаемой полезности Неймана—Моргенштерна отношение предпочтения r^2 применяется к множеству ρ простых распределений вероятностей на множестве *Y*. Согласно этой теории, если *S* и Ф — любые два распределения в ρ, то существует числовая функция ценности на ρ такая, что Ф ‰ *S*, если и только если $u(Ф) \geq u(S)$ и если $s(y^1)+s(y^2) + ...+s(y^n) =1$, где $y^i \in Y$, $i=, 2,\ldots, n$, а знак ‰ читается «не менее предпочтительно, чем» и обозначает отношение предпочтения ЛФР на множестве ρ.

Этим условиям удовлетворяет, например, функция

$$u(S)=s(y^1)u(y^1)+s(y^2)u(y^2)+ \dots +s(y^n)u(y^n),$$

где $u(y^1)=u(\Phi)$, когда $\Phi(y^1) = 1$.

Пусть требуется определить оценки факторов ЛФР, формирующего рекомендации по покупке цветного телевизора. Известно, что возможны два дефекта: y^n — во внешнем виде телевизора и y^i— в цветопередаче изображения. При этом y^n‰ y^i, т. е. телевизор с дефектом во внешнем виде предпочитается телевизору с дефектом в цветопередаче. В этом случае для оценивания факторов предлагаются альтернативы со структурой, представленной в табл. 4.21.

Таблица 4.21

Пример структуры альтернатив

Область проявления дефекта	Обозначения дефекта	Альтернативы					
		Первая $p=1$		Вторая			
				$p'=p_i$		$p''=1-p_i$	
Внешний вид	y^n	Нет	1	Есть	0	Есть	0
Цветопередача	y^i	Есть	0	Нет	1	Есть	0

В данной ситуации первый вопрос ЛФР может быть таким: «Вы рекомендуете сразу купить телевизор *В* с дефектом в цветопередаче и без дефектов во внешнем виде или рекомендуете участвовать в лотерее? Условия лотереи следующие. В урне лежат 10 белых шаров и 90 черных. Если вы достанете белый шар, то сможете купить телевизор *С* без дефекта в цветопередаче, но с дефектом во внешнем виде и только этот телевизор. Если вы достанете черный шар, то сможете купить телевизор *D,* имеющий дефекты и в цветопередаче, и во внешнем виде, т. е. хуже, чем телевизор *В.* Итак, рекомендуете брать телевизор *В* или рекомендуете вынимать шар?» Если ЛФР рекомендует купить сразу, без лотереи, телевизор *В,* то вопрос повторяют, причем условия лотереи изменяются: увеличивается вероятность приобретения телевизора *С.* Так поступают до тех пор, пока ЛФР не порекомендует лотерею вместо гарантированной покупки. Предположим, что если в урне окажется 20 белых и 80 черных шаров, то ЛФР будет безразлично к выбору между лотереей и гарантированной покупкой. В этом случае можно записать

$$u(y^i)=0{,}20u(y^n).$$

Пусть y^n =100 условным единицам ценности, тогда y^i = 20. Ценность телевизора без дефектов

$$u\,(y^n, y_1^i)= 1\,u(y_1^n) + 1\,u\,(y_1^i) = 1\cdot 100+1\cdot 20 = 120 \text{ (усл. ед.).}$$

Ценность телевизора с дефектом во внешнем виде

$$u(y_1^n, y_0^i) = 1\,u(y_1^n) + 0\,u\,(y_1^i) = 1\cdot 100+0\cdot 20 = 100 \text{ (усл. ед.).}$$

Ценность телевизора с дефектом в цветопередаче

$$u(y_0^n, y_1^i) = 0\,u(y_1^n) + 1\,u\,(y_1^i) = 0\cdot 100+1\cdot 20 = 20 \text{ (усл. ед.).}$$

Ценность телевизора с дефектами и в цветопередаче, и во внешнем виде

$$u(y_0^n, y_0^i) = 0\,u(y_1^n) + 0\,u\,(y_1^i) = 0\cdot 100+0\cdot 20 = 0$$

Условие аддитивности факторов проверяется следующей лотереей. Пусть, например, фактор y^n предпочтительнее y^i. ЛФР рекомендует сделать выбор между альтернативами, одна из которых предполагает лучший уровень фактора y^n и худший — фактора y^i, а другая — худший уровень фактора y^n и лучший уровень y^i. Если предпочитается первая альтернатива, то факторы y^n и y^i аддитивны. Если нет, то эти факторы необходимо либо объединить в один сложный фактор, либо исключить из рассмотрения. В рассматриваемом примере можно считать, что в ответ на предложение выбрать или телевизор с хорошей цветопередачей и дефектом во внешнем оформлении, или телевизор с дефектом в цветопередаче, но без дефекта во внешнем оформлении ЛФР рекомендует выбрать телевизор с хорошей цветопередачей. Это значит, что факторы y^n и y^i аддитивны и возможно выполнение арифметических операций по формуле (4.72).

Существует еще ряд методов определения ценности альтернатив, в которых используются элементы теории полезности. Воспользуемся положениями параграфов 4.7.5 и 4.7.6 для построения моделей оценки степени достижения целей.

4.7.7. Оценка различных структур и параметров модели

Для определения степени адекватности композиции $v\mathrm{O}\pi$, $v\in \mathrm{B}$, $\pi\in \Pi$ системы с отношениями $(H;\ R_H^i,\ i\in I)$ воспользуемся предложенной в п. 4.7.5 методикой вычисления значения коэффициентов $\rho(\varepsilon\ (Y_0\lambda))$ при фиксированном $\lambda_0\in\Lambda$. Один из известных методов позволяет сравнить векторы $n^k(n_1^k,\ n_2^2,\ \ldots,n_m^k)$, для которых n_q^k принимают значения из множества $\{0,1\}$. Для сравнения нескольких структур отобразим модель дерева целей консультирования в шкале интервалов на систему с отношениями

$(N_H;\ r^1,\ r^2,\ q(n_л,\ n_m))$ с помощью процедур λ. Одним из гомоморфизмов λ_n будет подмножество натуральных чисел N^+ с отношениями r^1, r^2, причем $\lambda(r^1)$ есть « = », а $\lambda(\ r^2)$ есть «$\geq$», т. е.

$$(N^+;\ \geqslant,\ =) = (N^+;\ r^2,\ r^1).$$

После L-го такта диалога комплекс математических моделей z формирует на экране дисплея вопрос о выявлении отношения r^2 на контрольном примере

$$\bar{B}^{L+1} = [\bar{H} + (\bar{H} + B^L z)\mu]d,$$

где $(\bar{H} + B^L z)\mu$ — информация, введенная в САК относительно системы $(H;\ R_H^i,\ i \in I)$. Отображение λ множества бинарных векторов $n^k \in N^k$ в числовую систему с отношениями $(N^+;\ r^1,\ r^2)$ осуществляется, например, с помощью свертки для аддитивной функции полезности

$$v_j(n_1, n_2, \ldots, n_k) = \sum_{i=1}^{k} \alpha_i n_{ij}, \tag{4.73}$$

где $\alpha_i \in N^+$, причем $\alpha_i > \alpha_p \leftrightarrow n_i r^2 n_p$.

Следовательно, формула (4.73) отображает на множестве V отношение порядка r^2 из системы $(N^+;\ r^1,\ r^2)$ тогда и только тогда, когда

$$v_k r_v^2 v_a \leftrightarrow \sum_{i=1}^{n} (\alpha_i n_i)_k > \sum_{j=1}^{n} (\alpha_j n_j) d, \quad n_i, n_j \in \{0, 1\}.$$

Отношение порядка r_v^2 на множестве V сравнивается с отношением порядка, предложенным ЛФР в результате выполнения L тактов диалога:

$$\bar{B}^{L+1} = [\bar{H}, \bar{Y} + (\bar{H}, \bar{Y} + B^2 z)\mu]d.$$

При этом для сравнения порядков используется выражение для определения коэффициента ранговой корреляции

$$\rho(\varepsilon(v)) = 1 - \frac{6(\Sigma_i (n_i^+ - n_i^+(d))^2 + T + T_d)}{k(k^2 - 1)}, \tag{4.74}$$

где n_i^+ — натуральное число (ранг), присвоенное на множестве V отображением λ_{NH} i-й структуре; $n_i^+(d)$ — отображение λ_H порядка i-й структуры по достижению цели $\bar{H}$; k — число сравниваемых структур; T и T_d определяются по формуле (4.70).

При заданном ЛФР множестве N_{OHB} и отношениях r^1 и r^2 на

$n_{0k} \in N_{OHB},\ k = 1, 2, \ldots, n$ (где n_{0k} — элементы подмножества

$n_{0k} \in N_{OHB} \subset N_{HB}$, образующие все конечные вершины дерева цели консультирования) возможно 2^n различных структур модели. Обозначим i-ю структуру модели дерева целей H через N_{HB}^i . В этом случае получается множество оценок

$$\rho_i(\varepsilon(N_{HB}{}^i)),\ i \in I_H = \overline{1,\ 2^n},$$

каждая из которых показывает на шкале интервалов степень адекватности отношению ε (порядка) относительно системы

(H; N_H^i, $i \in I$). Под оптимальной структурой модели дерева целей будем понимать такую N_{HB}, на которой достигается максимальное значение коэффициента степени адекватности отношению ρ;*

$$\rho(\varepsilon\ (N_{HB}^*)) > \rho(\varepsilon\ (N_{HB}^i)),\ i \in I_H\ .$$

В случае, когда подмножество $N_{OHB} \subset N_{HB}$ упорядочено, определяется вектор приоритета элементов дерева целей консультирования.

Пусть элементы $n_{01}, n_{02}, \ldots, n_{0n} \in N_{OHB}$ и упорядочены так, что $n_{01} > n_{02} > \ldots > n_{0n}$, где n_{01} — наиболее, а n_{0n} — наименее предпочтительный элемент для достижения цели $\bar{H}$. В качестве первой структуры дерева целей принимаем $\alpha_1 = 1$, $\alpha_2 = \ldots = \alpha_n = 0$, где $\alpha_1 = 1$ означает, что элемент n_1 включен в структуру модели, а

$\alpha_2 = \ldots = \alpha_n = 0$ — что элементы $n_2, \ldots, n_n$ не включены, т. е. модель дерева целей состоит из одного элемента. Вторая структура дерева целей будет $\alpha_1 = \alpha_2 = 1$, $\alpha_3 = \ldots = \alpha_n = 0$. Наконец, последняя структура

$$(S = 2^n):\ \alpha_1 = \alpha_2 = \ldots = \alpha_n = 1.$$

Обозначим первую структуру $N_{HB}^{(1)}$, вторую — $N_{HB}^{(2)}$ и т. д. Для каждой из структур $N_{HB}^{(1)}, \ldots, N_{HB}^{(2)}$ определяются коэффициенты степени адекватности отношению порядка ε относительно системы

(H; R_H^i $i \in I$):

$$\rho_1(\varepsilon(N_{HB}{}^{(1)})),\ \ \rho_2(\varepsilon(N_{HB}{}^{(2)})),\ \ldots,\ \rho_S(\varepsilon(N_{HB}{}^{(s)})),$$

которые численно показывают относительное влияние элемента n_{01} структуры на оптимальность, если сравниваются элементы

$n_{01}, \ldots, n_{(0i-1)}$ с элементами $n_{01}, \ldots, n_{0i}$ структуры модели.

В этом случае, отображая на множество N=[0,1] отношение порядка

$$\rho_1(\varepsilon(N_{HB}{}^{(1)})) : \rho_2(\varepsilon(N_{HB}{}^{(2)})) : \ldots : \rho_S(\varepsilon(N_{HB}{}^{(s)})),$$

получают такие значения $\alpha'_1, \alpha'_2, \ldots, \alpha_k'$, при которых выполняется условие

$$\alpha'_1 : \alpha'_2 : \ldots : \alpha'_n = \rho_1(\varepsilon(N_{HB}{}^{(1)})) : \rho_2(\varepsilon(N_{HB}{}^{(2)})) : \ldots$$

$$\ldots : \rho_n(\varepsilon(N_{HB}{}^{(n)})),\ \Sigma_i \alpha_i = 1.$$

Если $\rho_i(\varepsilon\ (N_{HB}^i)$ принимает отрицательное значение, то элемент n_{0i} исключается из структуры рассматриваемой модели целей консультирования. Предлагаемый способ определения вектора приоритета для элементов дерева целей позволяет решать задачу адаптации структуры модели к реальной системе в автоматическом режиме.

4.7.8. Пример оценки параметров модели

Воспользуемся результатами построения модели целей консультирования, изложенными в п. 4.7.4.

Для приведения исходных данных задачи к единым условиям получения максимальных значений показателей инвертируем оценки фактора *E* из табл. 4.19, т. е. вместо 0 напишем 1, и наоборот. В табл. 4.22 приведены значения коэффициента ранговой корреляции Спирмена для различных структур модели цели консультирования, вычисленные по формуле (4.74).

Таблица 4.22

Пример расчетных данных степени адекватности для различных структур модели

Состояния	Факторы					Веса факторов					M_1 $A+E$		M_2 $A+D$		M_3 $A+C$		M_4 $A+B$		M_5 A		Ранги, присвоенные ЛПР, n_d^+
	A	B	C	D	E	A	B	C	D	E	v_k	n_{M1}^+	v_k	n_{M2}^+	v_k	n_{M3}^+	v_k	n_{M4}^+	v_k	n_{M5}^+	
K_5	1	0	1	1	1	5	0	3	2	1	11	1	10	1	8	2,5	5	3	5	2,5	1
K_2	1	1	0	0	1	5	4	0	0	1	10	2	9	2,5	9	1	9	1	5	2,5	2
K_3	1	0	1	0	1	5	0	3	0	1	9	3,5	9	4	8	2,5	5	3	5	2,5	3
K_8	0	1	1	1	0	0	4	3	2	0	9	3,5	8	2,5	7	4,5	4	6	5	2,5	4
K_4	1	0	0	1	1	5	0	0	2	1	8	5	7	5,5	5	6	5	3	5	2,5	5
K_6	0	1	1	1	0	0	4	3	0	0	7	6	7	5,6	7	4,5	4	6	0	6,5	6
K_7	0	1	0	1	0	0	4	0	2	0	6	7	6	7	4	6	4	6	0	6,5	7
K·	0	0	0	1	1	0	0	0	2	1	3	8	2	8	0	8	0	8	0	6,5	8
Коэффициент										$\rho=$	0,95		0,94		0,87		0,86		0,76		1
Уровень значимости по критерию Стьюдента										$\alpha=$	0,001		0.001		0,005		0,005		0,025		—

Таблица 4.23

Оценка степени влияния факторов на адекватность модели

Коэффициент ранговой корреляции	Исключенные из модели факторы				
	A	*B*	*C*	*D*	*E*
$\rho(\varepsilon)$	0,76	0,87	0,73	0,90	0,94
$\rho(\varepsilon)_{\max}-\rho(\varepsilon)$	+0,19	+0.08	+0.22	+0,05	+0,01

Используя данные табл. 4.23, отношение весов факторов принимаем следующим:

$$\alpha'_A:\alpha'_B:\alpha'_C:\alpha'_D:\alpha'_E = 0{,}19:0{,}08:0{,}22:0{,}05:0{,}01.$$

Модель оценки степени достижения цели в линейной форме после нормировки критериев представляется в виде

$$H=0{,}35A+0{,}15B+0{,}4C+0{,}09D+0{,}01E.$$

Различным структурам модели соответствуют различные оценки степени адекватности исследуемой системы отношению порядка, что позволяет строго обосновать выбор структуры модели цели консультирования.

Предлагаемый метод оценки адекватности различных структур модели целей консультирования и ее параметров отношению порядка позволяет с достаточной точностью идентифицировать структуру модели целей ЛФР. При этом способ количественного определения числовых оценок вектора приоритета структуры модели дает возможность количественно определить весовой вектор целевой функции ЛФР при минимальной информации, полученной от него в интерактивном режиме.

С учетом особенностей консультирования в САК и формализованной постановки задачи исследования комплекса моделей, изложенных в п. 4.4, модель критериального оценивания состояния консультационного процесса должна удовлетворять следующим требованиям:

1. Выходные параметры модели критериального оценивания должны быть отображены в систему с отношениями

$$\overline{N}_H=(N_H;\ R_N^i,\ i\in I)$$

2. Композиция выходных параметров модели $\gamma_N \circ \gamma_F$

должна взаимно-однозначно отображать систему с отношениями $(N_Y;\ R_N^{\ i},\ i\in I)$ в систему с отношениями $(N_{HY};\ R_{NH}^{\ i},\ i\in I)$.

3. Входные параметры модели критериального оценивания должны быть выходными для модели консультационного процесса — системы с отношениями $(N_Y;\ R_N^{\ i},\ i\in I)$.

4. Модель критериального оценивания состояния консультируемой проблемы должна быть адекватна отношению порядка ε для системы $(H;\ R_H^{\ i},\ i\in I)$.

В п. 4.4 дано определение модели критериального оценивания состояния S консультационного процесса, адекватной отношению ε заданной системы с отношениями $\overline{F} = (F;\ R_F^{\ i},\ i\in I,\ \delta^{-1}(\varepsilon))$ как гомоморфизма $\delta\in\Delta$ в систему с отношениями $\overline{N}_F = (N_F;\ R_N^{\ i},\ i\in I,\ \varepsilon)$, причем $F\cap Y\neq\varnothing$. Там же предложено для задач исследования свойств модели разбить числовую систему с отношениями $\overline{N}_F=(N_F;\ R_N^{\ i},\ i\in I,\varepsilon)$ на две такие системы N_{FA} и N_{FY}, что

$$(N_F;\ R_N{}^i,\ i\in I,\ \varepsilon)\to$$
$$\to(N_{FA}(N_A(A)),$$
$$N_{FY}(N_Y(Y));$$
$$R_{NA}{}^i,\ R_{NY}{}^i,\ i\in I,\ \delta(\varepsilon)),$$

где $N_A(A)$ есть образ отображения $d_N : A\to N_A;\ N_Y(Y)$ есть образ отображения $\gamma_N : A\to N_A$. При этом определяются такие отображения d_{0N} и γ_{0N}, для которых числовая оценка степени адекватности модели отношению порядка ε принимает максимальное значение, т. е.

$$\max\rho(\varepsilon(d_{ON},\ \gamma_{ON};\ R_{AN}{}^i,\ R_{NY}{}^i,\ i\in I,\ \delta(\varepsilon)))=1,$$

где $R_{NA}^{\ i},\ R_{NY}^{\ i},\ i\in I,\varepsilon$ — множество отношений на множествах N_{FA} и N_{FY}.

Отображения d_N и γ_N, где $d_N,\ \gamma_N\in\Delta$, на практике реализуются диалоговыми процедурами, l-й такт которых описывается полиномом

$$\bar{B}^l=[(\bar{H},\ \bar{Y},\ \bar{A})+(\bar{Y},\ \bar{H},\ \bar{A}+B^{l-1}\Xi z)\mu]d,$$

$$l=\overline{1,\ L},\ B^0=\bar{Y},\ \bar{H},\ \bar{A},$$

где μ — состояние САК; d — состояние ЛФР; $\overline{Y}$ — консультационный процесс; $\overline{H}$ — цель консультирования.

На рис. 4.30 представлена задача оценки процедур для интерактивного отображения отношений предпочтения ЛФР.

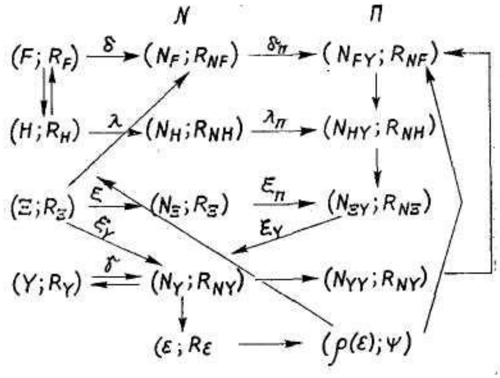

Рис. 4.30. Процедуры отображения отношений предпочтения ЛФР.

4.7.9. Процедуры интерактивного отображения отношений предпочтения ЛФР

Изложим описание нескольких практических процедур отображения предпочтений ЛФР при оценке непрерывного фактора:

$$d_N: A \rightarrow N_A; \quad \gamma_N: Y \rightarrow N_Y.$$

В случае, когда $Y=\{y_{1,}\ y_2 ...,\ y_k\}$, k=1, 2, …, n, где n — множество различных уровней непрерывного фактора y, $N \rightarrow \infty$. Использование для получения отображений $d_N \in \Delta$, $\gamma_N \in \Delta$ интерактивного режима позволяет выявить в момент времени t оценку предпочтения конкретного ЛФР $r^2(y_k, y_d, k, t, \overline{H})$ в отношении цели $\overline{H}$. Здесь y_k и y_d — объекты сравнения; k — индивидуум, которому соответствует это отношение. В п. 4.7.2 описаны интерактивные процедуры формализации дерева целей консультирования. В результате диалога на L-м такте сформируется дерево целей консультирования с отношением предпочтения

$$\gamma_N{}^m[\overline{H}, \overline{Y} + (\nu_B{}^L \bigcirc \pi_\Phi z + \gamma_N{}^{m-1})\, \mu]\, d, \qquad m = \overline{1, M}; \ \gamma^0 = 0,$$

где $\gamma_N{}^m$ — реакция ЛФР на m-м такте диалога; μ — состояние САК; d_L — состояние ЛФР; $v_B{}^L$ — реакция САК на L-м такте построения модели целей консультирования; $\overline{H}$, $\overline{Y}$ — состояние САК при достижении множеством выходов Y целей консультирования Y.

Вопрос, формируемый комплексом моделей с помощью рефлексивных полиномов, записывается в следующем виде:

$$\gamma_N{}^m(\mu)=[\gamma_N{}^{m-1}(d)+\nu_L(d)]\,\mu,\ m=\overline{1,M};\ \gamma_N{}^0=0,$$

где $\gamma_N{}^m\,(\mu)$ — вопрос, формируемый комплексом моделей на m-м такте; $\gamma_N{}^{m-1}\,(d)$ — ответ ЛФР на предыдущем такте; $v_L\,(d)$ — ответ ЛФР на L-м такте построения моделей целей консультирования. Вопросы

$$\tilde{\gamma}_N{}^m(\mu)\in\Gamma$$

в случае непрерывных факторов обозначим

$$\tilde{\gamma}_{NC}{}^m(\mu)\in\Gamma_C\subset\Gamma.$$

Существует несколько методов отображения отношения порядка в различные шкалы, определения которых даны в п 4.3. При этом вопрос $\gamma_N(\mu)$ формулируется следующим образом: «Фактор $n\in v^L$ непрерывный?» В случае утвердительного ответа комплекс программ подключается к блоку, отображающему множество $N_B\rightarrow N_Y$. В случае отрицательного ответа рассматриваемый фактор считается дискретным.

Существует ряд методов практического отображения отношений предпочтения ЛФР для дискретных факторов $\gamma_{ND}{}^i\in\Gamma_d\subset\Gamma$, где $\gamma_{ND}{}^i$ обозначает i-й метод отображения отношений предпочтения ЛФР. При этом одним из перспективных способов их практической реализации является применение интерактивных процедур, отображающих

$N_Y\rightarrow N_{FY};\ Y\rightarrow N_Y$. В случае, когда оцениваемый фактор имеет только два уровня, формируемые вопросы, представленные с помощью рефлексивных полиномов, будут записаны в следующем виде:

$$\tilde{\gamma}_{ND}{}^m(\mu)=[\gamma_{ND}{}^{m-1}(d)+\nu^L(d)]\,\mu,$$

а ответы ЛФР — в виде

$$\tilde{\gamma}_{ND}{}^m(d)=[H\cap Y+\gamma_{ND}{}^{m-1}(d)+\nu^L(\mu)]\,d.$$

Обозначим через n_l^k k-й уровень l-го фактора при достижении H-й цели, причем k=1, 2,…, K; l=1, 2,…, L. В случае дискретного фактора K=2 ЛФР отвечает на вопрос о том, какой именно из двух уровней предпочтителен в отношении цели H, и комплекс моделей биективно отображает этот уровень на множестве N_{HFY} или {0,1}.

В общем случае, когда $K\geq3$, после определения наиболее предпочтительного уровня n_l^1 и наименее предпочтительного n_l^k остальные K-2 уровней упорядочиваются в отношении степени достижения цели H так, чтобы $n_l^1>n_l^2>\ldots>n_l^k$, для чего используются процедуры интерактивного отображения, описываемые рефлексивными полиномами в виде

$$\pi^m(d)=(H\cap Y+\pi^{m-1}\mu)\,d,$$

где $\pi^m(d)$ — реакция ЛФР на m-м такте диалога процедуры биективного отображения множества Y на множество B_H, π — отображение отношений предпочтения ЛФР, $n \bigcirc v$ — композиция отображений n и v на множество N. В результате применения одного из методов γ_{HD} оценивания уровней фактора реализуются отображения множеств

$$Y \to N_{FY};\ A \to N_{FA}, \text{ где } N_Y \subset N.$$

4.7.10. Оценка процедур интерактивного отображения отношений предпочтения ЛФР

Обозначим через $N_{FA}(A, d_F)$ образ множества A при отображении d_F, а через $N_{FA}(A, \Delta)$ — множество всех образов множества A, полученных при гомоморфизме Δ. Проблему оценки степени адекватности множества значений $N_{FA}(A, d_F^i)$, где i — метод отображения отношений R_{RA} на множество N_{FA} при оценке достижения H-й цели, удобно исследовать, если ЛФР предъявляются в интерактивном режиме несколько оценок состояний консультируемой проблемы, принадлежащих области Парето, т. е. оценок, для которых

$$N_{FY}(y_{j+1}{}^k) > N_{FY}(y_j{}^l) \to (\exists y_i)\,(y_i \in Y)\, N_{FY}(y_{i+1}{}^q) < N_{FY}(y_i{}^p).$$

Другими словами, для l-го состояния при увеличении оценки по j-му показателю существует хотя бы один i-й показатель, оценка по которому уменьшается. Эти значения для i=l, 2,…, n; p, k, l, q= l, 2,…, L вводятся в модель оценки степени достижения цели с использованием свертки аддитивной функции полезности

$$N_{FY}(y_{1i}{}^k, y_{2i}{}^l, \ldots, y_{ni}{}^m) = \sum_{j=1}^{n} \alpha_{ij} n_j(y_{ij}).$$

Здесь

$$\alpha_1 : \alpha_2 : \ldots : \alpha_n = \rho_1(\varepsilon(N_{1B})) : \rho_2(\varepsilon(N_{2B})) : \ldots : \rho_n(\varepsilon(N_{nB}));$$

ρ — коэффициент ранговой корреляции оценки адекватности отношению порядка ε в зависимости от структуры модели целей консультирования. Аналогично определяются значения

$$\rho_i(\varepsilon(\gamma_N{}^i)) = 1 - \frac{6(\Sigma_j (N_j^+ - N_j^+(d))^2)}{n(n^2-1)},$$

где N_j^+ — натуральное число (ранг), присвоенное отображением $\pi \bigcirc v \bigcirc \gamma_N$ на множестве N_{FY}; $N_j^+(d)$ — композиция $\pi \bigcirc v \bigcirc \gamma_n$

порядка j -го состояния. Вместо свертки аддитивной функции полезности в модели оценки степени достижения цели можно использовать и другие функционалы.

Оптимальной процедурой отображения предпочтений ЛФР назовем такое отображение γ_{ON}, которому соответствует максимальное значение коэффициента степени адекватности модели отношению ε, выявленному для ЛФР в области Парето.

Согласно структуре комплекса моделей формирования повторяющихся рекомендаций, описанной в п. 4.6.2, между моделями критериального оценивания $(N_{FA}, N_{FY}, R_{NA}^{\ i}, R_{NY}^{\ i}, i\in I, \varepsilon)$ степени достижения цели $(N_{HA}, N_{HY}, R_{NH}^{\ i}), R_{NH}^{\ i}(Y)$, $i\in I$, ε), $I=\{1, 2,...,n\}$ и моделью консультационного процесса $(N_Y; R_N^{\ i}, i\in I, \varepsilon)$ существует следующая функциональная связь. Множество выходов модели консультационного процесса Y является множеством входов для модели критериального оценивания F, выход которой представляет собой множество действительных чисел (оценок) N_F, над которыми выполняется операция свертки в модели цели консультирования. Выход модели оценки степени достижения цели есть множество действительных чисел, биекция которых на множество состояний консультируемой проблемы сохраняет адекватность отношению порядка r^2.

Пример. Воспользуемся моделью оценки степени достижения цели, полученной в п.4.7.8:

$$N_H=0{,}35A+0{,}15B+0{,}4C+0{,}09\,D+0{,}01\,E.$$

Пусть, например, факторы D и E равны нулю, а факторы *A, B а C* дискретны и имеют четыре уровня. В ходе диалога ЛФР упорядочило уровни факторов так, что

$$A^4> A^3>A^2>A^1; \quad B^4> B^3>B^2>B^1; \quad C^4>C^3>C^2>C^1.$$

метрики

Применяя метод γ_{N1} непосредственной оценки, ЛФР присвоило значения уровням фактора, как показано в табл. 4.24.

Таблица 4.24

Ценность уровней факторов для метода непосредственной оценки

Номер уровня	Ценность фактора		
	A	B	C
4	100	100	100
3	20	30	50
2	30	10	20
1	0	0	0

САК задает ЛФР следующий вопрос: «Укажите, какое Вы рекомендуете числовое значение (от 0 до 100) оценки предпочтительности уровней A_i (i=l, 2,3,4) при достижении цели H».

Используя метод γ_{N2} непосредственно упорядоченной метрики, можно получить оценки, представленные в табл. 4.25.

Таблица 4.25

Ценность уровней факторов для непосредственно упорядоченной метрики

Номер уровня	Ценность фактора		
	A	*B*	*C*
4	6	6	6
3	5	5	5
2	3	3	3
1	0	0	0

Тогда ЛФР получает такое задание: «Упорядочите в восходящем порядке разности ценностей

$$[N_{FY}(A^j)-N_{FY}(A^{j+1})]; \quad [N_{FY}(B^j)-N_{FY}(B^{j+1})];$$
$$[N_{FY}(C^j)-N_{FY}(C^{j+1})],$$

где j=1, 2, 3». Пусть ЛФР, сопоставив следующие друг за другом факторы из первой ранжировки методом γ_{N2}, установило следующее отношение порядка ценностей:

$$[N_{FY}(A^4)-N_{FY}(A^3)]>$$
$$>[N_{FY}(A^3)-N_{FY}(A^2)]>[N_{FY}(A^1)].$$

Положив, что $N_{FY}(A^1)=0$, получим следующие числовые значения оценок:

$$N_{FY}(A^1)=0, \quad N_{FY}(A^2)=3, \quad N_{FY}(A^3)=5, \quad N_{FY}(A^4)=6.$$

Для определения числовой оценки степени адекватности полученных результатов ЛФР предъявляются следующие четыре вектора уровней:

$$y_1=(A^3, B^3, C^3)\ ; \quad y_2=(A^2,B^2,C^2);$$
$$y_3=(A,B,C); \qquad y_4=(A^1, B^1,C^1),$$

которые оно упорядочивает в отношении степени достижения цели: $y_2> y_3> y_1>y_4$. Подставив в выражение для оценки степени достижения цели числовые значения уровней из табл. 4.24, получаем следующие результаты упорядочения:

$N_F(y_1)$ = 0,35•50 + 0,15•30 + 0,40•50 = 42;
$N_F(y_2)$ =0,35•30 + 0,15•30 + 0,40•20 = 23;
$N_F(y_3)$ =0,35•50+0,15•10 + 0,40•50=39;
N_F (y_4) = 0,35 • 0+0,15 • 0 + 0,40 • 0 = 0.

Для непосредственно упорядоченной метрики оценки степени достижения цели принимают следующие значения:

$N_F(y_3)$=4,5; $N_F(y_1)$=3,0;
$N_F(y_2)$=4,6; $N_F(y_4)$=0.

Коэффициенты ранговой корреляции, полученные методами γ_{N1} и γ_{N2}, будут

$$\rho(\gamma_{N1}) = 1 - \frac{6 \cdot 2}{4(16-1)} = 0{,}8\,;$$

$$\rho(\gamma_{N2}) = 1 - \frac{0}{6} = 1\,.$$

Для отношения порядка, заданного ЛФР, значение коэффициента $\rho(\gamma_{N2})$ больше, чем $\rho(\gamma_{N1})$. Следовательно, метод $\rho(\gamma_{N2})$ предпочтительнее при построении моделей для САК.

4.8. Моделирование результатов применения сформированных рекомендаций и их оценка

4.8.1. Построение модели консультационного процесса

Рассмотрим более подробно проблему построения модели консультационного процесса для оценки применяемых рекомендаций.

Согласно определению, *под моделью консультируемой проблемы, адекватной отношению порядка заданной системы с отношениями* *(Y, R_Y^i, $i \in I$, $\gamma^{-1}(\varepsilon)$), понимается гомоморфизм $\gamma \in Г$в числовую систему с отношениями (N_Y; R_N^i, $i \in I$, ε), отображенную для формирования рекомендаций в систему с отношениями*

$$(N_Y; R_N{}^i, i \in I, \varepsilon) \to (N_{YA}(Y_A); \quad R_N{}^i, i \in I, \gamma(\varepsilon)).$$

В этом гомоморфизме консультируемой проблемы наиболее важным при построении является отображение d_Y:$A \to Y_A$, которое можно исследовать косвенным образом через систему с отношениями N_{YA}. Иначе говоря, необходимо найти отображение

$$d_Y \in D: \rho(d_Y, \varepsilon(\bar{N}_{YA}, \bar{Y}_A)) = \max.$$

Задача оценки адекватности степени декомпозиции модели имеет структуру, представленную на рис. 4.31.

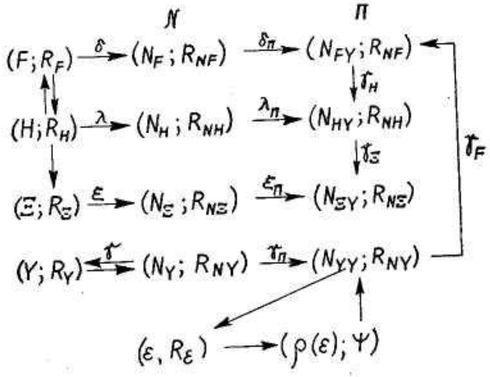

Рис. 4.31. Оценка адекватности степени декомпозиции цели консультирования (отношения в задаче).

Определение 19. Систему с отношениями

$$Y; R_Y^1 \wedge R_Y^3 \wedge \ldots$$

$$\ldots \wedge R_4^{2(n-4)+1} \wedge R_4^{2(n-4)+2}$$

будем называть подсистемой (n —2)-го порядка, адекватной отношению ε для системы с отношениями $(Y, R_Y^i, i\in I)$.

Определение 20. Под субъективной вероятностью $p(y_k)$ будем понимать отображение на множество действительных чисел в интервале [0,1] степени уверенности ЛФР в наступлении события y_k из множества Y, $y_k\in Y$. При этом $p(y_k)>p(y_l)$ будет означать, что ЛФР оценивает возможность появления y_k большей величиной, чем y_l.

В случае статического управления процессом система с отношениями $(Y; R_Y^i, i\in I)$ при отображении в числовую систему с отношениями $(N_{YA}(Y_A); R_N^i, i\in I)$ не имеет отображений множества

$Y\to A$, так как композиция $d_Y O \gamma_N$ для множеств Y и A будет гомоморфизмом $(N_Y; R_N^i, i\in I, \varepsilon)$:

$$(Y; R_Y^i, i\in I, \gamma^{-1}(\varepsilon)) \to (N_Y; R_N^i, i\in I, \varepsilon) \to$$

$$\to ((N_Y \times N_Y); R_N^i, i\in I, \varepsilon).$$

В системе уравнений (4.62) — (4.69) множество альтернатив A представляется единицей множества Y. Модель консультационного процесса представляется графом $(N_Y(Y); R_N^i, i\in I, \gamma(\varepsilon), \Gamma)$, изображенным на рис. 4.32.

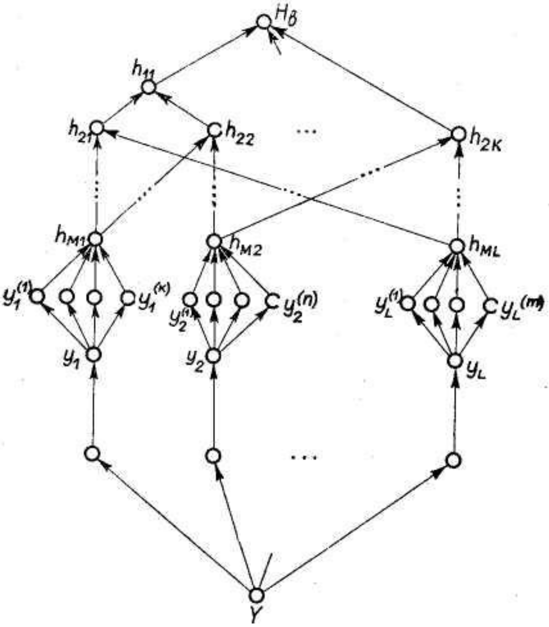

Рис. 4.32. Отношения в статической модели консультационного процесса.

Для множества

$$Y=\{y_1^n, y_2^n, ..., y_k^n\},$$

где n — максимально возможное число уровней переменных $y \in Y$, модель консультационного процесса представляется в виде декартова произведения множеств:

$$Y^k=y_1^i \times y_2^i \times ... \times y_k^i, \quad i=1, 2, \dots, n,$$

где i — номер уровня фактора.

Так как для каждого кортежа будет существовать отображение $\gamma_N : y_m \rightarrow N^*$, $m=1, 2, \dots, k$, то множество всех кортежей Y^k упорядочивается по степени достижения цели консультирования H. Описанные отношения в статической модели консультационного процесса представлены на рис. 4.32. Максимальная длина пути на графе $(N_Y(Y^k),\ R_N^i,\ i \in I,\ \gamma(\varepsilon),\ \Gamma)$ равна 1, минимальная — 0. Заметим, что в случае, когда вместо множества Y в модели будет множество альтернатив A, характеристиками которых является множество показателей Y^k, т. е.

$$A = Y^k=y_1^i \times y_2^i \times ... \times y_k^i, \quad i=1, 2, \dots, n,$$

для оценки альтернатив по достижению заданной цели консультирования H используется граф статической модели.

В случае, когда на графе $(N_Y(Y^k),\ R_N^i,\ i\in I,\ \gamma(\varepsilon),\ \Gamma)$ существуют пути, длина которых μ больше единицы, т. е. когда из вершины y_0 в вершину n_{0k} можно попасть только по двум дугам $(\mu \geq 2)$, имеет место *динамический объект моделирования*. Граф динамической модели консультационного процесса представлен на рис. 4.33.

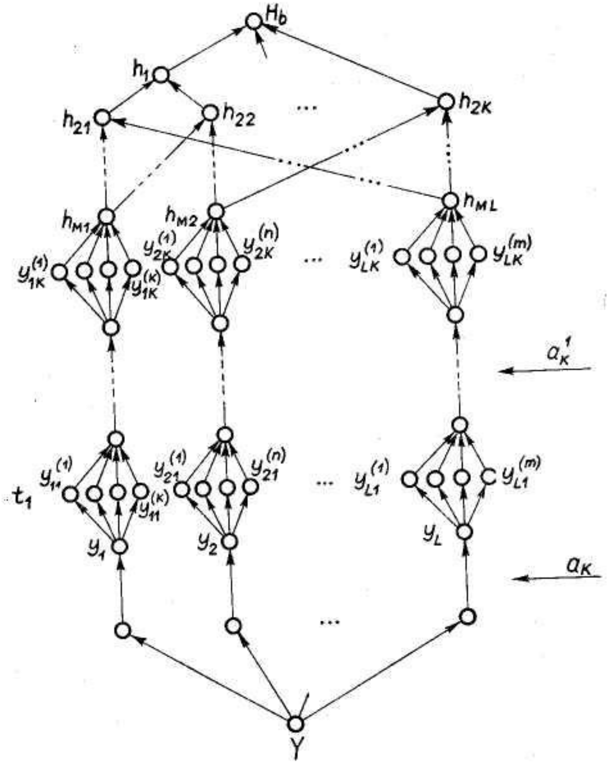

Рис. 4.33. Отношения в динамической модели консультационного процесса.

Как уже отмечалось ранее, случай определенности отношений R_Y^i соответствует заданному распределению вероятностей относительно каждой альтернативы (условие риска) $\forall\, a\in A$ для n переходов на множество уровней $Y = \{y_i^n,\ ...,\ y_{k.}^n\}$. На t-м этапе $(t$=1, 2,...,$T)$ значение оценки степени достижения цели консультирования с использованием байесовских решающих правил равно математическому ожиданию:

$$N_H{}^T(a_k{}^t | y_m{}^{t-1}) = \Sigma_j p_j N_{H_{kj}}(y_m{}^{t-1}),$$

причем на первом этапе

$$N_H{}^1(a_k{}^1 | y_0) = \sum_{j=1}^{n} p_j N_{H_{kj}}{}^t(y^0).$$

Множество решений $A_0 = \{a_0{}^1, \ldots, a_0{}^k\}$, где $a_0{}^t = \{N_{HY}{}^t(a_k{}^{t-1})\}_{k=1}{}^{m-1}$, является оптимальным по критерию Байеса, если в нем для любого $t = 1, 2, \ldots, T$ решения $a_0{}^t(y_k{}^{t-1})$ находятся из условия

$$N_H{}^t(N_{HY_k}{}^t | y_k{}^{t-1}) = \min N_H{}^t(a_k{}^t | y_m{}^{t-1}).$$

Основным недостатком байесовских оптимальных решений является то, что выбор альтернативы $a_0{}^t \in A$ не зависит от ранее выбранных воздействий $a_0{}^1, \ldots, a_k{}^{t-1}$ и от возможных состояний консультационного процесса на предыдущих этапах.

4.8.2. Субъективные вероятности и условие неопределенности

Ранее отмечалось, что в случае неопределенности отношений к началу формализации отображения $d_Y{}^{-1}: Y \rightarrow A$ множество $Y = d_Y(A) = \varnothing$. Проблема заключается в определении числовых характеристик частоты появления события y_j в зависимости от a_k при состоянии y_j. Здесь можно применить подход, основные положения которого состоят в следующем. Если задано отношение порядка на оценках частот появления $p(y_j | a_i, y_i)$, $\forall j \in I$, $I = \{1. 2, \ldots, n)$, т. е.

$$p(y_1 | a_i, y_i) > p(y_2 | a_i, y_i) > \ldots > p(\, p(y_n | a_i, y_i)),$$

то можно отобразить отношение порядка на указанных оценках в подмножество действительных чисел $N_{YA} \subset N$, удовлетворяющих следующему условию:

$$\exists p: p \in N_{YA} \wedge p = \gamma(y_i, a_i, y_j)$$

где $\gamma(y_i, a_i, y_j)$ — отображение трехместного отношения; y_i — состояние консультируемой проблемы; a_i — альтернатива, оцениваемая относительно возможности появления y_j. При этом необходимо выполнение ограничений

$$\sum_{j=1}^{n} p(a_i, y_i, y_j) = 1; \quad p(a_i, y_i, y_j) > 0, \quad \forall i, j \in I.$$

При построении модели консультационного процесса принимается, что начальные числовые оценки $p(y_i, a_i, y_j)$ будут членами арифметической прогрессии, каждый из которых определяется по формуле

$$p_j = \frac{n-j+1}{n(n+1)}.$$

В случае, когда ЛФР не может выявить отношение порядка на частотах появления состояний y_j $(j=1, 2,\ldots, n)$, их можно априори считать равновероятными:

$$p_1 = p_2 = \ldots = p_j = \ldots = p_n = \frac{1}{n}.$$

В теоретическом плане существует возможность получения числовых характеристик наступления события y_j в зависимости от a_i и y_i. Возможны и другие реализации получения оценок на частотах появления состояния консультационного процесса.

Существует ряд методов получения оценок субъективных вероятностей. Наиболее универсальным можно считать метод точечных оценок, основанный на выявленном отношении порядка появления состояний y_j

$$p(y_1) > p(y_2) > \ldots > p(y_n).$$

Этот метод требует минимального объема предварительной информации от ЛФР, что является существенным положительным фактором при практической реализации диалоговых процедур.

Без предварительного введения множества альтернатив задача оценки субъективной вероятности в какой-то степени теряет смысл, так как будет в основном отражать субъективное отображение консультационного процесса при нулевой альтернативе a_0, т. е. случай невмешательства ЛФР в осуществление процесса. Поэтому интерактивный режим включает такты определения множества альтернатив

$$v^M = (\bar{H} \cap \bar{Y} \cap \bar{A} + (\bar{H} \cap \bar{Y} \cap \bar{A} + \Phi^{M-1} z)\mu)d,$$

где μ — состояние САК; d — состояние ЛФР; $M=1, 2,\ldots, S$ — номер такта диалога; v^M описывает сведения, полученные ЛФР, на основе формализованных моделей цели консультирования $\bar{H}$ и критериального оценивания состояния консультационного процесса $\bar{F}$. Вопросы, которые САК ставит перед ЛФР в процессе построения модели консультируемой проблемы, аналогичны по структуре используемым в модели построения цели консультирования. Первый вопрос может быть следующим: «Назовите множество возможных альтернатив для достижения цели консультирования H_{HB}, разделив их запятой». После ответа ЛФР в САК вводится информация, описываемая следующим полиномом:

$$\nu(d) = (\bar{Y} \cap \bar{H} \cap \bar{A})\,\mu, \quad A = \{a_1, a_2, \ldots, a_m\}.$$

Другими словами, системы с отношениями $\bar{H}$ и $\bar{Y}$ дополнены множеством A без отношений на нем.

Далее ЛФР, отвечая на вопрос САК: «Уточните, какие более мелкие мероприятия вы подразумеваете под альтернативой $a_1 \in A$», разбивает множество A на подмножества с помощью интерактивных процедур.

Следующая группа интерактивных процедур отображает множество отношений R_A^i, i=l, 2, ..., I, на множество A, для чего из множества состояний Y выбирают такое y_n, которое наименее предпочтительно для достижения цели консультирования H, т. е.

$y_1 > .. . > y_{n-1} > y_n$. Вопрос САК формируется по известным методам, например: «Какая из перечисленных альтернатив наиболее эффективна для достижения цели N_{HB}?» Ответ ЛФР на третьем такте интерактивной процедуры можно формально описать полиномом

$$\pi^3(\mu) = (\bar{Y} \cap \bar{H} \cap \bar{A} + (\bar{Y} \cap \bar{H} \cap \bar{A} + \Phi^1 z)\,\mu)\,d.$$

После S тактов состояние комплекса моделей описывается следующим рефлексивным полиномом:

$$\pi^S(d) = ((\bar{Y} \cap \bar{H} \cap \bar{A}'\,;\, R_Y{}^i, R_H{}^i, R_{OA}{}^i, \ \ i \in I)\,d)\,\mu, \ A' \subset A,$$

где R_{OA}^i — подмножество отношений порядка $R_{OA}^i \subset R_A^i$, назначенных ЛФР на множестве A.

Вторая группа вопросов следующих l тактов дополняет множество R_{OA}^i до R_A^i, причем их особенностью будет трехместное отношение порядка в отличие от бинарного на предыдущих тактах. Вопрос формулируется следующим образом: «Какое из перечисленных состояний будет наиболее вероятным исходным при применении альтернатив a_i для состояния y_i?» После того как ЛФР удалит из списка названную альтернативу, САК ставит аналогичный вопрос, но предъявляет оставшиеся n—1 состояний.

Эти процедуры продолжаются до тех пор, пока не останутся два состояния. В итоге в САК отображается отношение порядка на множестве условных вероятностей перехода из j-го состояния в i-е при применении альтернативы a_i

$$p(y_1|a_i, y_j) > p(y_2|a_i, y_j) > \ldots > p(y_n|a_i, y_j).$$

Вопросы, связанные с отношениями на множестве альтернатив, можно дополнить отношением полезности применяемой альтернативы a_i для ЛФР с учетом возможностей ее реализации и последствий, т. е. с учетом достижения не только цели консультирования, но и собственных целей ЛФР. При этом вопросы формулируются, например, следующим образом: «Реализация какой из перечисленных альтернатив наиболее приемлема для вас при состоянии y_i объекта?» После удаления из списка названной альтернативы задается аналогичный вопрос до тех пор, пока не останется одна альтернатива. В итоге после ответа ЛФР состояние системы может быть описано полиномом

$$\pi = (\bar{H} \cap \bar{Y} \cap \bar{A} + (\bar{H} \cap \bar{Y} \cap \bar{A})\mu)d,$$

где $\bar{A}$ — система с отношениями $(A;\ R_A^i,\ i \in I,\ I = 1, 2, \ldots, n)$. Обработанные ответы ЛФР поступают на вход моделей критериального оценивания.

Частичной неопределенностью назовем ситуацию принятия решений, когда имеются рекомендации по $n \geq 4$ реализациям консультационного процесса $(Y^m;\ (R_Y^i)^m,\ i \in I),\ I = 1, 2, \ldots, n$, однако отсутствуют системы с отношениями

$$\bar{N}_{AY} = (N_{YA}(A); R_{YN}),\ i \in I,\quad I = \{1, \ldots, n\},$$

где A — множество альтернатив, R_{NY}^i — множество отношений на множестве альтернатив. Практическую трудность представляет восстановление композиции $d_{yt}^{-1} O d_{yt+1}$ (t — момент применения альтернативы) из-за отсутствия статистики, связанной с применением альтернативы a^t, и достаточного числа реализаций. В случае, когда

$n = 1, 2, 3$, предлагается в качестве априорной информации, сохраняющей отношение порядка моделируемой системы, использовать частоты условных переходов $p(y_i^{t+1}|y_k^t)$ для моментов времени $t = \{0, 1, 2, \ldots, T\}$:

$$p(y_i^{t+1}|y_k^t) = \frac{m(y_i^{t+1}|y_k^t)}{m(y_k^t)},$$

где $m(y_i^{t+1}|y_k^t)$ — число переходов из состояния y_k^t в состояние y_i^{t+1};

$m(y_k^t)$ — общее число случаев, когда наблюдалось состояние y_k^t.

4.8.3. Оценка последствий применяемых альтернатив методом Монте-Карло

Для оценки последствий применения выбранной альтернативы воспользуемся методами имитационного моделирования. Идея метода Монте-Карло в имитационном моделировании заключается в том, что разыгрывается случайное число от нуля до единицы и в зависимости от попадания его в тот или иной интервал считается, что произошло событие, соответствующее данному интервалу. Возникает вопрос об отображении информации, полученной от ЛФР в интерактивном режиме, на отношения между субъективными вероятностями в имитационной модели.

В п. 4.8.2 был описан способ получения отношения порядка для i-го состояния консультируемой проблемы на множестве альтернатив:

$$(a_1|y_k)r^2(a_2|y_k)r^2\dots r^2(a_{n-1}|y_k)r^2(a_n|y_k),$$

причем в САК оценки состояний консультируемой проблемы к моменту имитации упорядочены так, что $y_1r^2y_2\dots y_{k-1}r^2y_k$. В предположении, что при заданных

$$p(y_{i+1}{}^{t+1}|y_i{}^t, a_1{}^t) > p(y_{i+1}{}^{t+1}|y_i{}^t, a_2{}^t) > \dots > p(y_{i+1}{}^{t+1}|y_i{}^t, a_n{}^t)$$

наиболее вероятное априорное приращение числовой оценки вероятности условного перехода будет

$$\Delta p(y_{i+1}{}^{t+1}|y_i{}^t, a_1{}^t) > \Delta p(y_{i+1}{}^{t+1}|y_i{}^t, a_2{}^t) > \dots$$
$$\dots > \Delta p(y_{i+1}{}^{t+1}|y_i{}^t, a_n{}^t),$$

а наиболее вероятные изменения в числовых оценках значений вероятностей будут происходить за счет изменения условных вероятностей

$$p(y_i{}^{t+1}|y_i{}^t, a_k{}^t),\ \forall i \in I,\ I=\{1, 2, \dots, n\},$$

в упрощенном алгоритме расчета принимается, что

$$\Delta p(y_{i+1}{}^{t+1}|y_i{}^t, a_k{}^t) = \alpha_k p(y_i{}^{t+1}|y_i{}^t, a_k{}^t).$$

Здесь $\sum k\alpha_k = l$; α_k — отображение k-й альтернативы в подмножество $N_A \subset N$ при выполнении соотношения

$$|\Delta p(y_{i+1}{}^{t+1}|y_i{}^t, a_k{}^t)| = |\Delta p(y_i{}^{t+1}|y_i{}^t, a_k{}^t)|,$$

где t — номер шага имитации. Вычисленные значения вероятностей для (t +1)-го шага будут

$$p(y_{i+1}{}^{t+1}|y_i{}^t, a_k{}^t) = p(y_{i+1}{}^{t+1}|y_i{}^t) + \Delta p(y_{i+1}{}^{t+1}|y_i{}^t, a_k{}^t);$$
$$p(y_i{}^{t+1}|y_i{}^t, a_k{}^t) = p(y_i{}^{t+1}|y_i{}^t) + \Delta p(y_{i+1}{}^{t+1}|y_i{}^t, a_k{}^t).$$

Для определения необходимого числа испытаний используются расчетные формулы:

$$N = p(1-p)\left[\Phi^{-1}\left(\frac{1}{2}Q\right)\right]^2 e^{-2}.$$

Здесь N — необходимое число испытаний; p — вероятность события y_i; Q — вероятность того, что частота p^* отклонится от его вероятности p' меньше, чем на e:

$$Q = p[(p^* - p') < e],$$

где Ф — функция Лапласа. С помощью метода статических испытаний определяются математическое ожидание и дисперсия оценок состояний консультационного процесса, для чего выполняется N_e независимых испытаний. Математическое ожидание вычисляется по формуле

$$M[Y, a_k] = \Sigma_i (y_i|a_k) p^*(y_i|a_k) N^{-1}.$$

Значение дисперсии определяется следующим выражением:

$$D[Y, a_k] = (\Sigma_i (y_i|a_k)^2 - M[Y, a_k]^2 N^{-1})^{1/2}.$$

При этом можно вывести зависимости математического ожидания и дисперсии оценок от состояния процесса y_i и применяемой альтернативы a_k:

$$M[Y^{(t+1)}] = f_1(y_i{}^t, a_k{}^t, N^t);$$

$$D[Y^{(t+1)}] = f_2(y_i{}^t, a_k{}^t, M[Y^{(t+1)}]).$$

Это позволяет сделать вывод, что в теоретическом плане существует возможность сравнения в интерактивном режиме последствий применения различных альтернатив на числовых шкалах интервалов и порядка.

4.8.4. Оценка адекватности модели консультационного процесса поведению консультируемой проблемы

При декомпозиции модели консультационного процесса проблема оценки влияния числа уровней выделяемых состояний на степень адекватности в отношении порядка r^2 является довольно актуальной. Формально эта задача представляется следующим образом:

$$\text{opt}\,(S, K) = \max \rho(\varepsilon, S^{(K)} = \{S_1, S_2, \ldots, S_K\} \subset Y^n).$$

Под степенью декомпозиции модели консультационного процесса понимается число подсистем, адекватно отображающих отношения порядка на множестве состояний консультационного процесса.

В дальнейшем $S^{(K)}$ будет обозначать, что система S разбивается на K подсистем, т. е. $S^{(K)} = \{S_1, S_2, ..., S_K\}$ в отличие от $S^K = S_1 \times S_2 \times \ldots \times S_K$, которое применяется для сокращенной записи декартова произведения. Математическое ожидание оценки состояния консультируемой проблемы определяется выражением

$$M[Y] = \Sigma_i y_i p'_{(k)}(y_i) N^{-1} = f(p'(k)\,(y_i)),$$

где $p'(k)\,(y_i)$ — частота появления y_i при k-й степени декомпозиции. В этом случае при применении альтернативы a_q математическое ожидание можно представить в виде

$$M[Y, k, a_q] = [\Sigma_i y_i p'(k)\,(y_i | a_q)] N^{-1} = f(p(k), a_q).$$

В случае, если $a_q \in \varnothing$ (никакие воздействия к консультируемой проблеме не применяются) и $y_i \in \bigcup^{k} S_j$, можно записать

$$M[y_j^{t+1} | y_i^t] = [\Sigma_j y_j^{t+1} p'(k)\,(y_j^{t+1} | y_i^t)] N^{-1}.$$

Обозначим через $M[y_j^{t+1} | y_i^t]$ математическое ожидание состояния консультируемой проблемы для (t+1)-го шага с оценкой y_i на t-м шаге. При N испытаниях для всех i= 1, 2,…, n полученное множество оценок $M_k=\{M_1(k),...,M_n(k)\}$ упорядочено по степени близости к заданной цели консультирования $M_1(k)>M_2(k)>.. .> M_n(k)$. Далее упорядоченное множество оценок состояний модели консультационного процесса сравнивается с реально наблюдаемым порядком. При этом используется выражение коэффициента ранговой корреляции для k-й степени декомпозиции:

$$\rho_k = 1 - \frac{6[\Sigma_i N_\Delta{}^+ (M_i(k))^2 + T_Y(k) + T'_Y]}{n(n^2-1)},$$

где ρ_k — коэффициент ранговой корреляции Спирмена; $N_\Delta{}^+(M_i(k)) = N^+(M_i(k)) — N^+(y_i)$ — разность i-х рангов переменной $M_i(k)$ и y_i соответственно для комплекса моделей и консультационного процесса; n — максимально возможное число состояний консультируемой проблемы;

$$T_Y(k) = \frac{1}{12} \Sigma_j (t_Y{}^3(k) - t_Y(k))_j; \quad T'_Y = \frac{1}{12} \Sigma_j (t_{Y'}{}^3 - t_{Y'})_j;$$

$(t_Y(k))$ — количество одинаковых рангов множества M_k.; t_Y' — число таких рангов в переменной Y в j-й группе одинаковых рангов; j — номер группы одинаковых рангов. «Оптимальная» степень декомпозиции будет при таком значении $k_0 \in K=\{1, 2,...,M\}$, при котором

$$\mathrm{opt}\,(k, m) =$$
$$= \max \left[1 - \frac{6(\Sigma_i [N_\Delta^+(M_i(k, m))]^2 + T_Y(k, m) + T_Y)}{n(n^2-1)} \right],$$

где m — число испытаний, постоянная величина. С учетом дисперсии для «пессимистической» оценки (когда математическое ожидание

$M_i(k)_- = M_i(k) - 3D_i(Y)$) находим «оптимальную» степень декомпозиции из условия

$$\mathrm{opt}\,(k, D_-) =$$
$$= \max \left[1 - \frac{6(\Sigma_i (N_\Delta^+(M_i(k) - 3D_i(Y)))^2 + T_Y(k, m, D_-) + T_{Y'})}{n(n^2-1)} \right],$$

А для «оптимистической» оценки (когда $M_i(k)_+ = M_i(k) + 3D_i(Y)$)

$$\mathrm{opt}\,(k, D_+) = \max \left[1 - \frac{\begin{array}{c} 6(\Sigma_i (N_\Delta^+(M_i(k) + 3D_i(Y)))^2 + \\ + T_Y(k, m, D_+) + T_{Y'}) \end{array}}{n(n^2-1)} \right].$$

Таким образом, получены оценки различных структур моделей консультационного процесса, характеризующегося множеством состояний $i=\{1, 2, ...,n\}$, что позволяет выбрать такую степень декомпозиции $k_0 \in K$, $K = \{1, 2, ..., M\}$, при которой оценка по критерию адекватности отношения порядка ε модели и консультационного процесса будет максимальна.

Реальная ответная реакция моделируемой рекомендации на выбранную альтернативу $a_k^t \in A$, $k=1, 2, ...,m$, $t= 1, 2, ...,T$, может быть неадекватной отношению порядка ε, назначенному моделью.

Если при наличии N повторяющихся состояний консультируемой проблемы y_i^t в t-й момент времени выбирается альтернатива a_k^t, а на следующем этапе — альтернатива a_k^{t+1}, то с помощью методов статистического моделирования можно получить: q — число событий, заключающихся в том, что консультируемая проблема оказалась в состоянии y_i^{t+2}, причем $y_i^{t+2} > y_j^{t+2}$ по достижению цели консультирования H; n — число событий, состоящих в том, что консультируемая проблема осталась в состоянии y_j^{t+2} в момент времени t +2; m — число событий, заключающихся в том, что консультируемая проблема перешла в состояние y_k^{t+2}, причем $y_k^{t+2}< y_j^{t+2}$. При этом выполняется условие

$$N=q+m+n.$$

Допустим, что ЛФР способен задавать отношение порядка на множестве оценок вероятностей перехода консультационного процесса из состояния y_j^t при применении альтернативы a_k^t.

$$p(y_j^{t+1}|y_j^t, a_k^t) > p(y_i^{t+1}|y_j^t, a_k^t) > p(y_k^{t+1}|y_j^t, a_k^t)\ ;$$

$$p(y_j^{t+2}|y_j^{t+1}, a_p^{t+1}, y_j^t, a_k^t) > p(y_i^{t+2}|y_j^{t+1}, a_p, y_j^t, a_k^t) >$$

$$> p(y_k^{t+2}|y_j^{t+1}, a_p^{t+1}, y_j^t, a_k^t).$$

Если в результате моделирования получены числа n, q и m такие, что $n>q>m$ и консультационный процесс при N испытаниях попадает n', q', m' число раз в состояния y_i, y_j, y_k соответственно, причем $n'>q'>m'$ и $\rho(\varepsilon\ (p'\ (k))) = 1$, то адаптация моделей не требуется. При этом числовые значения

$$p'(y_j^{t+1}|y_j^t, a_k^t), p'(y_i^{t+1}|y_j^t, a_k^t), \ \ldots, p'(y_k^{t+2}|y_j^{t+1}, a_p^{t+1}, y_j^t, a_k^t),$$

полученные с помощью расчетов, остаются без изменений (рис. 4.34).

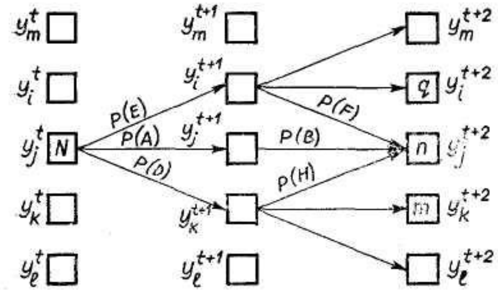

Рис. 4.34. Исходные величины для определения апостериорных вероятностей.

Если степень адекватности отношению порядка $\rho(\varepsilon\ (p'\ (k))) = 1$, при моделировании получено, например, что $n>q>m$, а реальное $q>n = m$, то априорные условные вероятности переходов пересчитываются по формуле Байеса:

$$p^*(y_i^{t+1}|a_k^t, y_j^t) =$$

$$= \frac{p(A)p(B)^{n'}p(C)^{q+m'} + p(E)p(F)^{n'}p(G)^{q'+m'} +}{p(A)p(B)^{n'}p(C)^{q+m'} + p(E)p(F)^{n'}p(G)^{q'+m'} + + p(D)p(H)^{n'}p(R)^{q'+m'}},$$

где

$$A = [y_i^{t+1}|a_k^t, y_i^t]\ ;\quad p(C) = 1 - p(B)\ ;\quad p(R) = 1 - p(H)\ ;$$

$$B = [y_j^{t+2}|a_k^t, y_j^t, y_j^{t+2}, a_p^{t+1}]\ ;\quad p(G) = 1 - p(F)\ ;$$

$$D = [y_k^{t+1}|y_j^t, a_k^t];\quad F = [y_j^{t+2}|a_k^t, y_j^t, y_i^{t+1}, a_p^{t+1}];$$

$$E = [y_i^{t+1}|y_j^t, a_k^t]\ ;\quad H = [y_j^{t+2}|y_j^t, a_k^t, y_k^{t+1}, a_k^{t+1}].$$

Полученные в результате расчетов $p^*(A)$, $p^*(E)$, $p^*(D)$,... корректируют в модели априорные вероятности $p(A)$, $p(E)$, $p(D)$,... После замены условных вероятностей проверяется отношение порядка на состояниях консультируемой проблемы и в случае, если

$$|\rho^*(\varepsilon(p)) - \rho(\varepsilon(p^*))| < e,$$

где e — заданная точность построения модели консультируемой проблемы по степени адекватности отношению порядка ε, адаптация прекращается. Это позволяет заменить в модели субъективные оценки ЛФР относительно эффективности применяемых альтернатив более точными условными вероятностями. Согласно структуре комплекса моделей рекомендаций для принятия повторяющихся решений, изложенной в п. 4.6.2, между моделями консультационного процесса $\bar{N}_{YA} = (N_{YA}, R_N^i, i \in I, \varepsilon)$, $I = \{1, 2, \ldots, n\}$, моделью критериальных оценок $\bar{N}_{FYA} = (N_{FYA}; R_N^i, i \in I, \varepsilon)$ и моделью поведения ЛФР $\bar{\bar{\Xi}} = (\Xi; R_\Xi^i, i \in I, \varepsilon)$ существуют следующие функциональные зависимости.

Множество выходов модели консультационного процесса N_Y является множеством входов для модели критериальных оценок N_{FY}, выходом которой является множество действительных чисел (оценок) N_{FYA}, которое, как описано в п. 4.6, с помощью операций свертки в модели цели консультирования биективно отражается на множество действительных чисел с адекватным отношением ε.

Множество альтернатив A, являющееся выходом модели Ξ, будет входом для модели критериальных оценок N_{FA}. После оценки эффективности $a_k \in A$ по достижению цели H при выходе $y \in Y$ и отображения отношения порядка на множество действительных чисел числовая оценка поступает в модель поведения ЛФР, в которой заложены различные функциональные отображения оценок N_{HYA}, N_{HA} (правила выбора). При рассмотрении для конкретного $y \in Y$ всевозможных троек $A \times Y \times P$, где P — множество вероятностных оценок пары (a, y), P=[0, 1], получается отношение порядка на множестве альтернатив A по достижению цели H при i-м состоянии консультируемой проблемы.

4.8.5. Пример оценки адекватности модели консультационного процесса

В качестве примера рассмотрим процесс имитации успеваемости учебной группы вуза при различных альтернативах консультирования.

Таблица 4.26

*** Результаты имитации при степени декомпозиции 1 : 14**

№ студ. в списке группы	Аттестации			Вероятность полож. аттестации	Результаты имитации		Округленно
	I	II	III		средний балл	отклонение	
01	1	1	0	0,97	3,64	±0,05	4
07	1	0	0	0,93	3,99	±0,06	4
16	1	0	0	0,99	3,99	±0,06	4
03	1	0	0	0,95	3,96	±0,06	4
06	0	1	1	0,94	4,92	±0,13	5
13	0	1	1	0,90	4,99	±0,14	5
02	0	1	0	0,50	3,91	±0,06	4
11	1	1	0	0,93	2,97	±0,06	3
05	1	1	0	0,98	3,00	±0,06	3
14	0	0	0	0,95	3,00	±0,06	3
15	0	1	0	0,93	3,99	±0,06	4
08	1	1	0	0,66	3,01	±0,06	3
09	0	1	1	0,90	3,97	±0,06	4
12	1	1	0	0,96	3,52	±0,05	3
04	1	1	0	1,00	5,00	±0,14	5
10	1	1	0	1,00	5,00	±0,14	5

$\rho(\varepsilon) = 0{,}84$ при уровне значимости 0,001 по критерию Стьюдента.

Таблица 4.27

***Результаты имитации при степени декомпозиции 1 : 2**

№ студ. в списке группы	Аттестации I	Аттестации II	Аттестации III	Вероятность полож. аттестации	Результаты имитации: средний балл	Результаты имитации: отклонение	Округленно
01	1	1	0	0,38	3,95	±0,06	4
07	1	1	0	0,33	3,98	±0,06	4
16	1	1	0	0,40	3,99	±0,06	4
03	1	1	0	0,39	3,93	±0,06	4
06	1	1	0	0,30	3,96	±0,06	4
13	1	1	0	0,34	3,98	±0,06	4
02	1	1	0	0,33	3,93	±0,06	4
11	1	1	0	0,46	4,00	±0,06	4
05	1	1	0	0,47	3,52	±0,05	3
14	1	1	0	0,52	3,49	±0,05	3
15	1	1	0	0,57	3,42	±0,05	3
08	1	1	0	0,55	3,52	±0,05	3
09	1	1	0	0,56	3,39	±0,05	3
12	1	1	0	0,56	3,51	±0,05	3
04	1	1	0	0,45	3,42	±0,05	3
10	1	1	0	0,50	3,52	±0,05	3

$\rho(\varepsilon) = 0{,}76$ при уровне значимости 0,001 по критерию Стьюдента.

Таблица 4.28

*** Результаты имитации при степени декомпозиции 1 : 4**

№ студ. в списке группы	Аттестации I	Аттестации II	Аттестации III	Вероятность полож. аттестации	Результаты имитации: средний балл	Результаты имитации: отклонение	Округленно
01	1	1	0	0,41	3,60	±0,05	4
07	1	1	0	0,38	3,46	±0,05	3
16	1	1	0	0,33	3,82	±0,05	4
03	1	1	0	0,47	3,98	±0,06	4
06	1	1	0	0,35	3,96	±0,06	4
13	1	1	0	0,46	3,98	±0,06	4
02	1	1	0	0,42	3,97	±0,06	4
11	1	1	0	0,40	4,00	±0,06	4
05	0	0	0	0,41	3,97	±0,06	4
14	0	0	0	0,35	3,94	±0,06	4
15	1	1	0	0,35	4,00	±0,06	4
08	1	1	0	0,55	3,00	±0,06	3
09	1	1	0	0,67	2,98	±0,06	3
12	1	1	0	0,49	3,01	±0,06	3
04	1	1	0	0,49	3,02	±0,06	3
10	1	1	0	0,60	3,01	±0,06	3

$\rho(\varepsilon) = 0{,}75$ при уровне значимости 0,001 по критерию Стьюдента.

Таблица 4.29

*** Результаты имитации при степени декомпозиции 1 :8**

№ студ. в списке группы	Аттестации			Вероятность полож. аттестации	Результаты имитации		Округленно
	I	II	III		средний балл	отклонение	
01	1	1	0	0,95	3,53	±0,05	3
07	1	1	0	0,48	4,92	±0,13	5
16	1	1	0	0,61	4,94	±0,13	5
03	0	1	1	0,91	3,99	±0,06	4
06	1	1	0	0,46	3,98	±0,06	4
13	1	1	0	0,59	3,97	±0,06	4
02	1	1	0	0,51	3,99	±0,06	4
11	1	1	0	0,39	3,97	±0,06	4
05	1	1	0	0,91	3,99	±0,06	4
14	0	0	0	0,51	3,99	±0,06	4
15	0	1	0	0,49	3,00	±0,06	3
08	1	1	0	0,60	2,99	±0,06	3
09	1	1	0	0,64	3,00	±0,06	3
12	1	1	0	0,67	3,01	±0,06	3
04	0	1	1	0,56	3,01	±0,06	3
10	0	1	1	0,54	3,00	±0,06	3

$\rho(\varepsilon) = 0{,}82$ при уровне значимости 0,001 по критерию Стьюдента.

Таблица 4.30

*** Результаты имитации при степени декомпозиции 1 : 14 и применении альтернативы «Опрос перед лекцией»**

№ студ. в списке группы	Аттестации			Вероятность полож. аттестации	Результаты имитации		Округленно
	I	II	III		средний балл	отклонение	
01	1	1	0	0,94	3,99	±0,05	3
07	0	1	1	0,59	4,29	±0,07	4
16	0	1	1	0,63	4,23	±0,07	4
03	0	1	1	0,73	4,29	±0,07	4
06	0	1	0	0,91	2,68	±0,08	3
13	1	0	1	0,94	4,22	±0,07	4
02	1	1	0	0,91	3,29	±0,05	3
11	1	1	0	0.94	4,29	±0,07	4
05	1	1	0	0,94	4,20	±0,07	4
14	0	0	0	0,97	3,96	±0,06	4
15	0	1	0	0,92	3,30	±0.05	3
08	0	1	1	0,92	3,54	±0,05	3
09	1	1	0	0,88	3,48	±0,05	3
12	1	1	0	0,97	3,43	±0,05	3
04	0	1	1	0,94	3,59	±0.05	4
10	1	1	0	0.98	3,50	±0,05	3

* ВАС ИНТЕРЕСУЮТ РЕЗУЛЬТАТЫ ДЛЯ ДРУГОЙ АЛЬТЕРНАТИВЫ?
— ДА.

Таблица 4.31

*** Результаты имитации при степени декомпозиции 1 : 14 и применении альтернативы «Система докладов»**

№ студ. в списке группы	Аттестации			Вероятность полож. аттестации	Результаты имитации		Округленно
	I	II	III		средний балл	отклонение	
01	1	1	0	0,92	3,39	±0,05	3
07	0	1	1	0,61	4,14	±0,06	4
16	0	1	1	0,62	4,21	±0,07	4
03	0	1	1	0,63	4,16	±0,07	4
06	0	1	0	0,96	2,41	±0,10	2
13	1	0	1	0,91	4,25	±0,07	4
02	1	1	0	0,84	3,33	±0,05	3
11	1	1	0	0,93	4,12	±0,06	4
05	1	1	0	0,94	4,20	±0.07	4
14	0	0	0	0,94	3,96	±0,06	4
15	0	1	0	0,95	3,21	±0,05	3
08	0	1	1	0,93	4,24	±0.05	3
09	1	1	0	0,97	3,35	±0,05	3
12	1	1	0	0,96	3,41	±0.05	3
04	0	1	1	0,89	3,24	±0,05	3
10	1	1	0	0.95	3,45	±0.05	3

* ВАС ИНТЕРЕСУЮТ РЕЗУЛЬТАТЫ ДЛЯ ДРУГОЙ АЛЬТЕРНАТИВЫ?
— ДА.

Таблица 4.32

*** Результаты имитации при степени декомпозиции 1 : 14 и применении Альтернативы «Проверка промежуточных результатов проектирования»**

№ студ. в списке группы	Аттестации I	Аттестации II	Аттестации III	Вероятность полож. аттестации	Результаты имитации: средний балл	Результаты имитации: отклонение	Округленно
01	1	1	0	0,93	3,25	±0,05	3
07	0	1	1	0,61	4,08	±0,06	4
16	0	1	1	0,60	4,01	±0,06	4
03	0	1	1	0,61	4,03	±0,06	4
06	0	1	0	0,93	2,20	±0,12	2
13	1	0	1	0,91	4,00	±0,06	4
02	1	1	0	0,85	3,09	±0,06	3
11	1	1	0	0,91	4,01	±0,06	4
05	1	1	0	0,97	4,03	±0,06	4
14	0	0	0	0,97	3,96	±0,06	4
15	0	1	0	0,91	3,06	±0,06	3
08	0	1	0	0,91	3,06	±0,06	3
09	1	1	0	0,96	2,99	±0,06	3
12	1	1	0	0,94	3,08	±0,06	3
04	0	1	1	0,90	3,20	±0,05	3
10	1	1	0	0,96	3,07	±0,06	3

* ВАС ИНТЕРЕСУЮТ РЕЗУЛЬТАТЫ ДЛЯ ДРУГОЙ АЛЬТЕРНАТИВЫ?
— ДА.

Таблица 4.33

*** Результаты имитации при степени декомпозиции 1 : 14 и применении альтернативы «Дополнительные консультации»**

№ студ. в списке группы	Аттестации I	Аттестации II	Аттестации III	Вероятность полож. аттестации	Результаты имитации: средний балл	Результаты имитации: отклонение	Округленно
01	1	1	0	0,90	2,91	±0,06	3
07	0	1	1	0,69	3,93	±0,06	4
16	0	1	1	0,70	3,99	±0,06	4
03	0	1	1	0,70	3,95	±0,06	4
06	0	1	0	0,91	2,03	±0,14	2
13	1	0	1	0,92	3,92	±0,06	4
02	1	1	0	0,94	3,00	±0,06	3
11	1	1	0	0,96	3,96	±0,06	4
05	1	1	0	0,92	3,96	±0,06	4
14	0	0	0	0,95	3,92	±0,06	4
15	0	1	0	0,93	3,00	±0,06	3
08	0	1	1	0,97	2,99	±0,06	3
09	1	1	0	0,94	3,01	±0,06	3
12	1	1	0	0,91	2,99	±0,06	3
04	0	1	1	0,91	2,98	±0,06	3
10	1	1	0	0,96	3,04	±0,06	3

* ВАС ИНТЕРЕСУЮТ СРАВНИТЕЛЬНЫЕ РЕЗУЛЬТАТЫ ПРИМЕНЕНИЯ АЛЬТЕРНАТИВ?
— ДА.

Таблица 4.34

*** Сравнительные результаты применения альтернатив 1—5**

№ студ. в списке группы	Реальные результаты аттестации*			Экзамен. оценка*	Альтернативы				
	I	II	III		1	2	3	4	5
01	1	1	1	4	3,39	3,39	3,34	3,25	2,91
07	1	0	0	4	4,29	4,14	4,16	4,08	3,93
16	0	1	1	3	4,23	4,21	4,08	4,01	3.99
03	0	1	1	5	4,29	4,16	4,06	4,03	3,95
06	0	1	0	5	2,68	2,41	2,17	2,20	2,03
13	1	0	1	3	4,22	4,25	4,06	4,00	3,92
02	1	1	1	4	3,29	3,33	3,02	3,09	3,00
11	1	1	0	4	4,29	4,12	4,15	4,01	3,96
05	1	1	0	4	4,20	4,20	4,12	4,03	3,96
14	0	0	0	4	3,96	3,96	3,96	3,96	3,92
15	0	1	0	4	3,30	3,21	3,16	3,06	3,00
08	0	1	1	5	3,54	3,24	3,29	3,06	2,99
09	1	1	1	3	3,48	3,35	3,14	2,99	3.01
12	1	1	0	4	3,43	3,41	3,13	3,08	2.99
04	0	1	1	5	3,59	3,29	3,25	3,20	2,98
10	1	1	1	3	3,50	3,45	3,40	3,07	3,04

* Результаты соответствуют реальным оценкам и аттестациям по исследуемой дисциплине за прошлый семестр.

Текст, исходящий от САК и выведенный на ее видеотерминальное устройство, отмечен символом *. К началу диалога в памяти САК находятся результаты экзаменационных сессий и ежемесячных аттестаций за все время обучения в вузе студентов предыдущего выпуска по специальности «Автоматизированные системы управления» для русского и латышского языков обучения. В табл. 4.26—4.34 единица соответствует положительному результату аттестации по данной дисциплине, нуль — отрицательному.

* КАКАЯ ГРУППА ВАС ИНТЕРЕСУЕТ? (РУС ИЛИ ЛАТ?)

— РУССКАЯ.

* КАКОЙ КУРС?

— 3.

* КАКОЙ СЕМЕСТР? (ОСЕННИЙ — 1, ВЕСЕННИЙ — 2)

— 1

* В ПРОШЕДШЕМ (ЧЕТВЕРТОМ) СЕМЕСТРЕ ИЗУЧАЛИСЬ СЛЕДУЮЩИЕ ДИСЦИПЛИНЫ: 1 — ФИЛОСОФИЯ, 2 — ФИЗИКА, 3 — МАТЕМАТИЧЕСКИЕ ОСНОВЫ КИБЕРНЕТИКИ, 4 — ТЕОРИЯ ИНФОРМАЦИИ, 5 — МЕХАНИКА, 6 — ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ЭЛЕКТРОТЕХНИКИ, УКАЖИТЕ НОМЕРА В

ПОРЯДКЕ УБЫВАЮЩЕЙ ВАЖНОСТИ ДИСЦИПЛИН (ПО ВАШЕМУ МНЕНИЮ) ДЛЯ СПЕЦИАЛИСТА ПО АСУ.

— 4, 3, 2, 6, 1, 5.

* В НАСТОЯЩЕМ (ПЯТОМ) СЕМЕСТРЕ ИЗУЧАЮТСЯ СЛЕДУЮЩИЕ ДИСЦИПЛИНЫ: 7 — ПОЛИТИЧЕСКАЯ ЭКОНОМИЯ, 8 — ИНОСТРАННЫЙ ЯЗЫК, 9 — ИМПУЛЬСНАЯ ТЕХНИКА, 10 — ОСНОВЫ ПОСТРОЕНИЯ АСУ, 11 — АНАЛОГОВЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, 12 — ПРОЕКТИРОВАНИЕ ПОДСИСТЕМ И ЗВЕНЬЕВ АСУ. УКАЖИТЕ НОМЕРА В ПОРЯДКЕ УБЫВАЮЩЕЙ ВАЖНОСТИ ДИСЦИПЛИН (ПО ВАШЕМУ МНЕНИЮ) ДЛЯ СПЕЦИАЛИСТА ПО АСУ.

— 10, 12, 11, 8, 7, 9.

* ДЛЯ КАКОЙ ДИСЦИПЛИНЫ ПРОВОДИТЬ ИМИТАЦИЮ?

— 10.

* КАКИЕ МЕРОПРИЯТИЯ ДЛЯ ПОВЫШЕНИЯ УСПЕВАЕМОСТИ ВЫ ЖЕЛАЕТЕ СРАВНИТЬ?

— 1 — ОПРОС ПЕРЕД ЛЕКЦИЕЙ, 2 — ДОПОЛНИТЕЛЬНЫЕ КОНСУЛЬТАЦИИ, 3 — ПРОВЕРКА ПРОМЕЖУТОЧНЫХ РЕЗУЛЬТАТОВ ПРОЕКТИРОВАНИЯ, 4 — СИСТЕМА ДОКЛАДОВ, 5 — САМОСТОЯТЕЛЬНАЯ РАБОТА.

* ДЛЯ ПОВЫШЕНИЯ УСПЕВАЕМОСТИ МОЖНО ПРИМЕНЯТЬ СЛЕДУЮЩИЕ МЕРОПРИЯТИЯ: 1 — ОПРОС ПЕРЕД ЛЕКЦИЕЙ, 2 — ДОПОЛНИТЕЛЬНЫЕ КОНСУЛЬТАЦИИ, 3 — ПРОВЕРКА ПРОМЕЖУТОЧНЫХ РЕЗУЛЬТАТОВ ПРОЕКТИРОВАНИЯ, 4 — СИСТЕМА ДОКЛАДОВ, 5 — САМОСТОЯТЕЛЬНАЯ РАБОТА. УКАЖИТЕ НОМЕРА МЕРОПРИЯТИЙ В ПОРЯДКЕ УМЕНЬШЕНИЯ ИХ ЭФФЕКТИВНОСТИ ДЛЯ ПОСТАВЛЕННОЙ ЦЕЛИ. — 1, 4, 5, 3, 2.

С учетом особенностей консультирования в САК, изложенных ранее, и формализованной постановки консультационной задачи по формированию рекомендаций исследования комплекса моделей, приведенной в п. 4.6, модель поведения ЛФР должна удовлетворять следующим требованиям:

— выходные параметры модели оценки степени достижения цели функционально отображаются на модель поведения ЛФР

$$(\Xi;\ H_{HA};\ H_{HYA};\ R_{\Xi}^{i};\ R_{NH}^{i},\ i \in I);$$

— выходные параметры модели поведения ЛФР и множество отношений R_N^i в системе с отношениями $(N_{Y\Xi};\ R_N^i,\ i \in I,\ \varepsilon)$ являются входными показателями для блока адаптации (ρ (r^2); φ), показанного на рис. 4.23, реализующего сравнение с множеством отношений R_Y^i в системе с отношениями $\overline{Y}=(Y, R_Y^i, i\in I, \varepsilon)$;

— множество решающих правил Ξ считается заданным, конечным и является структурным элементом модели поведения ЛФР;

— множество альтернатив A считается заданным, конечным и является структурным элементом модели поведения ЛФР;

— в модели поведения ЛФР $(\Xi;\ N_{HA};\ N_{HYA};\ R_\Xi^i,\ R_{NH}^i,\ i\in I)$ возможны изменения структуры решающего правила Ξ, т. е. выбор элемента $\xi\in\ \Xi$, а в структуре решающего правила $(\xi_0, R_\xi^i, i\in I)$, в свою очередь, — отношения R_ξ^i между элементами ξ_0;

— модель поведения ЛФР адекватна отношению порядка ε_i для моделируемой системы с отношениями

$$(\overline{H}\cap\overline{Y}\cap\overline{A};\ R_H^i, R_Y^i, R_A^i\ i\in I, \varepsilon)$$

4.8.6. Оценка степени адекватности формализованного правила выбора поведению ЛФР

Под моделью поведения ЛФР, управляющего консультационным процессом P при заданных множествах целей G и альтернатив A, понимается, согласно определению 12, гомоморфизм $\xi(\xi\in\Xi)$ *для заданной системы с отношениями* $(H,\ Y,\ A,\ R_A^i,\ i\in I,\ \varepsilon)$ *в числовую систему с отношениями* $(N_{HA},\ N_{HY},\ R_N^{\ i},\ i\in I,\ \varepsilon)$. Применительно к структуре консультирования в САК, изображенной на рис. 4.35, это означает, что для заданной цели консультирования G^0, отображенной на множество оценок z_0, выходов консультационного процесса Y_i, отображенных посредством F на множество Z, и множества альтернатив $A=\{a_1, a_2, ..., a_n\}$ модель правил выбора Ξ (A, Y) должна не только найти $A_0=\Xi(A)$, но и упорядочить множество A, т. е. получить систему с отношением порядка $(A, r^1, r^2, \varepsilon)$.

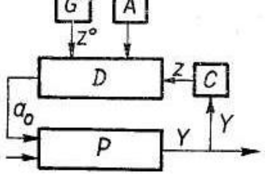

Рис. 4.35. Отношения между составными элементами САК.

В дальнейшем для исследования адекватности отношению порядка ε в отображениях вида

$$\xi_0 : y \to R\alpha \circ y,$$

где R — множество операций свертки, а $\circ$ — множество алгебраических операций. Задача оценки степени адекватности отношению порядка для каждого из формализованных правил рассматривается без учета отношения r^2 между элементами множества Y. Для заданного множества $\Xi_0(A) \in \Xi(A)$ необходимо разработать метод исследования влияния введенного отношения порядка в формализованных правилах выбора на оценку степени адекватности отношению ε, что позволяет приблизить САК к ЛФР. Структура функциональных связей, возникающих при исследовании формализованных правил выбора в комплексе моделей принятия решений, показана на рис. 4.22.

Определение 21. Множеством оптимальных по Парето рекомендаций A называется такое множество, каждая рекомендация которого не может быть улучшена ни по одному параметру без ухудшения хотя бы одного параметра:

$$\exists k \in I : F_1(j) < F_2(i) \to F_1(k) > F_2(i) \wedge F_1(k) > F_1(j).$$

Определение 22. Правилом выбора на множестве парето-оптимальных рекомендаций $\Xi(A)$ назовем функциональное отображение $A \to \varepsilon N$ на множество действительных чисел — оценок

$N_\Xi = Z(A)_B$ с отношением порядка ε.

Чтобы оценить адекватность формализованных правил выбора отношению порядка, воспользуемся ранговой корреляцией, основные положения которой изложены в п. 4.6.6. Если заданы: конечное множество принципов векторной оптимизации Ξ, которое обусловлено практическими возможностями, имеющимися при построении САК, цель консультирования H и множество показателей состояния консультируемой проблемы Y^n, то можно выявить отношение порядка ЛФР по степени достижения цели консультирования G системы с отношениями $(Y^n, R_D^i, i \in I)$. Для принципа векторной оптимизации ξ_1 получается система с отношениями $(Y^n, R_{\xi_i}^i, i \in I, \varepsilon)$. В общем случае имеет место система с отношениями $(Y^n, R_D^i, i \in I)$, где $I = \{1,2, ...,m\}$, есть множество принципов векторной оптимизации.

При использовании принципов векторной оптимизации приоритет критериев может быть задан с помощью: ряда приоритета I — упорядоченного множества индексов локальных критериев

$\vec{I}$ =(1, 2, ..., *m*), весового вектора $\alpha=\{\alpha_1, \alpha_2, ..., \alpha_m\}$ и вектора приоритета

$\lambda=\{\lambda_1, \lambda_2, \dots, \lambda_m\}$. При этом коэффициенты α_j и λ_j связаны между собой следующим образом:

$$\alpha_j=\frac{\prod\limits_{j=1}^{m}\lambda_j}{\sum\limits_{v=1}^{m}\prod\limits_{j=0}\lambda_j}; \quad \lambda_j=\frac{\alpha_j}{\alpha_{j+1}}.$$

При построении моделей использования повторяющихся рекомендаций задание приоритета критериев позволяет отобразить в ряде задач особенности организационных систем, связанные с выявленным отношением порядка.

Для каждого принципа

$$\xi\in\Xi,\ \Xi=\{\xi_1, \xi_2, \ldots, \xi_m\}$$

вычисляется коэффициент ранговой корреляции по формуле

$$\rho(\xi_i)=\rho(\xi_i, H, d, t),$$

где ρ зависит от цели консультирования *H*, момента времени *t*, конкретного ЛФР *d* и формализованного правила выбора ξ_i. Выбирается такое формализованное правило $\xi\in\Xi$, при котором

$$\mathrm{opt}\ (\xi_0)=\max\rho(\xi_j, H, k, t), \quad j\in J,\ J=\overline{1, m}.$$

Заметим, что в формализованных правилах приоритеты показателей оценок состояния консультируемой проблемы принимаются равными между собой:

$$\alpha_1=\alpha_2=\ldots=\alpha_k=\frac{1}{k}.$$

Для заданного числа *n* ($l\le n\le m$), где *m* — число правил выбора, улучшение степени адекватности выхода модели достигается путем изменения весового вектора $\alpha=(\alpha_1, \alpha_2, ..., \alpha_m)$.

4.8.7. Адаптация модели поведения ЛФР по параметрам правил выбора

После выполнения процедур, описанных в п. 4.8.6, определяются зависимости коэффициента ранговой корреляции от изменения весового вектора

$$\rho(\xi_j, \bar{d})=\rho(\xi_j, H, k, t, \bar{\alpha}_i),$$

где $\overline{\alpha}_i$ принадлежит декартову произведению

$$\alpha_1 \times \alpha_2 \times \ldots \times \alpha_n, \quad \Sigma \alpha_i = 1.$$

Затем определяется такой весовой вектор $\overline{\alpha}_0$, при котором

$$\rho(\xi_j, \overline{\alpha_0}) = \max \rho(\xi_j, H, k, t, \overline{\alpha}_0).$$

При решении задачи построения комплекса моделей формирования повторяющихся рекомендаций одной из сложнейших является проблема отображения функции полезности показателей $y \in Y$ для достижения цели *H*. Функция полезности, отражающая предпочтения ЛФР, непостоянна во времени. Кроме того, функции, сохраняющие порядок, могут быть разнообразными: от выпуклых до вогнутых, в зависимости от характера задачи и склонности ЛФР к риску. Используя при построении моделей различные виды функции полезности $(u_i | i \in M = \{1, 2, ..., m\})$ и вычисляя степень адекватности функции отношению порядка ε:

$$\rho^*(\xi_j, \overline{a}_i, u_k) = \max \rho(\xi_j, H, k, t, \overline{\alpha}_i, u_k),$$

определяем оптимальный для комплекса моделей вид функции предпочтения, в большей степени адекватной поведению ЛФР.

При построении комплекса моделей в основном используются качественные шкалы — шкалы наименований и порядка. Целесообразно определить адекватность модели и проблемы в шкале порядка. При этом возможны два критерия адекватности:

а) адекватность консультируемой проблеме;

б) адекватность поведению ЛФР.

В первом случае на вход модели консультируемой проблемы и реальной консультируемой проблемы подается управляющее воздействие и после ряда наблюдений оцениваются прогнозируемый моделями и фактический порядок на множестве исходов.

Во втором случае на вход модели и реальной консультируемой проблемы подается одно и то же состояние консультируемой проблемы $Y_i \neq Y_0$ и сравнивается отношение порядка на множестве *A*, назначенном ЛФР и моделью. В качестве критерия оценки точности комплекса моделей предлагается коэффициент

$$\rho_M{}^H = \frac{1}{12}(1+\rho_A)(1+\rho_P) + \frac{1}{G}(1+\rho_A) + \frac{1}{G}(1+\rho_P),$$

где ρ_A — коэффициент Спирмена, принимающий значение — $1 \leq \rho_A \leq 1$ для оценки адекватности поведения ЛФР; ρ_P — коэффициент Спирмена для оценки адекватности комплекса поведению консультируемой проблемы, принимающий значения — $1\leq \rho_P \leq 1$. При этом в случае полного совпадения отношения порядка для консультируемой проблемы и ЛФР $\rho_м$ = max=1. В противном случае значение $\rho_м$ = min ρ = 0.

Согласно структуре комплекса моделей для формирования повторяющихся рекомендаций, изложенной в п. 4.6.2, между моделями поведения ЛФР $(\Xi,\ H_H,\ N_A,\ R_N^{\ i}\ ,\ i\in I)$, консультируемой проблемой $\bar{N}_Y=(N_Y, R_N^{\ i}\ , i\in I, \varepsilon)$, критериальными оценками $(N_{FYA}, F_{FA}, R_N^{\ i}\ , i\in I, \varepsilon)$ и блоком адаптации (ρ (ε); φ), показанными на рис. 4.23, существуют следующие функциональные зависимости. Множество выходов консультируемой проблемы Y с отношением порядка r^2 является входным множеством для блока адаптации. Это упорядоченное множество (Y, r^2) сравнивается с выходом моделируемого процесса

(Y', r^2) при условии, что (Y, r^2)= (Y', r^2)=f(A).

Изменяя алгоритмическим путем (например, по методу Байеса) условные вероятности перехода $p(Y_v^{t+1}|Y_i,\ a_k^t)$, получаем такие величины p^* $(y_i^{t+1}\ |y_i^t,\ a_k^t)$, при которых модель консультируемой проблемы будет максимально адекватна реальной системе. Выходом блока адаптации будет множество корректирующих воздействий на параметры модели консультируемой проблемы. Если в поведение ЛФР вводится некоторая коррекция, то выходом комплекса моделей будет упорядоченное множество альтернатив A достижения цели G для фиксированного состояния процесса P и множества A. В этом случае упорядоченное множество A' реальной системы сравнивается по степени адекватности с множеством A. Варьируя — например, способом последовательного перебора — структурные элементы и параметры D в моделях поведения ЛФР, критериальные оценки C, оценки степени достижения цели консультирования G, фиксируем такую структуру комплекса моделей, при которой оценка степени адекватности отношению порядка ЛФР будет максимальна. Функциональные взаимосвязи между комплексом математических моделей, консультируемой проблемой и блоком адаптации (рис. 4.35) отражают возможные методы настройки комплекса моделей на поведение реальной моделируемой системы.

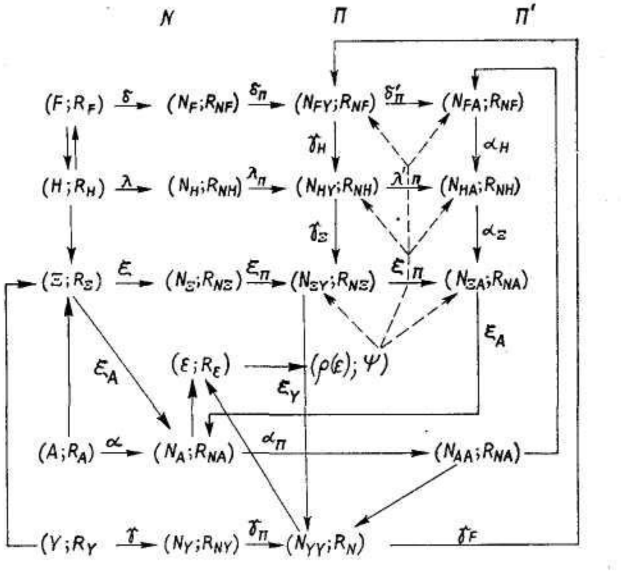

Рис. 4.35. Отношения между блоком адаптации и моделями *D, P, G* и *C.*

Пунктирные стрелки соответствуют алгоритмическим методам изменения параметров моделей ψ в зависимости от выбранного критерия адекватности ε и его числового значения ρ(ε).

4.8.8. Пример оценки формализованных правил выбора при построении моделей поведения ЛФР

Допустим, что ЛФР отобрало *n* из *m* *(n<m)* предъявленных предметов (альтернатив), причем характеристики каждой альтернативы были известны. Исходные данные представлены в табл. 4.35; оценки факторов нормированы по принципу, изложенному в п.4.4.3. В нашем примере ЛФР отобрало 8 альтернатив из 12. Последовательность альтернатив неизвестна.

Таблица 4.35

Исходные данные примера исследования формализованных правил выбора

Альтернативы	Факторы					Ранг, присвоенный ЛПР	Уточненный ранг
	A	*B*	*C*	*D*	*E*		
A_1	1	0	0,6	0,8	1	1—8	4,5
A_2	0,8	1	0,2	0	1	1—8	4,5
A_3	1	0	0,6	0	1	1—8	4,5
A_4	0,4	0,8	1	1	0	1—8	4,5
A_5	1	0	0	0,8	1	1—8	4,5
A_6	0,2	0,6	1	0	0	1—8	4,5
A_7	0	1	0	1	0	1—8	4,5
A_8	0,4	0	0,2	1	1	1—8	4,5
A_9	0	0,2	0,6	0	1	9—12	10,5
A_{10}	0	1	0	0,4	1	9—12	10,5
A_{11}	1	0,4	0	0	0,8	9—12	10,5
A_{12}	0,6	1	0	0,2	0,4	9—12	10,5

В табл. 4.36 приведены результаты применения различных формализованных правил выбора при условии равнозначности всех факторов, характеризующих альтернативы.

Как видно из табл. 4.36, наиболее адекватен поведению ЛФР принцип относительной уступки: ρ(ε)=0,51. Исследуем возможность увеличения степени адекватности ρ(ε) путем изменения весовых коэффициентов факторов. В табл. 4.37 показано влияние каждого из факторов на отношение порядка r^2, назначенное ЛФР.

Исключим из исходных данных (см. табл. 4.35) факторы *B* и *E* согласно выводам п. 4.7..7. Определим весовые коэффициенты факторов *A, C, D* как отношение значений ρ (*r, A*): ρ(r^2, *D*)*:* ρ(r^2, *C*)*,* т. е. ρ(r^2, *C*) : ρ(r^2, *D*)*:* ρ(r^2, *A*) =0,36:0,32:0,21=0,40:0,36:0,24.

Таблица 4.36

Степень адекватности формализованных правил выбора поведению ЛФР (исходные данные — в табл. 4.35)

Альтернативы	Ранги, уточненные по принципам				Ранг, присвоенный ЛФР
	максимина (Вальда)	Гурвица	последоват. максимина	относит. уступки	
A_1	6,5	6,5	1	1	4,5
A_2	6,5	6,5	4	3	4,5
A_3	6,5	6,5	7	5,5	4,5
A_4	6,5	6,5	2	2	4,5
A_5	6,5	6,5	6	4	4,5
A_6	6,5	6,5	10,5	10,5	4,5
A_7	6,5	6,5	12	9	4,5
A_8	6,5	6,5	4	5,5	4,5
A_9	6,5	6,5	10,5	10,5	10,5
A_{10}	6,5	6,5	8,5	12,0	10,5
A_{11}	6,5	6,5	8,5	7,5	10,5
A_{12}	6,5	6,5	4	7,5	10,5
Значения $\rho(\varepsilon)$	0,2	0,2	0,26	0,51	1
Уровень значимости по критерию Стьюдента	0,50	0,50	0,50	0,05	—

В табл. 4.38 приведены результаты расчета степени адекватности принципа относительной уступки поведению ЛФР при разных значениях весовых коэффициентов факторов.

Результаты расчета показывают, что при исходных данных, приведенных в табл. 4.35, наиболее адекватным поведению ЛФР будет принцип относительной уступки:

$$\Xi_1 = \max \sum_{i=1}^{k} \alpha_i s_i$$

при K=3 и весовом векторе $\alpha(C)$ =0,40; $\alpha(D)$=0,36; α (A) = 0,24. Степень адекватности этого принципа поведению ЛФР достигает значения 0,70. Решающее правило для данного примера можно записать следующим образом:

$$\Xi=0{,}40\ C+0{,}36D + 0{,}24\ a,$$

где C, D, A — нормированные оценки факторов альтернатив. Следует отметить, что при формализации решающего правила использовалась минимальная информация о выявленных предпочтениях ЛФР. Зная результат выбора ЛФР, возможно установить отношение порядка на множестве факторов и определить числовые характеристики весового вектора алгоритмическим путем.

Таблица 4.37

Степень адекватности факторов альтернатив поведению ЛФР

Альтернативы	Уточненные ранги по факторам					Ранг, присвоенный ЛФР
	A	B	C	D	E	
A_1	2,5	10,5	4	4,5	4	4,5
A_2	5	2,5	6,5	10	4	4,5
A_3	2,5	10,5	4	10	4	4,5
A_4	7,5	5	1,5	2	11	4,5
A_5	2,5	10,5	10	4,5	4	4,5
A_6	9	6	1,5	10	11	4,5
A_7	11	2,5	10	2	11	4,5
A_8	7,5	10,5	6,5	2	4	4,5
A_9	11	8	4	10	4	10,5
A_{10}	11	2,5	10	6	4	10,5
A_{11}	2,5	7,5	10	10	8	10,5
A_{12}	6	1,5	10	7	9	10,5
Значения $\rho(r^2, A)$	+0,21	−0,21	+0,36	+0,32	−0,01	1
Уровень значимости по критерию Стьюдента	0,50	0,50	0,50	0,50	0,05	—

Таблица 4.38

Степень адекватности принципа относительной уступки поведению ЛФР при разных значениях весовых коэффициентов факторов

Альтернативы	Уточненные ранги при весовых векторах				Ранг, присвоенный ЛФР
	0,33 0,33 0,33	0,40 0,36 0,24	0,50 0,33 0,17	0,17 0,33 0,50	
A_1	1,5	2	2	1	4,5
A_2	8	8	9	7	4,5
A_3	4,5	5	5	4	4,5
A_4	1,5	1	1	3	4,5
A_5	3	4	6	2	4,5
A_6	6	6	3	10	4,5
A_7	8	7	7	9	4,5
A_8	4,5	3	4	5	4,5
A_9	11	9,5	8	12	10,5
A_{10}	12	12	12	11	10,5
A_{11}	8	9,5	10	6	10,5
A_{12}	10	11	11	8	10,5
Значения $\rho(\varepsilon)$	0,66	0,70	0,65	0,49	1
Уровень значимости по критерию Стьюдента	0,02	0,02	0,05	0,20	—

4.9. Пример практической оценки сформированных рекомендаций с использованием САК

4.9.1. Пример анализа рекомендаций по оценке деятельности подразделений министерства в интерактивном режиме

Рассмотрим пример практической реализации диалога с САК при анализе деятельности подразделений министерства. Вопросы, выведенные на монитор, отмечены символом *.

* ПАКЕТ СЕРВИСНЫХ ПРОГРАММ И БАНК ДАННЫХ МИНИСТЕРСТВА К ВАШИМ УСЛУГАМ. ВЫ ХОТИТЕ ПОЛУЧИТЬ

ИНФОРМАЦИЮ, НАХОДЯЩУЮСЯ В БАНКЕ ДАННЫХ, ИЛИ ВВЕСТИ ИСХОДНУЮ ИНФОРМАЦИЮ С ПУЛЬТА?

— ИЗ БАНКА.

* ИТАК, ВЫ ХОТИТЕ РАБОТАТЬ НА ОСНОВЕ БАНКА ДАННЫХ?

— ДА.

* БАНК ДАННЫХ МИНИСТЕРСТВА К ВАШИМ УСЛУГАМ. КАКАЯ ИНФОРМАЦИЯ ВАС ИНТЕРЕСУЕТ: ПО ТРЕСТАМ ИЛИ ЗАКАЗЧИКАМ?

— ПО ТРЕСТАМ.

* ТАКИМ ОБРАЗОМ, ВЫ ХОТИТЕ ПОЛУЧИТЬ ИНФОРМАЦИЮ ПО ТРЕСТАМ?

— ДА.

* КАКИЕ ТРЕСТЫ ВАС ИНТЕРЕСУЮТ?

— ЛТОТ, РТОТ, ДОСТ, РОСТ, ЦОСТ.

* В БАНКЕ ДАННЫХ МИНИСТЕРСТВА ИМЕЮТСЯ ДАННЫЕ ПО ОБЪЕКТАМ С НАИМЕНОВАНИЯМИ: ЛТОТ, РТОТ, ДОСТ, РОСТ, ЦОСТ. ПОДТВЕРДИТЕ СВОЕ НАМЕРЕНИЕ ПОЛУЧИТЬ ЭТУ ИНФОРМАЦИЮ (ДА, НЕТ).

— ДА.

* ВАС ИНТЕРЕСУЮТ ВСЕ ПОКАЗАТЕЛИ ИЛИ ТОЛЬКО НЕКОТОРЫЕ ИЗ НИХ?

— НЕКОТОРЫЕ.

* ВВОДИТЕ ИНТЕРЕСУЮЩИЕ ВАС ПОКАЗАТЕЛИ. ВВОД ЗАКАНЧИВАЕТСЯ ПОСЛЕ ДИРЕКТИВЫ «КОНЕЦ». ПОСТАРАЙТЕСЬ НАЗЫВАТЬ УКРУПНЕННЫЕ ПОКАЗАТЕЛИ ТИПА «ОБЪЕМ РАБОТ, ПРОИЗВЕДЕННЫХ СОБСТВЕННЫМИ СИЛАМИ», «ИСПОЛЬЗОВАНИЕ ОСНОВНЫХ СТРОИТЕЛЬНЫХ МАШИН» И Т. Д. В ДАЛЬНЕЙШЕМ ВАМ БУДУТ ЗАДАНЫ УТОЧНЯЮЩИЕ ВОПРОСЫ.

— ЧИСЛЕННОСТЬ РАБОЧИХ.

* ПРОДОЛЖАЙТЕ ВВОД ИНТЕРЕСУЮЩИХ ВАС ПОКАЗАТЕЛЕЙ (ОН ЗАКАНЧИВАЕТСЯ ПО ДИРЕКТИВЕ «КОНЕЦ»).

— ТЕКУЧЕСТЬ РАБОЧИХ.

* ЧТО ИМЕННО ИЗ ИМЕЮЩЕГОСЯ В БАНКЕ ДАННЫХ ВАС ИНТЕРЕСУЕТ? ВВЕДИТЕ ЧЕРЕЗ ЗАПЯТУЮ ПОРЯДКОВЫЕ НОМЕРА:

1. БАЛАНС.
2. КОЛИЧЕСТВО УВОЛЕННЫХ РАБОЧИХ.

3. КОЛИЧЕСТВО РАБОЧИХ, УВОЛЕННЫХ ПО СОБСТВЕННОМУ ЖЕЛАНИЮ.

4. КОЛИЧЕСТВО РАБОЧИХ, УВОЛЕННЫХ ЗА НАРУШЕНИЕ ТРУДОВОЙ ДИСЦИПЛИНЫ.

5. ТЕКУЧЕСТЬ ЗА ТЕКУЩИЙ ГОД.

6. ТЕКУЧЕСТЬ ЗА ПРЕДШЕСТВУЮЩИЙ ГОД.

— 5.

* ПРОДОЛЖАЙТЕ ВВОД ИНТЕРЕСУЮЩИХ ВАС ПОКАЗАТЕЛЕЙ (ОН ЗАКАНЧИВАЕТСЯ ПО ДИРЕКТИВЕ «КОНЕЦ»).

— ВЫРАБОТКА НА ОДНОГО РАБОЧЕГО.

* ЧТО ИМЕННО ИЗ ИМЕЮЩЕГОСЯ В БАНКЕ ДАННЫХ ВАС ИНТЕРЕСУЕТ? ВВЕДИТЕ ЧЕРЕЗ ЗАПЯТУЮ ПОРЯДКОВЫЕ НОМЕРА:

1. ПЛАНОВАЯ ВЫРАБОТКА.
2. ФАКТИЧЕСКАЯ ВЫРАБОТКА.
3. ВЫПОЛНЕНИЕ ПЛАНА (%).
4. СРАВНЕНИЕ С ПРЕДШЕСТВУЮЩИМ ГОДОМ (%).

— 3.

* ПРОДОЛЖАЙТЕ ВВОД ИНТЕРЕСУЮЩИХ ВАС ПОКАЗАТЕЛЕЙ (ОН ЗАКАНЧИВАЕТСЯ ПО ДИРЕКТИВЕ «КОНЕЦ»).

— ОБЪЕМ РАБОТ, ПРОИЗВЕДЕННЫХ СОБСТВЕННЫМИ СИЛАМИ.

* ЧТО ИМЕННО ИЗ ИМЕЮЩЕГОСЯ В БАНКЕ ДАННЫХ ВАС ИНТЕРЕСУЕТ? ВВЕДИТЕ ЧЕРЕЗ ЗАПЯТУЮ ПОРЯДКОВЫЕ НОМЕРА:

1. ПЛАНОВЫЙ ОБЪЕМ (ТЫС. ГРН.).
2. ФАКТИЧЕСКИЙ ОБЪЕМ (ТЫС. ГРН.).
3. ВЫПОЛНЕНИЕ ПЛАНА (%).
4. ТЕМПЫ РОСТА К СООТВЕТСТВУЮЩЕМУ ПЕРИОДУ ПРЕДШЕСТВУЮЩЕГО ГОДА (%).

— 3.

* ИЗ ПРЕДЛАГАЕМОГО СПИСКА ПОКАЗАТЕЛЕЙ УКАЖИТЕ ОДИН, САМЫЙ ВАЖНЫЙ ДЛЯ ВАС ПРИ РЕШЕНИИ ВАШЕЙ ЗАДАЧИ.

1. ТЕКУЧЕСТЬ (%) ЗА ТЕКУЩИЙ ГОД.
2. ВЫПОЛНЕНИЕ ПЛАНА (%) ОБЪЕМА РАБОТ, ПРОИЗВЕДЕННЫХ СОБСТВЕННЫМИ СИЛАМИ.
3. ЧИСЛЕННОСТЬ РАБОЧИХ ЗА ТЕКУЩИЙ ГОД.

4. ВЫПОЛНЕНИЕ ПЛАНА ВЫРАБОТКИ (%) НА ОДНОГО РАБОТАЮЩЕГО.

— 2.

* ИЗ ПРЕДЛАГАЕМОГО СПИСКА ОСТАВШИХСЯ ПОКАЗАТЕЛЕЙ УКАЖИТЕ ОДИН, САМЫЙ ВАЖНЫЙ ДЛЯ ВАС ПРИ РЕШЕНИИ ВАШЕЙ ЗАДАЧИ

1. ТЕКУЧЕСТЬ (%) ЗА ТЕКУЩИЙ ГОД.

3. ЧИСЛЕННОСТЬ РАБОЧИХ ЗА ТЕКУЩИЙ ГОД.

4. ВЫПОЛНЕНИЕ ПЛАНА ВЫРАБОТКИ (%) НА ОДНОГО РАБОТАЮЩЕГО.

— 4.

* ИЗ ПРЕДЛАГАЕМОГО СПИСКА ОСТАВШИХСЯ ПОКАЗАТЕЛЕЙ УКАЖИТЕ ОДИН, САМЫЙ ВАЖНЫЙ ДЛЯ ВАС ПРИ РЕШЕНИИ ВАШЕЙ ЗАДАЧИ

1. ТЕКУЧЕСТЬ (%) ЗА ТЕКУЩИЙ ГОД.

3. ЧИСЛЕННОСТЬ РАБОЧИХ ЗА ТЕКУЩИЙ ГОД.

— 3.

* ПО РЕЗУЛЬТАТАМ ВАШИХ ОТВЕТОВ В ДАЛЬНЕЙШЕМ БУДУТ ПРИСВОЕНЫ СЛЕДУЮЩИЕ ВЕСОВЫЕ КОЭФФИЦИЕНТЫ:

1. ВЫПОЛНЕНИЕ ПЛАНА (%) ОБЪЕМА РАБОТ, ПРОИЗВЕДЕННЫХ СОБСТВЕННЫМИ СИЛАМИ — 0,4.

2. ВЫПОЛНЕНИЕ ПЛАНА ВЫРАБОТКИ (%) НА ОДНОГО РАБОТАЮЩЕГО — 0,3.

3. ЧИСЛЕННОСТЬ РАБОТАЮЩИХ ЗА ТЕКУЩИЙ ГОД — 0,2.

4. ТЕКУЧЕСТЬ (%) ЗА ТЕКУЩИЙ ГОД — 0,1.

* ВВЕДИТЕ НОМЕРА ПОКАЗАТЕЛЕЙ, ЧИСЛЕННОЕ ЗНАЧЕНИЕ КОТОРЫХ ТРЕБУЕТСЯ МИНИМИЗИРОВАТЬ.

— 4.

* ВВЕДИТЕ ДЛЯ КАЖДОГО УРОВНЯ ПОКАЗАТЕЛЯ «ВЫПОЛНЕНИЕ ПЛАНА (%) ОБЪЕМА РАБОТ, ПРОИЗВЕДЕННЫХ СОБСТВЕННЫМИ СИЛАМИ» ЧИСЛО ИЗ ИНТЕРВАЛА 0—100, ПРИЧЕМ САМОМУ ЖЕЛАТЕЛЬНОМУ ПРИПИШИТЕ ЦИФРУ 100, А В ПРОТИВНОМ СЛУЧАЕ — 0.

1. 100 — 100.

2. 90 —

3. 80 —

4. 70 — 0.

— 10, 1.

* ВВЕДИТЕ ДЛЯ КАЖДОГО УРОВНЯ ПОКАЗАТЕЛЯ «ВЫПОЛНЕНИЕ ПЛАНА ВЫРАБОТКИ (%) НА ОДНОГО

РАБОТАЮЩЕГО» ЧИСЛО ИЗ ИНТЕРВАЛА 0—100, ПРИЧЕМ САМОМУ ЖЕЛАТЕЛЬНОМУ ПРИПИШИТЕ ЦИФРУ 100, А В ПРОТИВНОМ СЛУЧАЕ — 0.

1. 100 — 100.
2. 95 —
3. 90 — 0.

— 98.

* ВВЕДИТЕ ДЛЯ КАЖДОГО УРОВНЯ ПОКАЗАТЕЛЯ ЧИСЛЕННОСТИ РАБОЧИХ ЗА ТЕКУЩИЙ ГОД ЧИСЛО ИЗ ИНТЕРВАЛА 0—100, ПРИЧЕМ САМОМУ ЖЕЛАТЕЛЬНОМУ ПРИСВОЙТЕ ЦИФРУ 100, А В ПРОТИВНОМ СЛУЧАЕ — 0.

1. 1700 - 100.
2. 1400 —
3. 1100 —
4. 800 — 0.

— 90, 70.

* ВВЕДИТЕ ДЛЯ КАЖДОГО УРОВНЯ ПОКАЗАТЕЛЯ «ТЕКУЧЕСТЬ (%) ЗА ТЕКУЩИЙ ГОД» ЧИСЛО ИЗ ИНТЕРВАЛА 0—100, ПРИЧЕМ САМОМУ ЖЕЛАТЕЛЬНОМУ ПРИПИШИТЕ ЦИФРУ 100, А В ПРОТИВНОМ СЛУЧАЕ — 0.

1. 0 — 100.
2. 10 —
3. 20 —
4. 30 —
5. 40 —
6. 50 — 0.

— 90, 60, 40, 20.

* С УЧЕТОМ ВАШИХ ОТВЕТОВ ИНТЕРЕСУЮЩАЯ ВАС ИНФОРМАЦИЯ МОЖЕТ БЫТЬ УПОРЯДОЧЕНА СЛЕДУЮЩИМ ОБРАЗОМ (табл. 4.39, 4.40):

Таблица 4.39

Упорядочивание трестов по показателям

№ п.п.	Наименование	Объем работ (%) (0,4)	Выработка (%) (0,3)	Численность работников (0,2)	Текучесть (0,1)	Суммарная полезность
1.	РОСТ	93,1	101,7	1291	9,2	70,32
2.	РТОТ	74,7	97,1	1601	39,4	42,52
3.	ЛТОТ	80,0	95,9	1163	46,3	33,31
4.	ДОСТ	84,6	92,2	1487	25,4	31,98
5.	ЦОСТ	75,1	91,3	915	26,9	14,21

Таблица 4.40

*** Результаты деятельности трестов в полезностных шкалах**

№ п.п.	Наименование	Объем работ (усл. ед.)	Выработка (усл. ед.)	Численность работников (усл. ед.)	Текучесть (усл. ед.)	Суммарная полезность
1.	РОСТ	15,16	29,502	16,54	9,12	70,32
2.	РТОТ	0,188	20,87	19,34	2,12	42,52
3.	ЛТОТ	0,40	17,35	14,84	0,72	33,31
4.	ДОСТ	2,056	6,468	18,54	4,92	31,98
5.	ЦОСТ	0,204	3,822	5,56	4,62	14,21

* ВАС ИНТЕРЕСУЮТ ОЦЕНКИ ОБЪЕКТОВ В ПОЛЕЗНОСТНЫХ ШКАЛАХ?

- ДА.

* НУЖНА ЛИ ВАМ РАСПЕЧАТКА ИНТЕРЕСУЮЩЕЙ ВАС ИНФОРМАЦИИ?

- ДА.

* ХОТИТЕ ЛИ ВЫ ПРОДОЛЖИТЬ РАБОТУ С БАНКОМ ДАННЫХ МИНИСТЕРСТВА?

— НЕТ.

* СИСТЕМА ЗАКОНЧИЛА РАБОТУ.

4.9.2. Описание алгоритмов программ для оценки сформированных рекомендаций в интерактивном режиме

Приводятся следующие разработанные и практически реализованные алгоритмы программ для оценки сформированных рекомендаций с помощью САК:

— формализации цели консультирования при вводе данных с пульта САК;

— формализации цели консультирования с учетом базы данных в САК;

— критериального оценивания состояния консультируемой проблемы при вводе данных с пульта САК;

— отображения качественных оценок консультируемой проблемы в количественные;

— критериального оценивания состояния консультируемой проблемы с учетом базы данных в САК;

— экспертного оценивания числовых значений вектора приоритета;

— построения функции полезности методом непосредственной оценки;

— ранжирования уровней фактора;

— построения функции полезности методом вероятностной средней точки;

— построения функции полезности методом вероятностно-упорядоченной метрики;

— формирования вектора весов интервалов;

— построения функции полезности методом ранжирования по бинарным факторам;

— нормализации факторов состояния;

— правила выбора;

— расчета глобальной оценки;

— оценивания степени адекватности модели;

— адаптации модели к поведению ЛФР.

Алгоритм формализации цели консультирования при вводе данных с пульта САК

Такт 1. Формализация цели консультирования.

Шаг 1. Запросить наименование цели N_{HB}.

Шаг 2. Ввести наименование цели N_{HB} (текстом до 80 символов).

Шаг 3. Запросить множество объектов F.

Шаг 4. Ввести объекты $f_i \in F; A_i$=1, 2, …, M.

Шаг 5. Запросить множество факторов Y.

Шаг 6. Ввести факторы $y \in Y$; A_i=1, 2, …, N.

Такт 2. Подключение алгоритма оценивания.

Такт 3. Запрос о необходимости перехода к такту 1.

Алгоритм формализации цели консультирования с учетом базы данных в САК

Такт 1. Формализация наименования цели консультирования.

Шаг 1. Запросить наименование цели консультирования N_{HB}.

Шаг 2. Ввести наименование цели N_{HB} (текстом до 80 символов).

Такт 2. Отображение множества объектов F.

Шаг 3. Запросить множество F.

Шаг 4. Ввести $f_i \in F$.

Шаг 5. Проверить: $f_i \in F_0$.

Если да, то перейти к шагу 8.

Если нет, то перейти к шагу 6.

Шаг 6. Сообщить о том, что f_i не воспринят.

Шаг 7. Положить $i = i — 1$.
Шаг 8. Если множество F введено, то перейти к шагу 9.
Если нет, то положить $i = i+1$ и перейти к шагу 4.
Такт 3. Отображение множества Y.
Шаг 9. Запросить множество Y.
Шаг 10. Ввести $y_i \in Y$.
Шаг 11. Проверить: $y_i \in Y_0$.
Если да, то перейти к шагу 14.
Если нет, то перейти к шагу 12.
Шаг 12. Сообщить о том, что y_j не найден.
Шаг 13. Положить $j = j —1$.
Шаг 14. Если Y введено, то перейти к шагу 15.
Если нет, то положить $j = j +1$ и перейти к шагу 10.
Шаг 15. Окончание работы алгоритма.

Алгоритм критериального оценивания состояния A^0 при вводе данных с пульта САК

Такт 1. Ввод сообщения пользователя.
Шаг 1. Указать пользователю объект $f_i \in F$.
Шаг 2. Указать пользователю фактор $y_j \in Y$.
Шаг 3. Ввести оценку f_i по y_j.
Такт 2. Ввод оценки в память.
Шаг 4. Если введенная оценка — число, то ввести ее в память и перейти к шагу 6. Если нет, то перейти к шагу 5.
Шаг 5. Перевести качественную оценку в числовую в соответствии со шкалой и ввести ее в память.
Такт 3. Проверка на законченность ввода оценок.
Шаг 6. Проверить: все ли y_j введены для данного f_i.
Если да, то перейтда к шагу 7.
Если нет, то положить $j = j +1$ и перейти к шагу 1.
Шаг 7. Проверить: все ли $f_i \in F_0$.
Если да, то перейти к шагу 8.
Если нет, то положить $i = i+1$ и перейти к шагу 1.
Шаг 8. Окончание работы алгоритма.

Алгоритм отображения качественных оценок консультируемой проблемы в количественные

Такт 1. Вызов из памяти эталонов качественных оценок.

Шаг 1. Прочитать с внешней памяти эталон $N_B^{(i)}$.
Шаг 2. Записать эталон $N_B^{(i)}$ в i-ю ячейку памяти.
Такт 2. Сравнение с эталоном.
Шаг 3. Сравнить оценку с эталоном $N_B^{(i)}$
Шаг 4. Если они эквивалентны, то перейти к шагу 5.
Если нет, то положить $i = i+1$ и перейти к шагу 3.
Такт 3. Перекодировка качественной оценки в числовую.
Шаг 5. Вызвать числовой эталон $N_Y^{(i)}$ (соответствующий $N_B^{(i)}$).
Шаг 6. Присвоить оценке значение числового эталона $N_Y^{(i)}$.
Шаг 7. Окончание работы алгоритма.

Алгоритм критериального оценивания состояния консультируемой проблемы с учетом базы данных в САК

Такт 1. Расчет γ_i, $\forall\, i=1,2,\ldots, N$.
Шаг 1. Выбрать из базы данных информацию для f_i по $\forall\, j=1,2,\ldots,M$.
Шаг 2. Просмотреть информацию и присвоить значение функции δ_i^j.
Шаг 3. Определить γ_i по формуле

$$\gamma_i = \frac{\Sigma_j \delta_i{}^j}{m},$$

где m — число консультируемых проблем; $j = 1, 2,\ldots, m$;

$$\delta_i^j = \begin{cases} c_i^j, \text{ если } c_i^j \text{ присутствует в базе данных}; \\ 0, \text{ если отсутствует.} \end{cases}$$

Такт 2. Формирование векторов c^j.
Шаг 4. Вызвать из базы данных значение оценки f_j по y_i-y_{ji}.
Шаг 5. Если оно существует, то положить $c_i{}^j = v_{ij}$.
Если нет, то положить $c_i{}^j = \gamma_i$.
Шаг 6. Если все c^j сформированы, то перейти к шагу 7.
Если нет, то перейти к шагу 4.
Шаг 7. Окончание работы алгоритма.

Алгоритм экспертного оценивания числовых значений вектора приоритета

Такт 1. Сравнение элементов множества Y.

Шаг 1. Проверить: y_i предпочтительнее y_j.

Если да, то перейти к шагу 2.

Если нет, то перейти к шагу 3.

В ином случае перейти к шагу 4.

Шаг 2. Положить k_i=k_i+1 и перейти к шагу 5.

Шаг 3. Положить $k_j = k_j$ +1 и перейти к шагу 5.

Шаг 4. Положить $k_i = k_i$ +0,5; $k_j = k_j$ + 0,5.

Такт 2. Проверка на окончание процедуры сравнения.

Шаг 5. Проверить: $j = N$.

Если да, то перейти к шагу 6.

Если нет, то положить j= j +1 и перейти к шагу 1.

Шаг 6. Проверить: $i = N$—1.

Если да, то перейти к шагу 7.

Если нет, то положить $i = i$+1 и перейти к шагу 1.

Шаг 7. Окончание работы алгоритма.

Алгоритм построения функции полезности методом непосредственной оценки

Такт 1. Ранжирование уровней.

Шаг 1. Передача управления алгоритму ранжирования уровней фактора и окончание работы алгоритма.

Алгоритм ранжирования уровней фактора

Такт 1. Выбор наиболее предпочтительного уровня.

Шаг 1. Указать пользователю все $y_{ij} \in y_i$.

Шаг 2. Ввести в память y_{ik}, где $y_{ik} > y_{ij}$ для любого j=1, 2,…, n

Шаг 3. Образовать множество y_i^N=$y_i \backslash y_{ik}$.

Такт 2. Проверка на законченность алгоритма.

Шаг 4. Проверить: y_i^N={Ø}.

Если да, то перейти к шагу 7.

Если нет, то перейти к шагу 5.

Шаг 5. Указать пользователю все $y_{ij} \in y_i^N$.

Шаг 6. Перейти к шагу 2.

Шаг 7. Окончание работы процедуры.

Такт 3. Оценивание уровней.

Шаг 8. Указать пользователю шкалу Ф.

Шаг 9. Указать пользователю y_{ij}.

Шаг 10. Ввести оценку n_j по y_{ij}.

Такт 4. Проверка на законченность ввода оценок.

Шаг 11. Проверить: все ли y_{ij} введены для данного y_i.

Если да, то перейти к шагу 12.

Если нет, то перейти к шагу 9.

Шаг 12. Окончание работы алгоритма.

Алгоритм построения функции полезности методом вероятностной средней точки

Такт 1. Построение массива оценок полезности U.

Шаг 1. Присвоить массиву U значения фиксированных уровней по шкале полезности.

Такт 2. Построение массива y.

Шаг 2. Ввести $y_{i\,\max}=y_i^+$.

Шаг 3. Ввести $y_{i\,\min}=y_i^-$.

Шаг 4. Ввести $y_{i\,\text{ср}}$ так, чтобы $U(y_{i\,\text{ср}})\approx(1/2)U(y_i^+)+U(y_i^-)$.

Шаг 5. Ввести $y_{i\,\text{ср}}^1$ так, чтобы $U(y_{i\,\text{ср}}^1)\approx(1/2)U(y_i^+)+U(y_{i\,\text{ср}}))$.

Шаг 6. Ввести $y_{i\,\text{ср}}^2$ так, чтобы $U(y_{i\,\text{ср}}^2)\approx(1/2)U(y_i^-)+U(y_{i\,\text{ср}}))$.

Такт 3. Ввод значений в память.

Шаг 7. Ввести в память массив Y.

Шаг 8. Ввести в память массив U.

Шаг 9. Окончание работы алгоритма.

Алгоритм построения функции полезности методом вероятностно-упорядоченной метрики

Такт 1. Ранжирование уровней.

Шаг1. Проверить, ранжированы ли уровни естественным путем.

Если да, то перейти к шагу 3.

Если нет, то перейти к шагу 2.

Шаг 2. Передать управление алгоритму ранжирования уровней.

Такт 2. Ранжирование интервалов.

Шаг 3. Предложить пользователю сравнить интервалы

$$(y_{ij} — y_{i(j+1)}) \text{ и } (y_{im} — y_{i(m+1)}).$$

Шаг 4. Ввести отношение предпочтения.

Шаг 5. Положить $m = m$+l.

Шаг 6. Если $m\geq N$, то перейти к шагу 7.

Если $m< N$, то перейти к шагу 3.

Шаг 7. Положить $j = j + 1$.

Шаг 8. Если $j\geq N—1$, то перейти к шагу 9.

Если $j<N—1$, то перейти к шагу 3.

Такт 3. Ввод значений в память.

Шаг 9. Присвоить веса интервалам (алгоритм см. ниже).

Шаг 10. Присвоить веса уровням.

Шаг 11. Ввести массив весов уровней в память.

Шаг 12. Окончание работы алгоритма.

Алгоритм формирования вектора весов интервалов

Такт 1. Сравнение интервалов между уровнями по данному критерию.

Шаг 1. Если интервал $s_i = (y_{ij} — y_{i(j+1)}) > s_k = (y_{im} — y_{i(m+1)})$, то перейти к шагу 2.
Если интервал $s_k > s_i$, то перейти к шагу 3.
В ином случае перейти к шагу 4.

Шаг 2. Положить $k_i = k_i+1$ и перейти к шагу 5.

Шаг 3. Положить $k_k = k_k+1$ и перейти к шагу 5.

Шаг 4. Положить $k_i = k_i + 0{,}5$; $k_k = k_k + 0{,}5$.

Такт 2. Проверка на окончание работы алгоритма.

Шаг 5. Если $j = N$, то перейти к шагу 6.
Если $j \neq N$, то положить $j = j +1$ и перейти к шагу 1.

Шаг 6. Если $i = N—1$, то перейти к шагу 7.
Если $i \neq N—1$, то положить $i = i+l$ и перейти к шагу 1.

Такт 3. Ввод значений в память.

Шаг 7. Ввести в память массив α_i, где $\alpha_i = k_i/\sum_i k_i$, $i=1, 2,\ldots, n$.

Шаг 8. Окончание работы алгоритма.

Алгоритм построения функции полезности методом ранжирования по бинарным факторам

Такт 1. Ранжирование уровней всех y_i при $i=1, 2,\ldots, n$.

Шаг 1. Положить $j =1$.

Шаг 2. Выбрать из памяти $y_i \in \{\alpha, \beta\}$.

Шаг 3. Передать управление алгоритму ранжирования уровней.

Шаг 4. Положить $j = j +1$.

Шаг 5. Если $j <n$, то перейти к шагу 2.
Если $j>n$, то перейти к шагу 6.

Такт 2. Ранжирование интервалов.

Шаг 6. Положить $j =1$.

Шаг 7. Положить $j_1 = j +1$.

Шаг 8. Предложить пользователю сравнить интервалы $(\alpha_j - \beta_j)$ и $(\alpha_{j1} - \beta_{j1})$.

Шаг 9. Ввести отношение предпочтения.

Шаг 10. Положить $j_1 = j_1 + 1$.

Шаг 11. Если $j_1 < n$, то перейти к шагу 8.
Если $j_1 > n$, то перейти к шагу 12.

Шаг 12. Положить $j = j + 1$.

Шаг 13. Если $j < n$, то перейти к шагу 7.
Если $j \geq n$, то перейти к шагу 14.

Такт 3. Ввод значений в память.

Шаг 14. Передать управление алгоритму весов.

Шаг 15. Присвоить веса уровням.

Шаг 16. Ввести массив весов уровней в память.

Шаг 17. Окончание работы алгоритма.

Алгоритм нормализации факторов состояния

Такт 1. Расчет формульных констант.

Шаг 1. Найти максимум среди оценок фактора y_i.

Шаг 2. Найти минимум среди оценок фактора y_i.

Шаг 3. Найти разность максимума и минимума оценок фактора y_i.

Такт 2. Нормализация.

Шаг 4. Нормализовать оценку f_i объекта по фактору y_i:

$$c'_j=(c_j- c_j^-)/(c_j^+- c_j^-),$$

где c_j — нормализуемая оценка j-го объекта по данному фактору; c_j^- — минимальная оценка по данному фактору; c_j^+ — максимальная оценка по данному фактору; c'_j — нормализованная оценка j-го объекта по данному фактору.

Шаг 5. Записать нормализованную оценку в память.

Такт 3. Проверка на окончание нормализации оценок данного фактора.

Шаг 6. Проверить, все ли $f_j \in F_0$.
Если да, то перейти к шагу 7.
Если нет, то положить $j=j+1$ и перейти к шагу 4.

Такт 4. Проверка на окончание нормализации оценок всех факторов.

Шаг 7. Проверить, все ли $y_i \in Y_0$.
Если да, то перейти к шагу 8.
Если нет, то положить $i = i+1$ и перейти к шагу 1.

Шаг 8. Окончание работы алгоритма.

Алгоритм правила выбора

Алгоритм использует аддитивное правило вычисления глобальной оценки объектов:

$$s_i = \Sigma'_j \alpha_j c'_{ij},$$

где s_i — глобальная оценка j-го объекта.

Алгоритм расчета глобальной оценки

Такт 1. Расчет глобальной оценки.

Шаг 1. Выбрать из памяти нормализованную оценку c'_{ij} объекта f_i по фактору y_j.

Шаг 2. Выбрать из памяти вес α_j фактора y_j.

Шаг 3. Добавить к глобальной оценке объекта f_i произведение c'_{ij} α_j

Такт 2. Проверка окончания расчета глобальной оценки f_i объекта.

Шаг 4. Проверить, все ли y_j введены.
Если да, то перейти к шагу 5.
Если нет, то положить $j = j+1$ и перейти к шагу 1.

Такт 3. Проверка на окончание работы алгоритма.

Шаг 5. Проверить, все ли f_i введены.
Если да, то перейти к шагу 6.
Если нет, то положить $i = i+1$ и перейти к шагу 1.

Шаг 6. Окончание работы алгоритма.

Алгоритм оценивания степени адекватности модели

Такт 1. Ввод множества Y_D.

Шаг 1. Положить i=l.

Шаг 2. Запросить $y_{id} \in Y_D$.

Шаг 3. Ввести $y_{id} \in Y_D$.

Шаг 4. Проверить $i=n$.
Если да, то перейти к шагу 7.
Если нет, то перейти к шагу 5.

Шаг 5. Положить $i = i+1$.

Шаг 6. Перейти к шагу 2.

Шаг 7. Закончить ввод множества Y_D.

Такт 2. Расчет коэффициента ρ.

Шаг 8. Положить $R = 0$.

Шаг 9. Положить i=l.

Шаг 10. Положить $R = R + (y_{id} - y_{im})^2$.

Шаг 11. Проверить: $i = n$.
Если да, то перейти к шагу 14.
Если нет, то перейти к шагу 12.
Шаг 12. Положить $i = i+1$.
Шаг 13. Перейти к шагу 10.
Шаг 14. Положить $\rho = 1\text{-}6R/n(n^2\text{-}1)$.
Шаг 15. Окончание работы алгоритма.

Алгоритм адаптации модели к поведению ЛФР

Такт 1. Перебор параметров y_{ij} модели Y_M ($y_{ij} \in Y_M$).
Шаг 1. Положить i=l.
Шаг 2. Включить n_i в модель.
Шаг 3. Положить j=1.
Шаг 4. Включить m_j в модель.
Шаг 5. Решить задачу на модели и получить Y_M.
Шаг 6. Определить ρ_{ij} по Y_M и Y_D.
Шаг 7. Проверить: $j = 3$.
Если да, то перейти к шагу 10.
Если нет, то перейти к шагу 8.
Шаг 8. Положить $j = j+1$.
Шаг 9. Перейти к шагу 4.
Шаг 10. Проверить: $j = 3$.
Если да, то перейти к шагу 13.
Если нет, то перейти к шагу 11.
Шаг 11. Положить $i = i+1$.
Шаг 12. Перейти к шагу 2.
Такт 2. Определение оптимальных параметров.
Шаг 13. Определить i_0, j_0 для ρ_{i0} при выполнении условия

$$j_0 = \max_{ij} \rho_{ij}.$$

Такт 3. Построение оптимальной модели.
Шаг 14. Включить в модель окончательно n_{i0} и m_{j0}.
Шаг 15. Окончание работы алгоритма.

Литература

1. European Directory of Management Consultants. 1995. London: FEACO-AP Information services, 1995.
2. Guidelines for the Use of Consultants by World Bank Borrowers and by the World Bank as Executing Agency. Washington (D.C.): The World Bank, 1992.
3. Hurley N. Management Consultancy Manual: Operating a Successful Management Consultancy Assignment. Ankara: SMIDO, 1990.
4. Kubr M. How to select and use consultants: A client's guide. Geneva: ILO, 1993.
5. Maister D. Professional Service Firm Management. Boston: Maister Associates. Inc., 1990.
6. Кудинов А. О рынке консалтинговых услуг. // www.bcg.ru.
7. Монахова Е. Управленческое консультирование конца XX века. // www.pcweek.ru/kis.
8. Посадский А.П., Хайниш С.В. Консультационные услуги в России. - М.: Финстатинформ, 1995, - 171 с.
9. Уткин Э.А. Консалтинг. – М.: ЭКМОС, 1998, - 256 с.
10. Интернет-сервер Гарвардской школы бизнеса: www.hbs.edu
11. BS7799: Информационная безопасность начинается с менеджмента. // Банковские технологии. №8, 1998, - С. 73-75.
12. Международный стандарт ИСО 9000. Системы менеджмента качества. Основные положения и словарь. 2-е изд. 2000-12-15. ISO - 2000.
13 Международный стандарт ИСО 9001. Системы менеджмента качества. Требования. 3-е изд. 2000-12-15. ISO – 2000.
14. Международный стандарт ИСО 9004. Системы менеджмента качества. Руководство по улучшению деятельности. 2-е изд. ISO – 2000.
15. ISO 9000 Introduction and Support Package: Guidelines on the Process Approach to quality management systems. ISO/TC 176/SC 2/N 544R. 17 May, 2001.
16. ISO 9000 Introduction and Support Package: Guidance on the Documentation Requirements of ISO 9001:2000. ISO/TC 176/SC 2/N 544R. 13 March, 2001.

17. Давид Марка, Клемент МакГоуэн. Методология структурного анализа и проектирования. Пер . с англ . М .:1993, 240 с ., ISBN 5-7395-0007-9

18. INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF0). Draft Federal Information Processing Standards Publication 183, 1993, December 2. www.idef.com

19. Р50.1.028-2001. Методология функционального моделирования. М.: Госстандарт России, 2000. www.cals.ru

20. Менеджмент качества и международные стандарты ИСО 9000 версии 2000 г. Материалы семинара в рамках Программы ИСО для развивающихся стран. Минск, Июль 2001 г. 79 с.

21. Framework for Managing Process Improvement. Vol.1. Electronic College of Process Innovation. DoD USA. May, 1994.

22. С. Бобровски - ORACLE 7 и вычисления клиент - сервер, Москва, "ЛОРИ", 1996.

23. М. Ричарде и др. - Oracle 7.3, Энциклопедия пользователя, Киев, "DiaSoft", 1997.

24. ORACLE7 SERVER - CONCEPT MANUAL.

25. ORACLE7 SERVER - SQL LANGUAGE REFERENCE MANUAL -Part Number 778-70-1292, December 1992.

26. С. Бемер, Г.Фратер - MS Access...для пользователя, Киев, "BHV", 1994.

27. Р. Дженнингс - Access 95 в подлинике, 2 тома, СПб, "BHV", 1997.

28. Р. Дженнингс - Microsoft Access 97 в подлинике, 2 тома, Санкт-Петербург, "BHV", 1997.

29. Керри Н. Праг, Мишель Р. Ирвин - Access 97, Библия пользователя, Киев-Москва, "Диалектика", 1997.

30. А. Горев, С. Макашаринов - Microsoft Visual FoxPro 3.0, Новые возможности для программиста, Санкт-Петербург, "Питер Пресс", 1995.

31. Б. Сосински - Разработка приложений в среде Visual FoxPro 5, Киев-Москва, "Диалектика", 1997.

32. Цыпкин Я. 3. Основы теории автоматических систем. — М.: Наука. Гл. ред. физ.-мат. лит., 1977. — 560 с.

33. Лукашин Ю. 77. Адаптивные методы краткосрочного прогнозирования. — М.: Статистика, 1979. — 254 с.

34. Марпл-мл. С. Л. Цифровой спектральный анализ и его приложения: Пер. с англ. — М.: Мир, 1990. — 584 с.

35. Дрейпер Н., Смит Г. Прикладной регрессионный анализ: В 2-х книгах: Пер. с англ. —М.: Финансы и статистика, 1986. —Кн. 1. — 366 с.

36. Дрейпер Н., Смит Г. Прикладной регрессионный анализ: В 2-х книгах: Пер. с англ. — М.: Финансы и статистика, 1987. — Кн. 2. — 351 с.

37. Бриллинджер Д. Временные ряды. Обработка данных и теория: Пер. с англ. — М.: Мир, 1980. — 536 с.

38. Ахметшин А. М., Киргизов И. А. Прогнозирование кратковременных экономических рядов как задача адаптивной экстраполяции функций с финитным спектром // Пращ М1жнар. конф. з индуктивного моделювання «МКIМ — 2002». — Львщ 2002. — С. 29—33.

39. *Клих Ю. А., Плотникова Л. И.* Спектральный анализ функций. — К.:УМКВО, 1992.— 108 с.

40. *Kasabov N. K.* Foundations of neural networks, fuzzy systems, and knowledge engineering. — Cambridge, Mass.: MIT Press, 1996. — 550 p.

41. *Беллман Р.* Введение в теорию матриц: Пер. с англ. — М.: Наука. Гл. ред. физ.-мат. лит., 1976. — 352 с.

Научно-практическое издание

Кононюк Анатолий Ефимович

Консалтология

Общая теория консалтинга

Книга 2

Авторская редакция

Подписано в печать 30.9.2010 г.
Формат 60х84/16.
Усл. печ. л. 16,5. Тираж 300 экз.

Издатель и изготовитель:
Издательство «Освита Украины»
04214, г. Киев, ул. Героев Днепра, 63, к. 40
Свидетельство о внесении в Государственный реестр
издателей ДК №1957 от 23.04.2009 г.
Тел./факс (044) 411-4397; 237-5992
E-mail: osvita2005@ukr.net, www.rambook.ru

структуры матрицы взаимодействия представляет собой элемент