

**РАДІОТЕХНІКА ТА ТЕЛЕКОМУНІКАЦІЇ**

УДК 621.311: 681.5

**М.Ф. Бондаренко, д.т.н., проф.,  
З.В. Дудар, к.т.н., проф.,  
М.В. Збітнєва, аспір.**

*Харківський національний університет радіоелектроніки*

**МОДЕЛІ ЕЛЕКТРИЧНИХ МЕРЕЖ  
І МЕТОДИ АВТОМАТИЧНОГО ФОРМУВАННЯ ЇХ ТОПОЛОГІЙ**

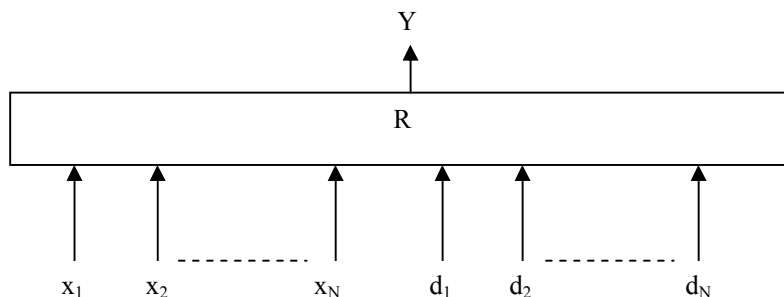
*Запропоновано моделі елементів електричних мереж, що базуються на теорії предикатів. Розроблено метод і алгоритм побудови орієнтованого симетричного графа, метод і алгоритм побудови топологічного графа електричної мережі, спосіб представлення графів.*

Сучасні тенденції розвитку автоматизованих систем диспетчерського управління (АСДУ) електричними мережами характеризуються підвищенням вимог до комплексів технічних засобів і програмного забезпечення диспетчерських пунктів. Застосування автоматизованих робочих місць, систем підтримки прийняття рішень, SCADA технологій дозволяє ефективно вирішувати задачі оперативного управління.

Серед завдань інформаційного забезпечення диспетчера багато уваги приділяється задачі графічного відображення стану контрольованих об'єктів (різного устаткування електричних станцій, підстанцій, ліній електропередач) [1]. З цією задачею пов'язаний комплекс питань, глибина вирішення яких визначає якісні показники управління електричними мережами. Це такі задачі, як виконання групової операції над ділянкою мережі за будь-якими вимогами, автоматична побудова схем заміщення [2], виявлення місця обриву електричної мережі та ін.

Для вирішення питання автоматичного формування топології електричної мережі була використана теорія предикатів [3] і розроблені наступні математичні моделі.

Елементи електричної мережі умовно розділені на дві групи. До першої групи віднесені елементи, що можуть бути описані моделлю (рис. 1), отриманою на базі предиката першого порядку R. До цієї групи відносяться елементи, що мають дві точки підключення одного класу напруги (вимикачі, роз'єднувачі, короткозамикачі та ін.).



*Рис. 1. Модель елемента електричної мережі:*

*$x_1, \dots, x_N$  – параметри-властивості;  $d_1, \dots, d_N$  – параметри-дії;  $y$  – параметр-результат*

До другої групи елементів віднесені трансформатори. Вони можуть бути подані предикатами R, зв'язаними між собою за правилами переходу до схем заміщення. У табл. 1 показані моделі силових трансформаторів.

Для опису структури підстанції пропонується чотирирівнева предикатна модель (П-модель), що враховує послідовність з'єднання елементів, їхнє взаємне розташування і властивості. Зображення П-моделі у вигляді орієнтованого графа наведено на рис. 2. Цей граф показує взаємне з'єднання елементів, гілок і вузлів електричної мережі. Будемо називати його топологічним.

П-модель складається з чотирьох типів предикатів R, P, Q, I. Кожний з них виконує операції над тією ділянкою мережі, за яку він відповідає. Предикат R охоплює один елемент, предикат P – два елементи, предикат Q – елементи гілки, предикат I – елементи вузла. Тип операцій відповідає розв'язуваній задачі і визначає формули та параметри предикатів.

*Таблиця 1*

<b>Двообмоточний трансформатор</b>	<b>Трансформатор з розщипленою обмоткою</b>	<b>Триобмоточний трансформатор</b>
------------------------------------	---	------------------------------------

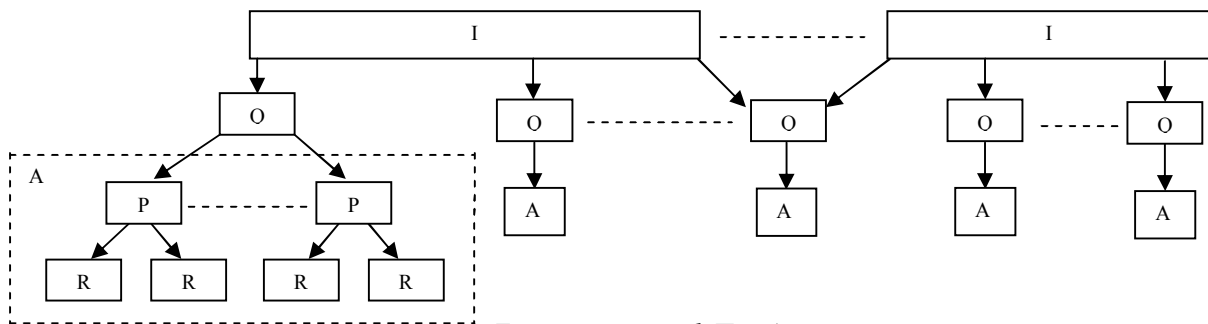
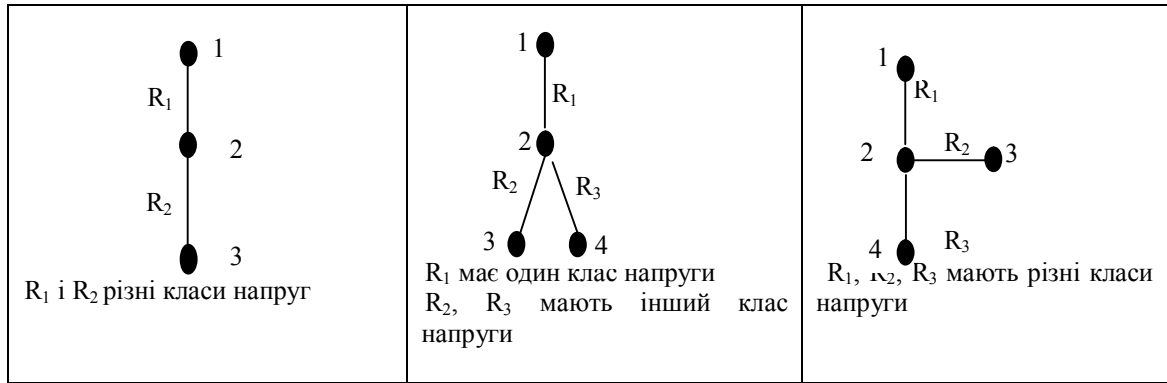


Рис. 2. Топологічний граф П-моделі

Розглянемо математичний опис П-моделі для задачі перевірки зв'язності електричної мережі.

Перший рівень П-моделі утворений предикатами першого порядку R, що визначають наявність розриву усередині елемента x<sub>i</sub> і мають наступний вигляд:

$$R(x_i) = \begin{cases} 1, & \text{всередині елемента } x_i \text{ немає розриву,} \\ 0, & \text{інакше.} \end{cases} \quad (1)$$

Область визначення R містить безліч станів елементів підстанції: M<sub>R</sub> = {a<sub>1</sub>, a<sub>2</sub>, a<sub>3</sub>, a<sub>4</sub>}, де a<sub>1</sub> – "Включено", a<sub>2</sub> – "Виключено", a<sub>3</sub> – "Справний", a<sub>4</sub> – "Несправний".

Область значень предиката R: N<sub>R</sub> = {0, 1}. Предикат R повертає одиницю, якщо x<sub>i</sub> = a<sub>1</sub> ∨ a<sub>3</sub>, і нуль, коли x<sub>i</sub> = a<sub>2</sub> ∨ a<sub>4</sub>.

На другому рівні П-моделі знаходиться предикат другого порядку P, що має вигляд:

$$P(R_i(x_i), R_j(x_j), C_{ij}) = \begin{cases} 1, & \text{якщо } R_i(x_i) \wedge R_j(x_j) \wedge C_{ij} = 1, \\ 0, & \text{інакше,} \end{cases} \quad (2)$$

де R<sub>i</sub>(x<sub>i</sub>), R<sub>j</sub>(x<sub>j</sub>) – предикати першого порядку, обумовлені співвідношенням (1);

C<sub>ij</sub> – змінна, що приймає значення, яке дорівнює одиниці за наявності зв'язку між елементами x<sub>i</sub> і x<sub>j</sub>, і нулю у протилежному разі;

i, j ∈ [1, ..., n], n – кількість елементів на підстанції.

Область визначення і область значень предиката P: M<sub>P</sub> = N<sub>P</sub> = {0, 1}.

Предикат P повертає одиницю у випадку наявності повної зв'язності на ділянці кола з двох елементів, тобто за умови, що R<sub>i</sub>(x<sub>i</sub>) ∧ R<sub>j</sub>(x<sub>j</sub>) ∧ C<sub>ij</sub> = 1.

На третьому рівні П-моделі знаходяться гілки, а на четвертому – вузли електричної мережі. У табл. 2 наведено відповідність термінів вузол і гілка термінам теорії графів і теорії предикатів.

Таблиця 2

Теорія електричних кіл	Теорія графів	Теорія предикатів
Вузол	Вершина	Елемент універсума
Гілка	Ребро	Відношення між елементами універсума

Для забезпечення необхідного формального опису вершин і ребер введені предикати гілки і вузла.

Предикат гілки Q має вигляд:

$$Q(P_1(R_i(x_i), R_j(x_j), C_{ij}), \dots, P_n(R_i(x_i), R_j(x_j), C_{ij})) = \begin{cases} 1, & P_1 \wedge \dots \wedge P_n = 1, \\ 0, & \text{інакше,} \end{cases} \quad (3)$$

де P<sub>1</sub>, ..., P<sub>n</sub> – предикати, обумовлені співвідношенням (2);

$n$  – кількість елементів на підстанції.

Область визначення і область значень  $Q$ :  $M_Q = N_Q = \{0,1\}$ .

Предикат  $Q$  повертає одиницю за умови наявності повної зв'язності на ділянці ланцюга, що відноситься до гілки, тобто якщо всі предикати  $P$ , розташовані на даній гілці, повертають одиницю.

Предикат вузла  $I$  поєднує гілки, з'єднані з одним вузлом, і має такий вигляд:

$$I(Q_1(P_1(R_1(x_1), R_j(x_j), C_{ij}), \dots, P_n(R_1(x_1), R_j(x_j), C_{ij})), \dots, Q_n(P_1(R_1(x_1), R_j(x_j), C_{ij}), \dots, P_n(R_1(x_1), R_j(x_j), C_{ij}))), C_{1..n}) = \begin{cases} 1, & \text{якщо } Q_1 \wedge \dots \wedge Q_n \wedge C_{1..n} = 1, \\ 0, & \text{інакше,} \end{cases} \quad (4)$$

де  $Q_1, \dots, Q_n$  – предикати гілок, обумовлені співвідношенням (3);

$C_{1..n}$  – змінна, що приймає значення одиниці за наявності зв'язку в точці з'єднання гілок, і нуль – у протилежному разі;

$n$  – кількість гілок на підстанції.

Область визначення і область значень  $I$ :  $M_I = N_I = \{0,1\}$ .

Предикат вузла повертає одиницю у випадку відсутності обривів на ділянці кола, що відповідає вузлу, тобто за умови  $Q_1 \wedge \dots \wedge Q_n \wedge C_{1..n} = 1$ .

Для формального опису підстанції в цілому, як єдиного блоку, недостатньо диз'юнктивно з'єднати предикати вузлів. На електричній схемі підстанції перехід від одного вузла до іншого здійснюється за допомогою суміжної гілки. Виходячи з цього, матимемо таке співвідношення, що описує схему підстанції:

$$Q_1(I_i \wedge I_j) \vee Q_2(I_i \wedge I_j) \vee Q_3(I_i \wedge I_j) \vee Q_4(I_i \wedge I_j) \dots Q_n(I_i \wedge I_j), \quad (5)$$

де  $i, j \in [1..n]$ ;  $n$  – кількість вузлів на підстанції.

Порядок з'єднання предикатів  $R, P, Q, I$  порушує предикат шини  $R_{ш.}$ . У зв'язку з тим, що на схемах заміщення цей елемент є вузлом, то вершина топологічного графа, яку представляє  $R_{ш.}$ , виходить ізольованою. Щоб не порушувати зв'язність, необхідно підключати  $R_{ш.}$  до відповідного вузла  $I$  (рис. 3).

Вибір відповідної структури даних для представлення топологічного графа  $\Pi$ -моделі впливає на ефективність алгоритмів на її основі. Зупинимося на цій проблемі детальніше.

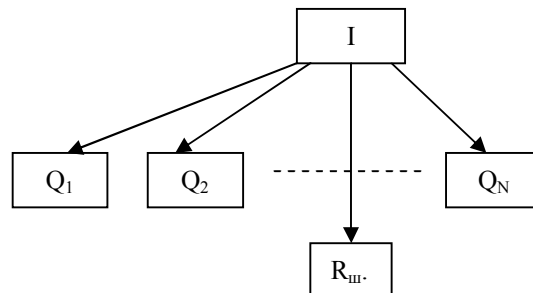


Рис. 3. Підключення предиката шини

Існують різні способи представлення графів у пам'яті ЕОМ, ефективність яких оцінюється за двома критеріями: обсягом займаної пам'яті і швидкістю виконання операцій над графами. Найчастіше використовуються матриця суміжності, матриця інцидентів, списки суміжності і масив дуг [4, 5, 6].

Спосіб представлення графів вибирається, виходячи із завдань конкретної задачі. Для задач диспетчерського управління критерій швидкості виконання операцій більш важливий, ніж обсяг потрібної пам'яті. Тестові іспити показали, що за швидкістю списки суміжності дають менший сумарний час. А для зменшення обсягу пам'яті пропонується спосіб представлення графів, названий нами багатозв'язним списком суміжності.

Багатозв'язний список суміжності – це спосіб представлення графа за допомогою спискової структури, яка відображає суміжність вершин і складався з одномірного масиву покажчиків  $\Gamma$ :  $array[1..p]$  of  $\uparrow N$  на вершині графа, кожна з яких подана у вигляді структури  $N$ :  $node\ v:1..p, next: array[1..a]$  of  $\uparrow N\ endnode$ . ( $p$  – кількість вершин графа,  $a$  – кількість суміжних вершин вершини  $v$ )".

На рис. 5 наведено приклад багатозв'язного списку суміжності, складеного за неорієнтованим графом (рис. 4).

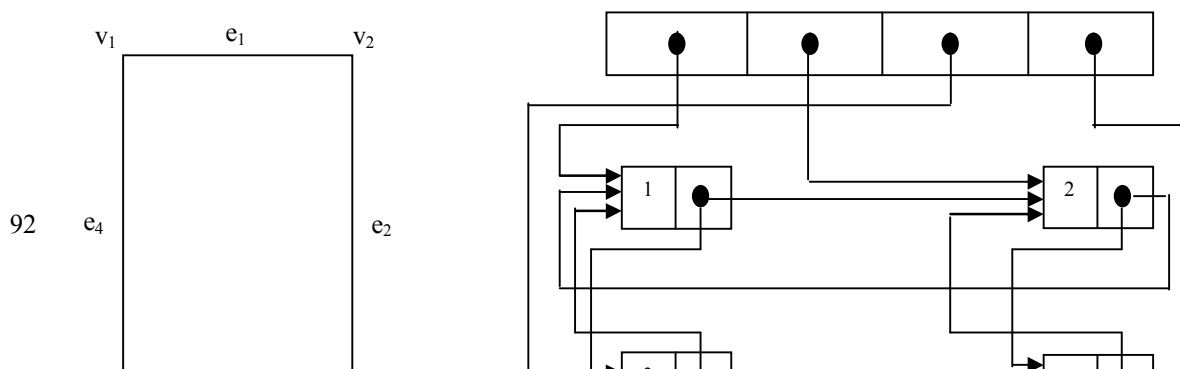


Рис. 4. Граф

Рис. 5. Багатозв'язний список суміжності

Порівняння існуючих представлень з багатозв'язним списком суміжності наведено в табл. 3.

Таблиця 3

Способи представлення графів

Спосіб представлення графа	Обсяг пам'яті для кожного представлення		Швидкість доступу до інформації (кількість кроків)	
	кількість ребер(E), вершин (V)	кількість покажчиків	чи існує ребро (дуга) з вершини x до вершини y?	до яких вершин ведуть ребра з вершини x?
<b>Матриця суміжності</b>	$V \times V$	0	1	V
<b>Матриця інцидентності</b>	$E \times V$	0	E	E
<b>Списки суміжності</b>	E – для орієнтованого графа, $2 \times E$ – для неорієнтованого графа	$V + E$ – для орієнтованого графа, $V + 2 \times E$ – для неорієнтованого графа	Кількість суміжних з x вершин	Кількість суміжних з x вершин
<b>Масив ребер (дуг)</b>	$2 \times E$	0	E	E
<b>Багатозв'язний список суміжності</b>	V	$V + E$ – для орієнтованого графа, $V + 2 \times E$ – для неорієнтованого графа	Кількість суміжних з x вершин	Кількість суміжних з x вершин

Формування предиката R П-моделі здійснюється користувачем у підсистемі побудови елементів, а формування предикатів P, Q, I і топологічного графа – у підсистемі побудови схем АСДУ.

Розглянемо методи, що описують автоматичну побудову топології мережі у вигляді П-моделі.

**Метод побудови орієнтованого симетричного графа**

Формування топологічного графа електричної мережі припускає наявність даних про зв'язки між елементами мережі. Для цього необхідно отримати орієнтований симетричний граф (ОС-граф) [7], симетричність якого зумовлена тим, що на етапі формування зв'язків не можна визначити, яка з двох вершин є початком, а яка – кінцем дуги. Спосіб визначення взаємозв'язку елементів усередині однієї підстанції полягає у визначенні зон можливих підключень елементів і використанні методів точного з'єднання [8]. Для того, щоб можна було з'єднувати разом кілька підстанцій, треба ввести властивість "Зовнішній зв'язок".

Метод формування ОС-графа полягає в наступному.

Вершинам ОС-графа ставляться у відповідність предикати R, що описують елементи електричної мережі. Протилежно направлені дуги формуються, якщо відповідні графічні образи елементів з'єднані один з одним.

Алгоритм формування ОС-графа має такий вигляд:

Вхідні дані: схема електричного кола.

Вихідні дані: орієнтований симетричний граф, поданий багатозв'язним списком суміжності.

Початок.

- 1) Make;
  - 2) For (для) кожної вершини  $v_i \in V, i = 1 \dots n$
  - 3)     DoFor (для) кожної вершини  $v_{i+1} \in V, i = 1 \dots n$
  - 4)         Do If  $\text{Coord}[v_i] = \text{Coord}[v_{i+1}]$
  - 5)             Then  $\text{MakeEdges}(v_i, v_{i+1});$
- Кінець.

У рядку 1 процедура Make формує множину вершин графа. Цикл у рядках 2–5 перевіряє збіг екранних координат гарячих точок елементів. При отриманні позитивного результату процедура MakeEdges формує два симетричних орієнтованих ребра.

**Метод побудови топологічного і ОАА-графів**

Топологічний граф будується шляхом аналізу ОС-графа. При цьому виникають такі проблеми:

- 1) вибір наступної непереглянутої вершини;
- 2) повторне створення того ж вузла (гілки);
- 3) зациклення при обході графа.

Вирішення першої проблеми полягає у виключенні симетричності, а другої і третьої – у виключенні циклів. Отже, при побудові топологічного графа необхідно паралельно перетворити ОС-граф в орієнтований ациклічний антисиметричний граф (ОАА-граф) [7].

Існують два методи обходу вершин графа: обхід у ширину і обхід у глибину [4]. Вони різняться порядком перегляду вершин. Оскільки для побудови топологічного і ОАА-графів треба проаналізувати усі вершини, то не має значення, який із методів використовувати. Зупинимось на методі вглибину.

Суть методу полягає в наступному.

Для поточної вершини  $v_i$  крім першої  $v_1$  необхідно вилучати дугу, яка веде від  $v_i$  до пройденої перед цим вершини  $v_{i-1}$ . Тим самим виключається симетричність.

Треба провести аналіз ОС-графа, а саме правильно визначити і зв'язати між собою предикати P, Q і I.

Предикат I відповідає вузлу. Вузлом може бути або шина, або точка перетину трьох і більше гілок. У місці виявлення вузла присутнє транзитивне відношення (T), що спричиняє зациклення і створення копій I і Q. Необхідно порушити T-відношення, видаливши усі дуги між утворюючими його вершинами, ввести вершину-вузол  $V_{node}$  і розташувати покажчики так, як це показано на рис. 6.

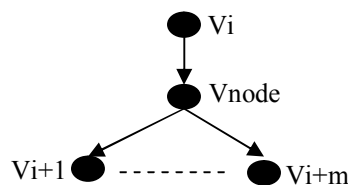


Рис. 6. Введенн. . . . .и-вузла:  $V_i$  – вершина, по ребрах якої виявлено вузол;  $V_{node}$  – вершина-вузол, що вводиться;  $V_{i+1}, \dots, V_{i+m}$  – вершини, суміжні  $V_i$

Предикати P і Q формуються при виявленні кінця гілки, а саме:

- 1) при виявленні вузла;
- 2) при виявленні тупика, тобто вершини, в якій немає вихідних дуг.

До моменту виявлення кінця гілки вершини розміщуються в такі тимчасові сховища:

- 1) стек обходу – сховище, що містить непроаналізовані вершини в порядку, що відповідає методу вглибину;

2) стек вузлів – сховище, що містить інформацію про вузли до моменту їхнього повного формування, а саме: кількість ще не створених гілок, посилання на вже створені гілки;

3) стек гілки – сховище, що містить вершини, які відносяться до однієї гілки, до моменту виявлення її кінця.

Топологічний і ОАА-графі вважаються сформованими, коли всі вершини ОС-графу проаналізовані.

### Алгоритм формування графів

Представимо алгоритм формування топологічного і ОАА-графу у вигляді сукупності продукцій, виконуваних для кожної вершини ОС-графу.

Вхідні дані – орієнтований симетричний граф.

Вихідні дані – топологічний граф і орієнтований ациклічний антисиметричний граф.

Нехай  $v_i$  – поточна досліджувана вершина.

Призначення продукцій:

- продукція 1 – перетворення симетричного графа в антисиметричний;
- продукції 2, 3 – перевірка умов існування предиката I;
- продукції 4, 5 – перевірка умов предикатів Q і P;
- продукції 6, 7 – формування предикатів Q і P;
- продукції 8, 9, 10, 11, 12 – формування предиката I:
  - продукція 8 відповідає початковому стану алгоритму, вузол створюється;
  - продукція 9 – розглянута вершина не містить вихідних дуг, вузол доповнюється;
  - продукція 10 – знову сформована гілка повинна приєднатися до двох вузлів: уже створеному вузлу зі стека вузла і тільки що створеному, вузол доповнюється і вузол створюється;
  - продукція 11 – розглянута вершина відповідає елементу – шина, вузол доповнюється і вузол створюється;
  - продукція 12 – вбудовування елементів – вузлів і видалення циклів;
- продукції 13, 14, 15 – робота зі стеком вузлів;
  - продукції 13, 14 – визначення кількості гілок для стека вузла;
  - продукція 15 – видалення вузла зі стека вузла;
- продукція 16 – робота зі стеком гілки;
- продукція 17 – перевірка умови зупинки алгоритму.

### Послідовність продукцій

#### Початок.

- 1) **Якщо** – вершина  $v_i$  не корінь,  
**То** – видалити дугу між  $v_i$  і вершиною  $v_{i-1}$ , де  $i$  – порядок перегляду вершин
- 2) **Якщо** – вершина  $v_i$  не шина,  
**I**  
– вершина  $v_i$  містить кількість вихідних дуг  $\geq 2$ ,  
**I**  
– вершини суміжні з  $v_i$  не є шинами,  
**То** – має місце існування вузла.
- 3) **Якщо** – вершина  $v_i$  шина,  
**То** – має місце існування вузла  
**I**  
– вершина  $v_i$  є вузлом.
- 4) **Якщо** – має місце існування вузла  
**ЧИ**  
вершина  $v_i$  не містить вихідних дуг,  
**То** – має місце існування гілки, тобто предиката Q.
- 5) **Якщо** – має місце існування гілки,  
**I**  
– у стеку гілки кількість елементів  $> 1$ ,  
**То** – має місце існування предиката P.
- 6) **Якщо** – має місце існування предиката P,  
**То** – для кожних двох сусідніх предикатів R у стеці гілки:  
– сформувати предикат P;  
– сформувати дуги на відповідні предикати R і помістити в P;  
– сформувати властивості предиката P і доповнити загальні властивості для предикатів R, привласнити предикату P унікальний індекс.
- 7) **Якщо** – має місце існування гілки, тобто предиката Q  
**I**

- предикат (и)  $P$  сформовані,  
**То** – сформувати предикат  $Q$ ,  
– сформувати дуги між  $Q$  і відповідними предикатами  $P$ ;  
– сформувати властивості предиката  $Q$  і доповнити загальні властивості для предикатів  $P$ ,  
– привласнити предикату  $Q$  унікальний індекс.  
**Інакше** – сформувати предикат  $Q$ ;  
– сформувати дуги між  $Q$  і відповідним предикатом  $R$ ;  
– сформувати властивості предиката  $Q$  і доповнити властивості предиката  $R$ ;  
– привласнити предикату  $Q$  унікальний індекс;  
– очистити стек гілки.
- 8) **Якщо** – стек вузла порожній  
**I**  
– вершина  $v_i$  не шина  
**I**  
– предикат  $Q$  сформований,  
**То** – сформувати вузол, тобто предикат вузла;  
– приєднати до вузла гілку  $Q$ ;  
– занести вузол у стек вузла.
- 9) **Якщо** – стек вузла не порожній  
**I**  
– вершина  $v_i$  не містить вихідних дуг  
**I**  
– предикат  $Q$  сформований,  
**То** – дістати зі стека вузла верхній вузол  $top$  і приєднати до нього сформовану гілку (предикат  $Q$ );  
– зменшити число гілок верхнього вузла  $top$  у стеку вузла на 1;  
– виконати продукцію 15.
- 10) **Якщо** – стек вузла не порожній  
**I**  
– вершина  $v_i$  не шина  
**I**  
– предикат  $Q$  сформований,  
**То** – дістати зі стека вузла верхній вузол  $top$  і приєднати до нього сформовану гілку (предикат  $Q$ );  
– зменшити число гілок верхнього вузла  $top$  у стеку вузла на 1;  
– виконати продукцію 15;
- 11) **Якщо** – вершина  $v_i$  шина  
**I**  
– предикат  $Q$  сформований,  
**То** – сформувати покажчик на предикат  $R$ , що відповідає шині;  
– дістати зі стека вузла верхній вузол  $top$  і приєднати до нього сформований покажчик на предикат  $R$ .
- 12) **Якщо** – має місце існування вузла  
**I**  
– вершина  $v_i$  не шина,  
**То** – створити вершину – вузол  $v_{node}$ ;  
– сформувати дугу від  $v_i$  до  $v_{node}$ ;  
– сформувати дуги від  $v_{node}$  до усіх вершин, суміжних з  $v_i$ ;  
– вилучити всі дуги між  $v_i$  і вершинами, суміжними з  $v_i$ ;  
– вилучити всі дуги між вершинами, суміжними з  $v_i$ .
- 13) **Якщо** – вершина  $v_i$  не шина,  
**То** – дістати зі стека вузла верхній вузол і присвоїти йому число гілок вузла = числу вихідних дуг вершини.
- 14) **Якщо** – вершина  $v_i$  шина,  
**То** – дістати зі стека вузла верхній вузол і присвоїти йому число гілок вузла = (числу вихідних дуг вершини – 1), у цьому випадку одна гілка вузла-шини вже сформована.
- 15) **Якщо** – число гілок вузла = 0,  
**То** – видалити вузол зі стека вузла.
- 16) **Якщо** – не має місця існування гілки,  
**То** – додати вершину в стек гілки.  
**Інакше** – очистити стек гілки.
- 17) **Якщо** – стек вузла порожній  
**I**  
– стек гілки порожній,  
**То** – **Кінець алгоритму.**

Використання запропонованих моделей і методів у програмних комплексах АСДУ дозволить істотно скоротити час вирішення задач, що базуються на топології електричної мережі.

#### ЛІТЕРАТУРА:

1. *Архипова Е.Н.* Требования к математическим моделям энергетических объектов, используемых в современных тренажерах для оперативного персонала // Энергосбережение и водоподготовка. – 1999. – № 4. – С. 72–78.
2. *Борданов С.А., Борданова Ю.С.* Автоматическое построение схемы замещения на ПЭВМ для расчета и анализа режимов электрических систем // Электроснабжение и автоматизация промышленных предприятий. – Чуваш. гос. ун-т. – Чебоксары, 1997. – С. 99–102.
3. *Шабанов-Кушнаренко Ю.П.* Теория интеллекта: Математические средства. – Харьков: Вища шк. Изд-во при Харьк. ун-те, 1984. – 143 с.
4. *Новиков Ф.А.* Дискретная математика для программистов. – С.-Пб.: Питер, 2001. – 304 с.
5. *Литский В.* Комбинаторика для программистов: Пер. с польск. – М.: Мир, 1988. – 213 с.
6. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: построение и анализ. – М.: МЦНМО, 2000. – 960 с.
7. *Кристофидес Н.* Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.
8. *Дударь З.В., Охрименко М.В., Диалло Тьерно Амиру.* Редакторы схем в составе прикладного программного обеспечения АСДУ // Вестник Харьковского государственного политехнического университета. – 2000., – В. 125. – С.71–77.

**БОНДАРЕНКО Михайло Федорович** – доктор технічних наук, професор, ректор Харківського національного університету радіоелектроніки.

Наукові інтереси:

– автоматизовані системи диспетчерського управління.

Тел. (0572) 43–30-58, email: swell@kture.kharkov.ua.

**ДУДАР Зоя Володимирівна** – кандидат технічних наук, професор, завідувач кафедри програмного забезпечення ЕОМ Харківського національного університету радіоелектроніки.

Наукові інтереси:

– автоматизовані системи диспетчерського управління.

Тел. (0572) 40-94-46, email: swell@kture.kharkov.ua.

**ЗБИТНЄВА Майя Вячеславівна** – аспірант кафедри програмного забезпечення ЕОМ Харківського національного університету радіоелектроніки.

Наукові інтереси:

– автоматизовані системи диспетчерського управління.

Тел. (0572) 40-94-46, email: swell@kture.kharkov.ua.

Подано: 5.02.2002.

**Бондаренко М.Ф., Дудар З.В., Збітнєва М.В.** Моделі електричних мереж і методи автоматичного формування їх топологій

**Бондаренко М.Ф., Дудар З.В., Збітнєва М.В.** Модели электрических сетей и методы автоматического формирования их топологий.

**Bondarenko M.F., Dudar Z.V., Zbitnyeva M.V.** Models of electrical networks and automatic forming methods of their topologies.

УДК 621.311: 681.5

М.Ф. Бондаренко, З.В. Дударь, М.В. Збитнєва. Модели электрических сетей и методы автоматического формирования их топологий.

Предложены модели элементов электрических сетей, базирующиеся на теории предикатов. Разработаны метод и алгоритм построения ориентированного симметрического графа, метод и алгоритм построения топологического графа электрической сети, способ представления графов.

УДК 621.311: 681.5

M.F. Bondarenko, Z.V. Dudar, M.V. Zbitnyeva. Models of electrical networks and automatic forming methods of their topologies.



Models of electrical network elements, basing on theories of predicates are offered. Method and algorithm of making oriented symmetric graph, method and algorithm of making topological graph of electrical network, way of representation of graphs were developed.