

Р.Б. Дунець

АЛГОРИТМ ПОШУКУ КОНТУРІВ У ТОПОЛОГІЇ СХЕМ СИСТЕМ КЕРУВАННЯ

Важливою задачею при побудові моделюючих систем роботи схем керування є аналіз топології схем з метою виявлення контурів — обернених зв'язків між компонентами. Розв'язок цієї задачі дозволяє коректно визначати послідовність моделювання компонентів схем систем керування.

Існуючі методи [1] аналізу — комбінаторні методи, які передбачають значні затрати часу на виконання великої кількості порівнянь з'єднань компонентів. Крім того, ці методи не розраховані на подання схем з'єднань компонентів у вигляді матриць суміжностей [2], які найбільш придатні для побудови моделюючих програм.

У роботах [3, 4] для розділення схеми на яруси, як першого етапу аналізу топології схем, запропоновано використання логічних операцій, а саме: логічного множення ($\times$) матриць, кон'юнкції ($\&$) матриць, диз'юнкції ($\vee$) матриць, інверсія матриці. У даній роботі показано, що ці операції також можуть бути застосовані для аналізу топології схем систем керування, зокрема для виявлення всіх компонентів схеми, які утворюють контури.

Якщо в схемі керування є хоч би один контур, то, очевидно, в такій схемі мусить бути хоч би один компонент, який би мав більш ніж один вхід. Один із входів такого компонента буде з'єднаний із своїм же виходом або з виходом іншого компонента, який своїм входом сполучений навіть через інші компоненти з виходом даного компонента.

Наведемо теореми, які лежать в основі алгоритму пошуку компонентів, що входять в контури схеми.

Теорема 1. Якщо існує матриця суміжності з'єднань компонентів схеми по входах $A = [a_{ij}]_{n \times n}$, а також матриця-рядок $K = [k_i]_n$, в якій є тільки один елемент, відмінний від нуля, що відповідає компоненту схеми, який входить в контур, то в результаті логічного множення утвориться матриця $R = K^T \times A$, одиничні елементи котрої

будуть відповідати компонентам схеми, що своїми входами з'єднані з компонентом контуру.

Доведення. Нехай елемент r_j матриці R , який визначається як $r_j = k_1 \& a_{j1} \vee \dots \vee k_i \& a_{ji} \vee \dots \vee k_n \& a_{jn}$, дорівнює одиниці. Якщо $r_j = 1$, то хоч би одна з кон'юнкцій повинна набути одиничного значення. Нехай згідно з умовами теореми в матриці K її елемент $k_i = 1$. Тоді в матриці A її елемент a_{ji} повинен дорівнювати одиниці. Якщо $a_{ji} = 1$, то це означає, що j -й компонент схеми з'єднаний по входу з i -м компонентом схеми, який входить у контур схеми, бо прийняли, що $k_i = 1$. Теорема доведена.

Наслідки теореми:

1. Компонент схеми, який зв'язаний з компонентом схеми з контуру, може не входити в контур.

2. Якщо матриця K містить більше одного одиничного елемента, то вони можуть належати до різних контурів.

Теорема 2. Якщо існує матриця суміжності з'єднань компонентів схеми по входах $A = [a_{ij}]_{n \times p}$, а також матриця-рядок $K = [k_i]_p$, в якій є тільки один елемент, відмінний від нуля, що відповідає компоненту схеми, який входить в контур, то в результаті логічного множення утвориться матриця $S = K \times A$, одиничні елементи котрої будуть відповідати компонентам схеми, що своїми виходами з'єднані з компонентом контуру.

Доведення теореми аналогічне теоремі 1.

Теорема 3. Якщо існує матриця R , одиничні елементи якої відповідають компонентам схеми, що своїми входами з'єднані з компонентом контуру, і матриця S , одиничні елементи котрої відповідають компонентам схеми, що своїми виходами з'єднані з компонентом контуру, то в матриці $K = R \& S$ її одиничні елементи відповідають компонентам схеми, які утворюють контур.

Доведення теореми аналогічне теоремі 1.

Алгоритм пошуку компонентів контурів наведений на рис. 1. Він передбачає циклічне виконання f кроків ($f = 1, 2, 3, \dots$). Спочатку при аналізі топології схеми керування приймається, що кожен з її компонентів входить у контур, тобто всі елементи матриці-рядка K_0 є одиничні. Далі виконується операція $R_f = K^{T_{f-1}} \times A$ пошуку множини

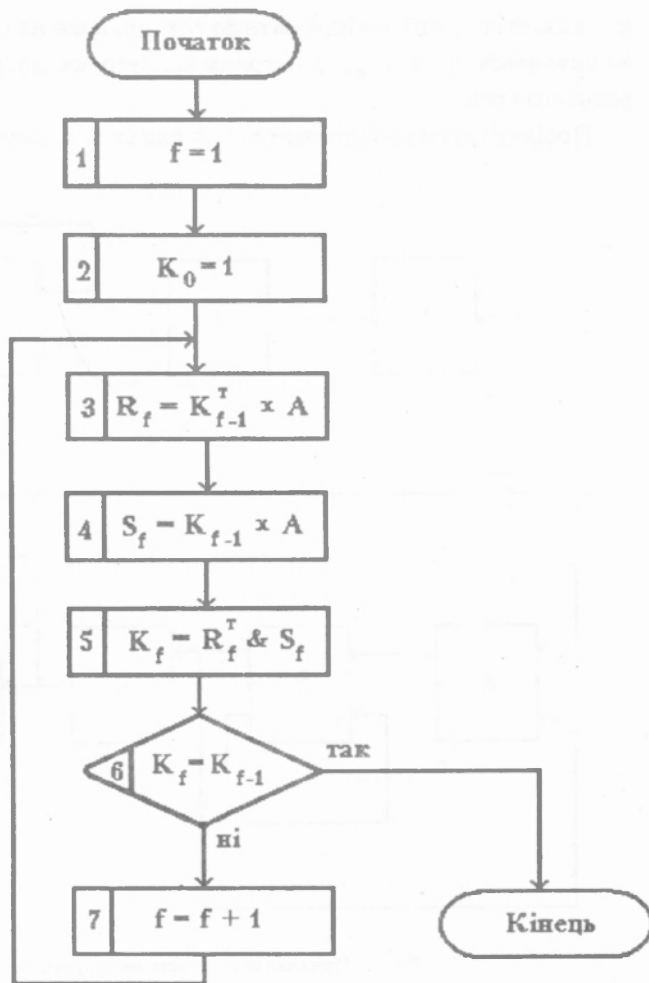


Рис. 1. Алгоритм пошуку компонентів контурів.

компонентів, які своїми входами з'єднані з компонентами контурів. За допомогою операції $S_f = K_{f-1} \times A$ визначається множина компонентів, які своїми виходами з'єднані з компонентами контурів, а операції $K_f = R_f^T \& S_f$ — множина компонентів схеми, котрі можуть утворювати контури. Саме можуть утворювати, а не утворюють, бо спочатку припускалося, що всі компоненти входять у контури. Для остаточного визначення множини компонентів усіх контурів потріб-

но вказані операції повторювати до тих пір, поки на черговому кроці не виявиться, що утворена матриця K_f повторює матрицю K_{f-1} попереднього кроку.

Проілюструємо роботу алгоритму для схеми, наведеної на рис. 2.

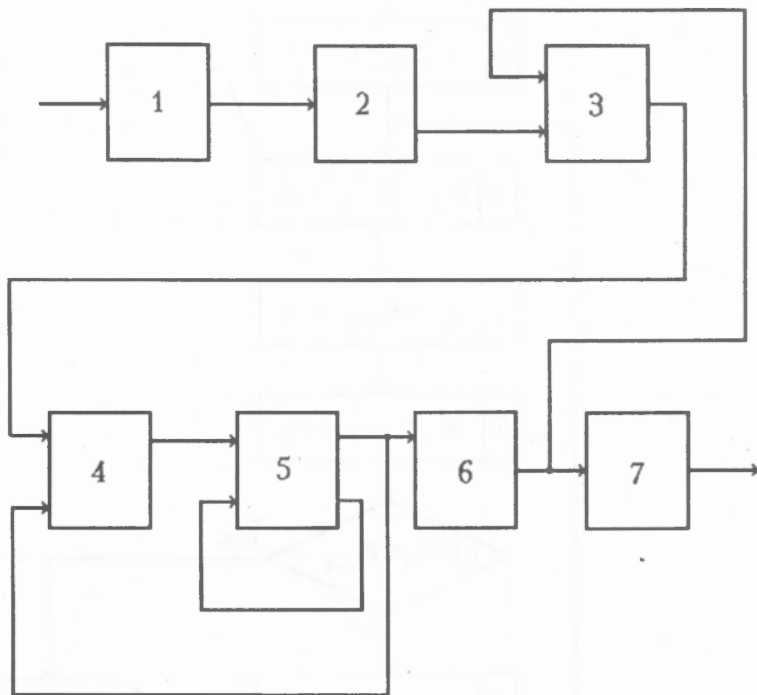


Рис. 2. Приклад схеми системи керування.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad K_0 = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1];$$

$$f = 1$$

$$R_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

$$S_1 = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0],$$

$$K_1 = [0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \& [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0] = [0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0],$$

$$K_1 \neq K_2;$$

$$f=2$$

$$R_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

$$S_2 = [0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0] \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0],$$

$$K_2 = [0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1] \& [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0] = [0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0],$$

$$K_2 \neq K_1;$$

$$f=3$$

$$R_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

$$S_3 = [0011110] \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} = [0111110],$$

$$K_3 = [0011111] \& [0111110] = [0011110],$$

$K_3 - K_2$;

Кінець. Контур утворюють компоненти 3, 4, 5, 6.

Запропонований алгоритм, а особливо логічні операції над матрицями, просто реалізуються на ЕОМ. Цей алгоритм може бути використаний при побудові моделюючих програм автоматизованих систем проектування.

1. Алгоритм визначення послідовності моделювання роботи елементів в схемах на основі логічних операцій /Дунець Р.Б. //Тези доп. І Укр. конф. з автоматизованого керування: К., 1994. Ч. II. С. 298. 2. Вавилов А.А., Имаев Д.Х. Машинные методы расчета систем управления. Л., 1981. 3. Верж К. Теория графов и ее применения. М., 1962. 4. Дунець Р.Б. Визначення послідовності моделювання аналогових елементів в системах автоматичного керування //Інформаційні технології та розпізнавання образів: Збірник наукових праць. 1993. Т. III. Ч. 2.

Стаття надійшла до редколегії 24.01.96