

Gonnet G.H., Hare D.E.G. On the Lambert W Function //Advances in Comput. Math. 1996. Vol 5. P.329-359.

УДК 519.72

В.А. Ліщинський, О.П. Хома

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ СИНТЕЗУ СКІНЧЕННИХ АВТОМАТІВ

Основою синтезу скінченних автоматів будь-якого типу - детермінованих, імовірнісних, нечітких, розпізнавачів і перетворювачів, структурних автоматних моделей - є частковий синтез функції переходів з відповідним зважуванням її елементів. Тому найдоцільніше покращувати побудову скінченних автоматів, створюючи ефективні методи і алгоритми синтезу функцій переходу.

Пропонується ефективний за часовим та ємнісним критеріями метод побудови функції переходів, заданої безпосередньо. Вхідним для нього є регулярний вираз R у вхідному алфавіті X автомата, результатом - послідовність F трійок $u_i x u_j$ таких, що $u_j = \delta(u_i, x)$, $u_i, u_j \in U$, $x \in \{\varepsilon\} \cup X$, де δ - функція переходів, E - множина станів автомата, ε - порожнє слово. Метод використовує вхідний магазин M_R , вихідний M_F , внутрішній M , лічильники J та I . У магазинах M_R і M_F зберігаються відповідно регулярний вираз і функція переходів або їхні частини. Лічильник I призначено для підрахунку кількості пар дужок, а J - кількості певних станів. Магазин M забезпечує правильне створення індексів станів автомата. Лічильники і магазин M ідентифіковані своїми алфавітами.

Функція переходів будується за один перегляд регулярного виразу R як результат переходу $\pi : [R; O; O; \emptyset; \rho_0] \rightarrow [\emptyset; j; i; \emptyset; F]$, де ρ_0 - початковий стан автомата, $i \in I$, $j \in J$. На кожному кроці у залежності від символу у вершині вхідного магазину застосовується лише одне з наступних правил.

$$[S\omega_R; j; i; \omega_M; \omega_F c] \rightarrow [\omega_R; j; i; \omega_M; \omega_F \subset S],$$

$$[S\omega_R; j; i; \omega_M; \omega_F S'] \rightarrow [\omega_R; n; i; \omega_M; \omega_F S' u_n, u_n S],$$

$$[(\omega_R; j; i; \omega_M; \omega_F] \rightarrow [\omega_R; j; m; m\omega_M; \omega_F p_m p_m],$$

$$\begin{aligned}
[]\omega_R; j; i; l\omega_M; \omega_F] &\rightarrow [\omega_R; j; i; \omega_M; \omega_F k_l, k_l], \\
[<\omega_R; j; i; \omega_M; \omega_F] &\rightarrow [\omega_R; j; m; -m\omega_M; \omega_F p_m, (p_m, k_m)], \\
[>\omega_R; j; i; -l\omega_M; \omega_F] &\rightarrow [\omega_R; j; i; \omega_M; \omega_F k_l, (p_l, k_l)], \\
[\cup\omega_R; j; i; -l\omega_M; \omega_F] &\rightarrow [\omega_R; j; i; -l\omega_M; \omega_F k_l, (p_l, k_l)], \\
[\cup\omega_R; j; i; l\omega_M; \omega_F] &\rightarrow [\omega_R; j; i; l\omega_M; \omega_F k_l, p_l], \\
[\emptyset; j; i; \omega_M; \omega_F] &\rightarrow [\emptyset; j; i; \omega_M; \omega_F k_0],
\end{aligned}$$

де ω_R - кінцеве підслово виразу R ;
 ω_F - початкова частина послідовності F ;
 ω_M - слово алфавіту M ;
 c - допоміжний символ, $c \notin X$;
 l, m, n - цілі додатні числа, $m = i + 1$, $n = j + 1$;
 S, S' - літери вхідного алфавіту;
 $"<", ">"$ - ітераційні дужки;
 $u_n, k_l, k_m, p_l, p_m, k_0$ - стани автомата.

Після правила 9 ніяке інше правило не застосовується.

Для усунення трійок типу $a\epsilon b, a, b \in U$ слід послідовно виконати наступні дії.

1. Множину ваг M_a стану a замінити множиною $M_a \cup M_b$, де M_b - множина ваг стану b .

2. Для кожної трійки типу bSb :

а) якщо існує хоча б одна трійка $aS'd, d \in U$ така, що $d \neq a$ і $S' = S$, то доповнити послідовність F трійкою aSb ;

б) якщо не існує трійки $aS'd$, то доповнити F трійкою aSa .

Вилучити трійку $a\epsilon b$ з F .

3. Вилучити з послідовності F усі трійки типу aSb , якщо немає трійок $dS''a, S'' \in X$.

4. Множиною заключних станів автомата є множина усіх станів, позначених k_0 .

Побудовано програму, яка реалізує запропонований метод разом з усуненням зайвих трійок.