JAO - Języki, Automaty i Obliczenia - Wykład 3

Postępujemy podobnie jak przy domknięciu tranzytywnym macierzy boolowskiej.

Niech $Q = \{1, 2, .., n\}$. Dla automatu skończonego A zdefiniujemy $i \xrightarrow{w}_k j$ gdy można za pomocą słowa w przejść od stanu i do j tak aby wszystkie stany pośrednie były o numerach co najwyżej k.
Konstruujemy macierze

$$R^k_{i,j} = \{ w : i \xrightarrow{w}_k j \}.$$

Elementy macierzy sa odpowiadającymi wyrażeniami regularnymi opisującymi $R^k_{i,j}$. Mamy:

$$R^0_{i,j} = \{ a \in \Sigma : i \xrightarrow{a} j \}.$$

$$R^{k+1}_{i,j} = R^k_{i,j} \cup R^k_{i,k+1} \cdot (R^k_{k+1,k+1})^* \cdot R^k_{k+1,j}.$$

$$L(A) = \bigcup_{j \in F} R^n_{q_0,j}.$$

Rozwiązywanie układu równań
Korzystamy z następującego faktu:

Fakt.
Jeśli $\varepsilon \notin A$ to
jedynym rozwiązaniem równania $X \; = \; AX \cup B$ jest A^*B.

Oczywistym jest, że $X \subseteq A^*B$. Trzeba jeszcze uzasadnić zawieranie

$$\forall\, w \in A^*B \text{ zachodzi } w \in X$$

Dowód indukcyjny ze względu na długość w.

Zakładamy, że nie ma ε-przejść w automacie.
Niech $L_q = L(A, q)$ będzie zbiorem słów które A akceptuje startując ze stanu q.
Oznaczamy $\varepsilon(q) = \{\varepsilon\}$ gdy $q \in F$, $\varepsilon(q) = \emptyset$ gdy $q \notin F$.

Konstruujemy układ równań:

$$L_i = \bigcup_j R^0_{i,j} \cdot L_j \cup \varepsilon(i)$$

Układ rozwiązujemy metodą eliminacji kolejnych zmiennych korzystając z równania $X = AX \cup B$. Na koniec:

$$L(A) = L_{q_0}.$$

Niech EZ_n będzie następującym automatem:

$\Sigma = \{ a_{i,j} : 1 \leq i, j \leq n,\ i \neq j \}$

$Q = \{1, 2, \ldots n\}$

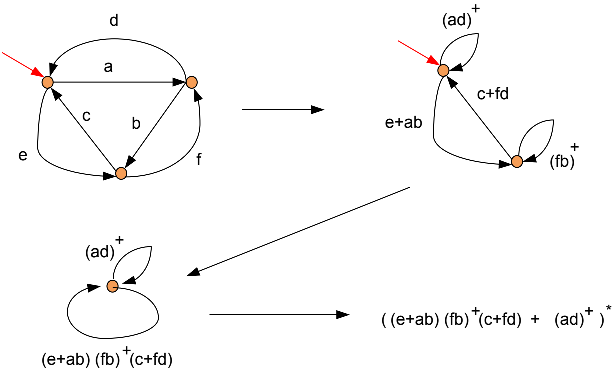
jedyne przejścia automatu są typu $i \xrightarrow{a_{i,j}} j$

$q_0 = 1,\ F = \{ q_0 \}$

Andrzej Ehrenfeucht, H. Paul Zeiger w pracy "Complexity Measures for Regular Expressions". J. Comput. Syst. Sci. 12(2): 134-146 (1976), udowodnili:

Twierdzenie

Dla $n \geq 2$ najkrótsze standardowe wyrażenie regularne dla języka $L(EZ_n)$ ma długość co najmniej 2^{n-2}.


d
a
c
b
e
f
(ad)+
c+fd
e+ab
(fb)+
(ad)+
(e+ab) (fb)+(c+fd)
((e+ab) (fb)+(c+fd) + (ad)+)*

Automat A ma dodatkowo przejścia (tranzycje) postaci $p \xrightarrow{\varepsilon} q$. Robimy domknięcie tej relacji ze względu na ciągi liter, przeplatanych z dowolnie długimi ciągami ε-tranzycji. Wtedy

$$L(A) = \{ w : q_0 \xrightarrow{w} q \in F$$

Usuwanie ε-tranzycji.

- Obliczamy domknięcie tranzytywne relacji $\xrightarrow{\varepsilon}$;
- powiększamy zbiór stanów akceptujących
  $$F' = \{ p : p \xrightarrow{\varepsilon}^* q \in F \}$$
- Dla $a \in \Sigma$ jeśli $p \xrightarrow{\varepsilon}^* q' \xrightarrow{a} q$ w orginalnym automacie to dodajemy $p \xrightarrow{a} q$ w zmodyfikowanym automacie.
- usuwamy ε-tranzycje.

Dla wyrażenia W skonstruujemy rekurencyjnie automat z ε-tranzycjami spełniający następujące warunki:

- jeden stan początkowy i jeden akceptujący, dwa różne stany
- nie wychodzi tranzycja ze stanu akceptującego
- z każdego stanu wychodzą tylko ε-tranzycje, co najwyżej dwie, albo jedna a-tranzycja dla pewnego $a \in \Sigma$
- nie ma ε-cyklu

Początkowo dla wyrażeń opisujących jedno słowo puste lub jeden symbol alfabetu konstruujemy automaty dwustanowe.
Mając automaty A_1, A_2 dla podwyrażeń W_1, W_2 konstruujemy automat A dla $W_1 W_2$, $W_1 + W_2$, lub W_1^* będący kompozycja automatów A_1, A_2. Zakładamy przy tym że zbiory stanów A_1, A_2 sa rozłączne oraz, że q_0', q_0'' są stanami początkowymi, a q_a', q_a'' są stanami akceptującymi odpowiednio automatów A_1, A_2 .

Automat A dla większego wyrażenia W konstruujemy nastepujaco.

Suma: $W = W_1 + W_2$.
Tworzymy nowy stan q_0 jako początkowy oraz tranzycje

$$q_0 \xrightarrow{\varepsilon} q_0', \;\; q_0 \xrightarrow{\varepsilon} q_0'', \;\; q_a'' \xrightarrow{\varepsilon} q_a'$$

q_0' przestaje byc początkowy, q_a'' przestaje byc akceptujący.

Konkatenacja: $W = W_1 \cdot W_2$. Utożsamiamy stan q_a' ze stanem q_0'.

Gwiazdka: $W = W_1^*$.
Dodajemy nowy stan początkowy q_0, i nowy akceptujący q_a. Stan q_0' przestaje być początkowy, a stan q_a' przestaje byc akceptujący.
Dodajemy tranzycje:

$$q_0 \xrightarrow{\varepsilon} q_a', \;\; q_a' \xrightarrow{\varepsilon} q_0', \;\; q_a' \xrightarrow{\varepsilon} q_a.$$

Oznaczamy przez $sh(W)$ głębkość zagnieżdżenia gwiazdkowego wyrażenia W.

Definicja

Wyrażenie regularne nazywamy płaskim gdy $sh(W) \leq 1$, język nazywamy płaskim gdy może być być opisany płaskim wyrażeniem regularnym.

Płaskim jest np. wyrazenie

$$ab(aa + bbc)^* + abc(ab)^*(bbcb)^*$$

Dla zbioru $X \subseteq N^k$ oznaczmy przez $< X >$ zbiór wszystkich liniowych kombinacji wektorów z X, gdzie współczynniki przy liniowych kombinacjach są nieujemne.

Oznaczmy

$$(z+ < X >) = \{ z + x \ : \ x \in X \}.$$

Definicja

Zbiory postaci $(z+ < X >)$ dla skończonych X nazywamy **liniowymi**, a skończone sumy takich zbiorów nazywamy **semiliniowymi**.

Funkcja Parikha. Niech $\Sigma = \{a_1, a_2, \ldots, a_k\}$.
Dla słowa w oznaczmy
$\Psi(w) = (\#_{a_1}(w), \#_{a_2}(w), \#_{a_3}(w), \ldots \#_{a_k}(w)$.
Dla języka L oznaczamy

$$\Psi(L) = \{ \Psi(w) \ : \ w \in L \}.$$

Na przykład: $\Psi(bb(aa + bb)^* + aa(ab)^*(bbb)^* =$

$$([0,2] + < \{[2,0], [0,2]\} >) \cup ([2,0] + < \{[1,1], [0,3]\} >).$$

Twierdzenie

Jeśli L jest językiem regularym to $\Psi(L)$ jest semiliniowy oraz istnieje język regularny płaski L' taki, że $\Psi(L) = \Psi(L')$.

Oczywiście suma i konkatenacja płaskich języków jest płaskim językiem. Wystarczy udowodnić, że dla języka płaskiego L istnieje płaski język L' taki, że $\Psi(L) = \Psi(L')$. Niech $L = L_1 \cup L_2 \cup L_3 \cup \ldots L_m$, gdzie $\Psi(L_i)$ są zbiorami liniowymi.

$$\Psi((L_1 \cup L_2 \cup L_3 \cup \ldots L_m)^*) \;=\; \Psi(L_1^* \, L_2^* \, L_3^* \, \ldots L_m^*)$$

Teza wynika z implikacji:

- $\Psi(W) = (z + < X >) \;\Rightarrow\; \Psi(W^*) \;=\; (z + < X + z >)$
- $[\,(\forall\, i \leq m\,)\, \Psi(W_i) = (z_i + < X_i >)\,] \quad \Rightarrow$

  $\Psi(W_1 \cdot W_2 \cdot \ldots W_m) \;=\; (z_1 \,+ z_2 \,+ \ldots z_m + < X_1 + X_2 + \ldots X_m >)$

Automat z wyjściem A ma dodatkowo ma określony alfabet wyjściowy Δ oraz funkcję przejść λ.

- Dla automatu typu Mealy $\lambda : \; Q \times \Sigma \to \Delta$, (automat czytając symbol a zaczynając w stanie q wypisuje symbol wyjściowy b, zapis $q \xrightarrow{a/b} q'$).
- Dla automatów typu Moore'a $\lambda : \; Q \to \Delta$ (automat wypisje symbol $\lambda(\delta(q, a))$. Najpierw wykonuje tranzycje a potem wypisuje. Automat typu Mealy łatwo przerobic technicznie na Moore'a wpisując wyjście jako część informacji związanej ze stanem.

Ciekawym przykładem jest automat czytający trzy liczby n-bitowe od najmnej znaczących cyfr i wypisujący ich n-bitową sumę (być może oprócz najbardziej znaczącego bitu).

Dla wejścia 011100110 wyjściem jest 001101010. Dla drugiego automatu $\lambda((q, i)) = i)$.