Własności domknięcia automatów z więzami równościowymi

Marta Maciejewska

18 kwietnia 2012

Spis treści

Własności domknięcia

	AWEDC	AWCBB	Automaty redukcyjne
przecięcie			
suma			
determinizacja			
dopełnienie			

Automat skończony na drzewach

Definicja

Skończony automat niedeterministyczny $A = (Q, \Sigma, Q_f, \Delta)$, gdzie

- Q - zbiór stanów
- $Q_f \subseteq Q$ - zbiór stanów akceptujących
- Σ - alfabet (skończony, urangowany)
- Δ - reguły przejścia

Wersja "z dołu do góry"

Reguły przejścia Δ postaci

$$f(q_1(x_1), \dots, q_n(x_n)) \to q(f(x_1, \dots, x_n))$$

gdzie:

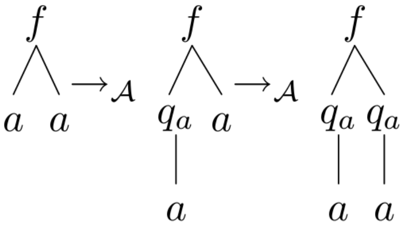
- $f \in \Sigma_n$
- $q, q_1, \dots, q_n \in Q$

Przykład 1

$$\Sigma = \{f_2, g_1, a_0\},\ Q = \{q_a, q_g, q_f\},\ Q_f = \{q_f\},$$

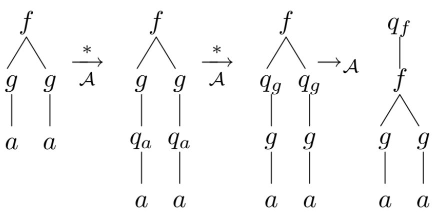
$$\Delta = \{$$
$$\begin{array}{rcl} a & \to & q_a(a) \\ g(q_g(x)) & \to & q_g(g(x)) \\ g(q_a(x)) & \to & q_g(g(x)) \\ f(q_g(x), q_g(y)) & \to & q_f(f(x, y)) \end{array}$$
$$\}$$

Przykład 1



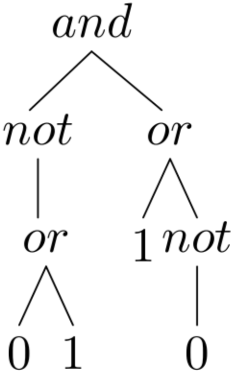
$$
\begin{array}{rcl}
a & \rightarrow & q_a(a) \\
g(q_g(x)) & \rightarrow & q_g(g(x)) \\
g(q_a(x)) & \rightarrow & q_g(g(x)) \\
f(q_g(x), q_g(y)) & \rightarrow & q_f(f(x, y))
\end{array}
$$

Przykład 1



$$
\begin{aligned}
a &\rightarrow q_a(a)\\
g(q_g(x)) &\rightarrow q_g(g(x))\\
g(q_a(x)) &\rightarrow q_g(g(x))\\
f(q_g(x), q_g(y)) &\rightarrow q_f(f(x, y))
\end{aligned}
$$

Przykład 2 - prawdziwe formuły logiczne



Przykład 2 - prawdziwe formuły logiczne

$$\Sigma = \{0_0,\ 1_0,\ not_1,\ and_2,\ or_2\},\quad Q = \{f, t\},\quad Q_f = \{t\}$$

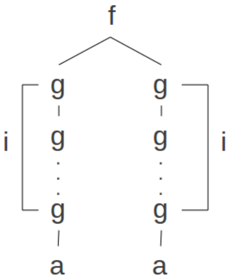
$$\Delta = \{$$

$$\begin{array}{rcl@{\qquad}rcl}
0 & \to & f, & not(t) & \to & f, \\
1 & \to & t, & not(f) & \to & t, \\
and(t,t) & \to & t, & or(t,t) & \to & t, \\
and(t,f) & \to & f, & or(t,f) & \to & t, \\
and(f,t) & \to & f, & or(f,t) & \to & t, \\
and(f,f) & \to & f, & or(f,f) & \to & f
\end{array}$$

$$\}$$

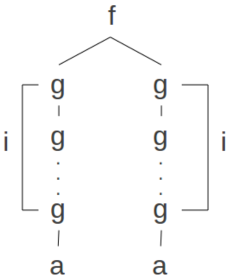
Przykład 3

$$L = \{f(g^i(a), g^i(a)) \mid i > 0\}$$



Przykład 3

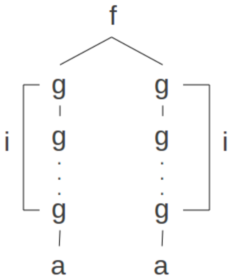
$$L = \{f(g^i(a), g^i(a)) \mid i > 0\}$$



- nie jest rozpoznawany przez taki automat

Przykład 3

$$L = \{f(g^i(a), g^i(a)) \mid i > 0\}$$



- nie jest rozpoznawany przez taki automat
- lemat o pompowaniu

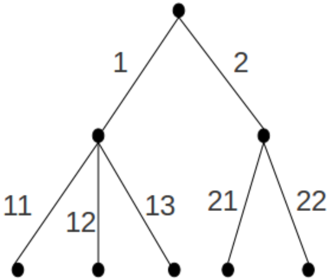
Automaty z warunkami równości i nierówności

Definicja

Automat z warunkami równości i nierówności (AWEDC)
$A = (Q, \Sigma, Q_f, \Delta)$, gdzie

- Q - zbiór stanów
- $Q_f \subseteq Q$ - zbiór stanów akceptujących
- Σ - alfabet (skończony, urangowany)
- Δ - reguły przejścia postaci $\mathbf{f}(\mathbf{q}_1(\mathbf{x}_1), ..., \mathbf{q}_n(\mathbf{x}_n)) \xrightarrow{\mathbf{c}} \mathbf{q}(\mathbf{f}(\mathbf{x}_1, ..., \mathbf{x}_n))$
- c - kombinacja boolowska warunków równości/nierówności

Postać warunków



Przykładowe warunki:

$1 = 2, \quad (13 \neq 22) \vee (13 = 21)$

Przykład 1

Pełne zrównoważone drzewa binarne nad alfabetem $\Sigma = \{f_2, a_0\}$:

$Q = Q_f = \{q\}$,

$$
\Delta = \{
\begin{array}{rcl}
a & \rightarrow & q \\
f(q,q) & \xrightarrow{1=2} & q
\end{array}
\}
$$

Determinizacja

Definicja

Automat $A \in AWEDC$ jest deterministyczny jeśli dla każdego t istnieje co najwyżej jeden stan q taki, że $t \xrightarrow{*}_A q$

Determinizacja

Twierdzenie

Dla każdego automatu $A \in AWEDC$ istnieje deterministyczny automat A', który akceptuje ten sam język co A.

A' może być policzony w czasie wykładniczym i jego rozmiar jest wykładniczy względem A.

Determinizacja

Przypadek automatów skończonych:

Determinizacja

Przypadek automatów skończonych:

Niech $A = (Q, \Sigma, Q_f, \Delta)$

$A_d =_{def} (Q_d, \Sigma, Q_{df}, \Delta_d)$, gdzie:

Determinizacja

Przypadek automatów skończonych:

Niech $A = (Q, \Sigma, Q_f, \Delta)$

$A_d =_{def} (Q_d, \Sigma, Q_{df}, \Delta_d)$, gdzie:

- Q_d to wszystkie podzbiory Q

Determinizacja

Przypadek automatów skończonych:

Niech $A = (Q, \Sigma, Q_f, \Delta)$

$A_d =_{def} (Q_d, \Sigma, Q_{df}, \Delta_d)$, gdzie:

- Q_d to wszystkie podzbiory Q
- Q_{df} to wszystkie podzbiory Q zawierające jakiś stan akceptujący automatu A

Determinizacja

Przypadek automatów skończonych:

Niech $A = (Q, \Sigma, Q_f, \Delta)$

$A_d =_{def} (Q_d, \Sigma, Q_{df}, \Delta_d)$, gdzie:

- Q_d to wszystkie podzbiory Q
- Q_{df} to wszystkie podzbiory Q zawierające jakiś stan akceptujący automatu A
- dla $s, s_1, ..., s_n \in Q_d$, $f(s_1, ..., s_n) \to s \in \Delta_d$ wtw.
  $s = \{q \in Q | \exists q_1 \in s_1, ..., \exists q_n \in s_n, f(q_1, ..., q_n) \to q \in \Delta\}$.

Determinizacja

Dla AWEDC:

Determinizacja

Dla AWEDC:

- obliczanie warunków przejść:

  dla $s, s_1, \ldots, s_n \in Q_d$,

Determinizacja

Dla AWEDC:

- obliczanie warunków przejść:

  dla $s, s_1, \ldots, s_n \in Q_d$, $f(s_1, \ldots, s_n) \xrightarrow{c} s$, gdzie

Determinizacja

Dla AWEDC:

- obliczanie warunków przejść:

  dla $s, s_1, \ldots, s_n \in Q_d$, $f(s_1, \ldots, s_n) \xrightarrow{c} s$, gdzie

$$c \stackrel{def}{=} \Big(\bigwedge_{q \in s} \bigvee_{f(q_1,\ldots,q_n) \xrightarrow{c_r} q \in \Delta, q_i \in s_i} c_r \Big) \wedge \Big(\bigwedge_{q \notin s} \bigwedge_{f(q_1,\ldots,q_n) \xrightarrow{c_r} q \in \Delta, q_i \in s_i} \neg c_r \Big)$$

Determinizacja

Dla AWEDC:

- obliczanie warunków przejść:

  dla $s, s_1, \ldots, s_n \in Q_d$, $f(s_1, \ldots, s_n) \xrightarrow{c} s$, gdzie

$$c \stackrel{def}{=} \Big(\bigwedge_{q \in s} \bigvee_{f(q_1,\ldots,q_n) \xrightarrow{c_r} q \in \Delta, q_i \in s_i} c_r\Big) \wedge \Big(\bigwedge_{q \notin s} \bigwedge_{f(q_1,\ldots,q_n) \xrightarrow{c_r} q \in \Delta, q_i \in s_i} \neg c_r\Big)$$

- reszta konstrukcji tak jak dla automatów bez warunków

Determinizacja

	AWEDC	AWCBB	Automaty redukcyjne
przecięcie			
suma			
determinizacja	2^n		
dopełnienie			

Przecięcie

$$A_1 = \{Q_1, \Sigma, Q_{f1}, \Delta_1\},\ A_2 = \{Q_2, \Sigma, Q_{f2}, \Delta_2\}$$

Przecięcie

$$A_1 = \{Q_1, \Sigma, Q_{f1}, \Delta_1\},\ A_2 = \{Q_2, \Sigma, Q_{f2}, \Delta_2\}$$

- automat produktowy

Przecięcie

$A_1 = \{Q_1, \Sigma, Q_{f1}, \Delta_1\}$, $A_2 = \{Q_2, \Sigma, Q_{f2}, \Delta_2\}$

- automat produktowy
- jeśli $f(q_1, \ldots, q_n) \xrightarrow{c} q \in \Delta_1$ i $f(q'_1, \ldots, q'_n) \xrightarrow{c'} q' \in \Delta_2$, to $f((q_1, q'_1), \ldots, (q_n, q'_n)) \xrightarrow{c \wedge c'} (q, q') \in \Delta$

Przecięcie

	AWEDC	AWCBB	Automaty redukcyjne
przecięcie	$O(n^2)$		
suma			
determinizacja	$O(2^n)$		
dopełnienie			

Suma

$$A_1 = \{Q_1, \Sigma, Q_{f1}, \Delta_1\},\ A_2 = \{Q_2, \Sigma, Q_{f2}, \Delta_2\}$$

Suma

$A_1 = \{Q_1, \Sigma, Q_{f1}, \Delta_1\}$, $A_2 = \{Q_2, \Sigma, Q_{f2}, \Delta_2\}$

- $Q = Q_1 \cup Q_2$

Suma

$A_1 = \{Q_1, \Sigma, Q_{f1}, \Delta_1\}$, $A_2 = \{Q_2, \Sigma, Q_{f2}, \Delta_2\}$

- $Q = Q_1 \cup Q_2$
- $Q_f = Q_{f1} \cup Q_{f2}$

Suma

$A_1 = \{Q_1, \Sigma, Q_{f1}, \Delta_1\}$, $A_2 = \{Q_2, \Sigma, Q_{f2}, \Delta_2\}$

- $Q = Q_1 \cup Q_2$
- $Q_f = Q_{f1} \cup Q_{f2}$
- $\Delta = \Delta_1 \cup \Delta_2$

o ile $Q_1 \cap Q_2 = \emptyset$ (zawsze możemy tak zrobić)

Suma

	AWEDC	AWCBB	Automaty redukcyjne
przecięcie	$O(n^2)$		
suma	$O(n)$		
determinizacja	$O(2^n)$		
dopełnienie			

RA

Dopełnienie

$$A = \{Q, \Sigma, Q_f, \Delta\}$$

Dopełnienie

$$A = \{Q, \Sigma, Q_f, \Delta\}$$

- determinizacja
- $Q_f = Q \backslash Q_f$

Dopełnienie

	AWEDC	AWCBB	Automaty redukcyjne
przecięcie	$O(n^2)$		
suma	$O(n)$		
determinizacja	$O(2^n)$		
dopełnienie	$O(2^n)$		

AWCBB

Przykład 2

- założmy, że mamy pewien skończony zbiór słów L

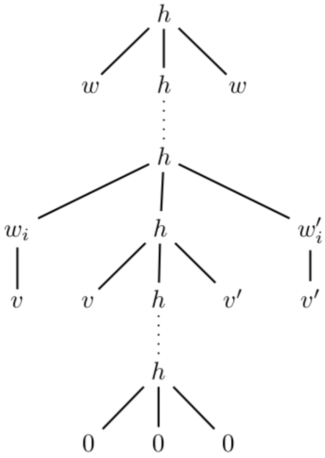
Przykład 2

- załóżmy, że mamy pewien skończony zbiór słów L
- możemy rozpoznawać ścieżki, na których kolejne litery tworzą pewne słowo z L

Przykład 2

- załóżmy, że mamy pewien skończony zbiór słów L
- możemy rozpoznawać ścieżki, na których kolejne litery tworzą pewne słowo z L
- możemy rozpoznawać ścieżki, na których kolejne litery tworzą pewne słowo z L^*

Przykład 3



Sprawdzanie pustości

Twierdzenie

Problem pustości dla *AWEDC* jest nierozstrzygalny.

Automaty z warunkami równościowymi pomiędzy braćmi

Definicja

Automat $A \in AWEDC$ nazwiemy automatem z warunkami równościowymi pomiędzy braćmi (AWCBB) jeśli wszystkie warunki mają postać $i = j$, gdzie $i, j \in \mathbb{N}_+$

Przykłady

- Pełne zrównoważone drzewa binarne nad $\Sigma = \{f_2, a_0\}$:

  $Q = Q_f = \{q\}$,

  $$\Delta = \{ \begin{array}{rcl} a & \rightarrow & q \\ f(q,q) & \xrightarrow{1=2} & q \end{array} \}$$

Przykłady

- Pełne zrównoważone drzewa binarne nad $\Sigma = \{f_2, a_0\}$:

  $$Q = Q_f = \{q\},$$

  $$\begin{array}{l} \Delta = \{ \\ \qquad \begin{array}{rcl} a & \rightarrow & q \\ f(q,q) & \overset{1=2}{\rightarrow} & q \end{array} \\ \} \end{array}$$

- $f(t, t)$

Własności domknięcia

Fakt

AWCBB jest podklasą AWEDC zamkniętą na operacje sumy, przecięcia i dopełnienia.

▸ konstrukcje dla AWEDC

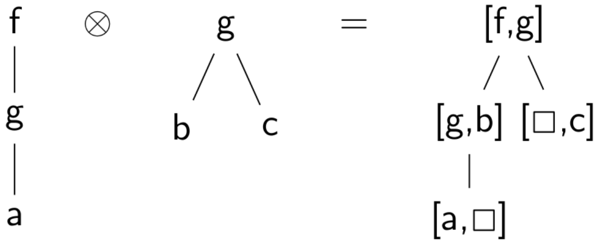
Własności domknięcia

	AWEDC	AWCBB	Automaty redukcyjne
przecięcie	$O(n^2)$	$O(n^2)$	
suma	$O(n)$	$O(n)$	
determinizacja	$O(2^n)$	$O(2^n)$	
dopełnienie	$O(2^n)$	$O(2^n)$	

RA

Projekcja i cylindryfikacja

Kodowanie pary drzew:



Projekcja i cylindryfikacja

Definicja

Niech $R \subseteq T(\Sigma)^n (n \geqslant 1), 1 \leqslant i \leqslant n$.

Projekcja i cylindryfikacja

Definicja

Niech $R \subseteq T(\Sigma)^n (n \geqslant 1), 1 \leqslant i \leqslant n$. I-tą projekcją R nazywamy relację $R_i \subseteq T(\Sigma)^{n-1}$ taką, że

Projekcja i cylindryfikacja

Definicja

Niech $R \subseteq T(\Sigma)^n (n \geqslant 1), 1 \leqslant i \leqslant n$. I-tą projekcją R nazywamy relację $R_i \subseteq T(\Sigma)^{n-1}$ taką, że

$$R_i(t_1, ..., t_{n-1}) \Leftrightarrow \exists t \in T(\Sigma)\ R(t_1, ..., t_{i-1}, t, t_i, ..., t_{n-1})$$

Projekcja i cylindryfikacja

Definicja

Niech $R \subseteq T(\Sigma)^n (n \geqslant 0), 1 \leqslant i \leqslant n+1$.

Projekcja i cylindryfikacja

Definicja

Niech $R \subseteq T(\Sigma)^n (n \geqslant 0), 1 \leqslant i \leqslant n+1$. I-tą cylindryfikacją R nazywamy relację $R^i \subseteq T(\Sigma)^{n+1}$ taką, że

Projekcja i cylindryfikacja

Definicja

Niech $R \subseteq T(\Sigma)^n (n \geqslant 0), 1 \leqslant i \leqslant n+1$. I-tą cylindryfikacją R nazywamy relację $R^i \subseteq T(\Sigma)^{n+1}$ taką, że

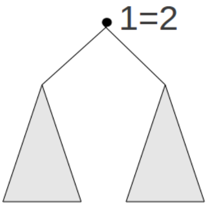
$$R^i(t_1, \ldots, t_{i-1}, t, t_i, \ldots, t_n) \Leftrightarrow R(t_1, \ldots, t_{i-1}, t_i, \ldots, t_n)$$

Projekcja i cylindryfikacja

- Jak rozpoznawać i-tą cylindryfikację $L(A) \ : \ A \in AWCBB$?

Projekcja i cylindryfikacja

- Jak rozpoznawać i-tą cylindryfikację $L(A) \ : \ A \in AWCBB$?
- Problem:



Projekcja i cylindryfikacja

- Jak rozpoznawać i-tą projekcję $L(A)$: $A \in AWCBB$?

Projekcja i cylindryfikacja

- Jak rozpoznawać i-tą projekcję $L(A) \ : \ A \in AWCBB$?
- Potencjalny problem:
  - $00(00, 01) \vDash 1 \neq 2$

Projekcja i cylindryfikacja

- Jak rozpoznawać i-tą projekcję $L(A) \ : \ A \in AWCBB$?
- Potencjalny problem:
  - $00(00, 01) \vDash 1 \neq 2$
  - $0(0, 0) \nvDash 1 \neq 2$

Projekcja i cylindryfikacja

Fakt

Klasa automatów z warunkami równościowymi na braciach nie jest zamknięta na operacje projekcji i cylindryfikacji.

Automaty redukcyjne

Definicja

Automat $A \in AWEDC$ nazwiemy automatem redukcyjnym, jeśli istnieje porządek na stanach automatu A taki, że:

Automaty redukcyjne

Definicja

Automat $A \in AWEDC$ nazwiemy automatem redukcyjnym, jeśli istnieje porządek na stanach automatu A taki, że:

- dla każdego przejścia $f(q_1, ..., q_n) \xrightarrow{c} q$ zachodzi $q \leqslant min(q_1, ..., q_n)$

Automaty redukcyjne

Definicja

Automat $A \in AWEDC$ nazwiemy automatem redukcyjnym, jeśli istnieje porządek na stanach automatu A taki, że:

- dla każdego przejścia $f(q_1, ..., q_n) \xrightarrow{c} q$ zachodzi $q \leqslant min(q_1, ..., q_n)$
- jeśli c zawiera równość, zachodzi $q < min(q_1, ..., q_n)$

Automaty redukcyjne

Definicja

Automat $A \in AWEDC$ nazwiemy automatem redukcyjnym, jeśli istnieje porządek na stanach automatu A taki, że:

- dla każdego przejścia $f(q_1, \ldots, q_n) \xrightarrow{c} q$ zachodzi $q \leqslant min(q_1, \ldots, q_n)$
- jeśli c zawiera równość, zachodzi $q < min(q_1, \ldots, q_n)$
- dla każdego ε – przejścia $q \to q'$ zachodzi $q' \leqslant q$

Własności domknięcia

Fakt 1.

Klasa automatów redukcyjnych jest zamknięta na sumę i przecięcie.

Własności domknięcia

Fakt 1.

Klasa automatów redukcyjnych jest zamknięta na sumę i przecięcie.

Fakt 2.

Nie istnieje algorytm determinizujący dla automatów redukcyjnych.

Własności domknięcia

Fakt 1.

Klasa automatów redukcyjnych jest zamknięta na sumę i przecięcie.

Fakt 2.

Nie istnieje algorytm determinizujący dla automatów redukcyjnych.

Fakt 3.

Nie wiadomo, czy klasa automatów redukcyjnych jest zamknięta na dopełnienie.