Limbaje Formale, Automate şi Compilatoare

Curs 3

2013-14

Curs 3

Automatul determinist echivalent

Teorema 1

Pentru orice automat A cu ϵ - tranziţii există un automat A' determinist echivalent cu A

Dacă $A = (Q, \Sigma, \delta, q_0, F)$ atunci $A' = (Q', \Sigma, \delta', q_0', F')$ unde:

- $Q' = 2^Q$
- $q_0' = Cl(q_0)$
- $\delta'(S, a) = Cl(\delta(S, a)) \quad S \in Q', a \in \Sigma$
- $S \in F' \Leftrightarrow S \cap F \neq \emptyset$

Automatul determinist echivalent

Teorema 1

Pentru orice automat A cu ϵ - tranziţii există un automat A' determinist echivalent cu A

Dacă $A = (Q, \Sigma, \delta, q_0, F)$ atunci $A' = (Q', \Sigma, \delta', q_0', F')$ unde:

- $Q' = 2^Q$
- $q_0' = Cl(q_0)$
- $\delta'(S, a) = Cl(\delta(S, a)) \quad S \in Q', a \in \Sigma$
- $S \in F' \Leftrightarrow S \cap F \neq \emptyset$

Au loc:

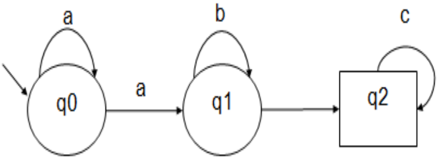
- $\delta'(q_0', w) = \hat{\delta}(q_0, w), \forall w \in \Sigma^*$
- $L(A') = L(A)$

Automatul determinist echivalent - algoritm

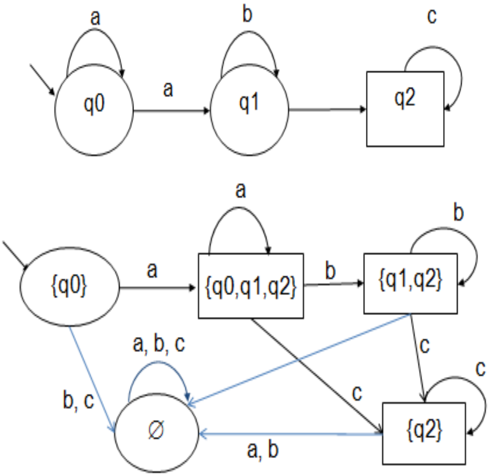
- Intrare: Automatul A (cu ϵ - tranziţii) ; $Cl(S)$
- Ieşire: Automatul determinist $A' = (Q', \Sigma, \delta', q'_0, F')$, echivalent cu A.

$q'_0 = Cl(\{q_0\})$; $Q' = \{q'_0\}$;
$marcat(q'_0) = false$; $F' = \emptyset$;
if $(q'_0 \cap F \neq \emptyset)$ then $F' = F' \cup \{q'_0\}$;
while $(\exists S \in Q' \&\& !marcat(S))$ { // S este nemarcat
    for $(a \in \Sigma)${
        $S' = Cl(\delta(S, a))$;
        $\delta'(S, a) = S'$;
        if $(S' \notin Q')${
          $Q' = Q' \cup \{S'\}$;
          $marcat(S') = false$;
          if $(S' \cap F \neq \emptyset)$ then $F' = F' \cup \{S'\}$;
        }
    }
    $marcat(S) = true$;
}

Exemplu



Exemplu



Curs 3

De la gramatici de tip 3 la automate finite

- Pentru orice gramatică G de tip 3 (în formă normală) există un automat A (nedeterminist) astfel ca $L(A) = L(G)$:

În gramatica G	În automatul A
T	$\Sigma = T$
N	$Q = N \cup \{f\}, F = \{f\}$
S	$q_0 = S$
$q \rightarrow ap$	$p \in \delta(q, a)$
$q \rightarrow a$	$f \in \delta(q, a)$
dacă $S \rightarrow \epsilon$	se adaugă S la F

De la automate finite la gramatici de tip 3

- Pentru orice automat finit (determinist) există o gramatică G de tip 3 astfel ca $L(A) = L(G)$:

În automatul A	În gramatica G
Σ	$T = \Sigma$
Q	$N = Q$
q_0	$S = q_0$
$\delta(q, a) = p$	$q \to ap$
$\delta(q, a) \in F$	$q \to a$
dacă $q_0 \in F$	se adaugă $q_0 \to \epsilon$

Curs 3

Stări accesibile

- Fie $A = (Q, \Sigma, \delta, q_0, F)$ automat finit determinist

Starea q este accesibilă în A dacă există un cuvânt $w \in \Sigma^*$ astfel încât $q = \delta(q_0, w)$.

Stări inseparabile

Fie $A = (Q, \Sigma, \delta, q_0, F)$ un automat finit determinist.

Definiţie 1

Stările q_1 şi q_2 sunt inseparabile în raport cu F, (notat $q_1 \rho q_2$) ddacă

$$\forall w \in \Sigma^* : \delta(q_1, w) \in F \Leftrightarrow \delta(q_2, w) \in F$$

Stări inseparabile

Fie $A = (Q, \Sigma, \delta, q_0, F)$ un automat finit determinist.

Definiţie 1

Stările q_1 şi q_2 sunt inseparabile în raport cu F, (notat $q_1 \rho q_2$) ddacă

$$\forall w \in \Sigma^* : \delta(q_1, w) \in F \Leftrightarrow \delta(q_2, w) \in F$$

- Dacă există $w \in \Sigma^*$ cu $\delta(q_1, w) \in F$ şi $\delta(q_2, w) \notin F$ (sau invers), stările q_1 şi q_2 sunt separabile (de către w), şi notăm $q_1 \ \underline{sep}\ q_2$
- $q_1 \ \underline{sep}\ q2 \Leftrightarrow \neg q_1 \rho q_2$.

Stări inseparabile

Fie $A = (Q, \Sigma, \delta, q_0, F)$ un automat finit determinist.

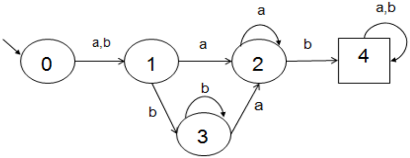
Definiţie 1

Stările q_1 şi q_2 sunt inseparabile în raport cu F, (notat $q_1 \rho q_2$) ddacă

$$\forall w \in \Sigma^* : \delta(q_1, w) \in F \Leftrightarrow \delta(q_2, w) \in F$$

- Dacă există $w \in \Sigma^*$ cu $\delta(q_1, w) \in F$ şi $\delta(q_2, w) \notin F$ (sau invers), stările q_1 şi q_2 sunt separabile (de către w), şi notăm $q_1 \underline{sep}\ q_2$
- $q_1 \underline{sep}\ q2 \Leftrightarrow \neg q_1 \rho q_2$.
- **Observaţie:** dacă $q_1 \in F$ şi $q_2 \notin F$, atunci $q_1 \underline{sep}\ q_2$

Exemplu



Automat minimal

Observaţii:

- Relatia ρ este relaţie de echivalenţă.
- $\exists a \in \Sigma : \delta(p, a) \underline{sep} \delta(q, a) \Longrightarrow p \underline{sep} q$.

Automat minimal

Observaţii:

- Relatia ρ este relaţie de echivalenţă.
- $\exists a \in \Sigma : \delta(p, a)\ \underline{sep}\ \delta(q, a) \Longrightarrow p\ \underline{sep}\ q$.

Teorema 2

Fie A un automat determinist cu toate stările accesibile. Daca toate stările din A sunt separabile în raport cu F, atunci nu există un alt automat A' cu număr mai mic de stări şi $L(A) = L(A')$.

Automatul minimal

Fie $A = (Q, \Sigma, \delta, q_0, F)$ un automat finit determinist si relaţia ρ.

- Dacă $\forall q_1, q_2 \in Q$: q_1 <u>*sep*</u> q_2, atunci A este minimal.
- Altfel, automatul minimal:

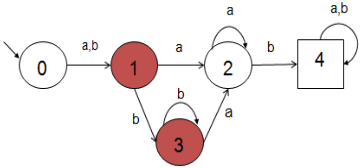
$$A_\rho = (Q/\rho, \Sigma, \delta_\rho, [q_0], F/\rho)$$

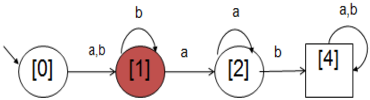
- Q/ρ - clasele de echivalenţă ale relaţiei ρ:

$$Q/\rho = \{[q] | q \in Q\}$$

- $\delta_\rho([q], a) = [\delta(q, a)]$
- $[q_0]$ clasa de echivalenţă în care se află starea q_0
- $F/\rho = \{[q] | q \in F\}$

Exemplu





Automatul minimal

Fie automatul minimal: $A_\rho = (Q/\rho, \Sigma, \delta_\rho, [q_0], F/\rho)$

- Q/ρ - clasele de echivalenţă ale relaţiei ρ:
- $\delta_\rho([q], a) = [\delta(q, a)]$
- $[q_0]$ clasa de echivalenţă în care se află starea q_0
- $F/\rho = \{[q] | q \in F\}$

Teorema 3

Fie automatul determinist A, cu toate stările accesibile. Automatul A_ρ construit ca mai sus este automatul cu număr minim de stări care acceptă limbajul $L(A)$.

Algoritm pentru determinarea relației ρ

- Fie $A = (Q, \Sigma, \delta, q_0, F)$, $Q = \{q_0, q_1, \ldots, q_n\}$
- Tablou *separabil*$[q_i, q_j]$:
  - *separabil*$[q_i, q_j] = 1$ ddacă q_i <u>*sep*</u> q_j (*separabil*$[q_i, q_j] = 0$ ddacă $q_i \rho q_j$)
  - inițial *separabil*$[q_i, q_j] = 1$ ddacă $q_i \in F, q_j \notin F$ (sau invers)
  - dacă există $a \in \Sigma$ cu $\delta(q_i, a)$ <u>*sep*</u> $\delta(q_j, a)$, atunci q_i <u>*sep*</u> q_j , adică : dacă *separabil*$[q_i, q_j] = 0$ și există $a \in \Sigma$ cu *separabil*$[\delta(q_i, a), \delta(q_j, a)] = 1$, atunci *separabil*$[q_i, q_j] = 1$

Algoritm pentru determinarea relaţiei ρ

- *lista*$[p, r]$: $(p \neq r)$
  - o pereche de stări inseparabile (q_i, q_j) ($separabil[q_i, q_j] = 0$) se adaugă la *lista*$[p, r]$ dacă:
    - există $a \in \Sigma$ astfel încât $p = \delta(q_i, a)$, $r = \delta(q_j, a)$
    - $separabil[p, r] = 0$

  $\Longleftrightarrow$

  - $(qi, qj) \rightarrow lista[\delta(q_i, a), \delta(qj, a)]$ dacă $separabil[q_i, q_j] = 0$ şi $separabil[\delta(q_i, a), \delta(qj, a)] = 0$

Algoritm pentru determinarea relaţiei ρ

- $lista[p,r]:(p \neq r)$
  - o pereche de stări inseparabile (q_i, q_j) ($separabil[q_i, q_j] = 0$) se adaugă la $lista[p, r]$ dacă:
    - există $a \in \Sigma$ astfel încât $p = \delta(q_i, a)$, $r = \delta(q_j, a)$
    - $separabil[p, r] = 0$

  $\Longleftrightarrow$

  - $(qi, qj) \rightarrow lista[\delta(q_i, a), \delta(qj, a)]$ dacă $separabil[q_i, q_j] = 0$ şi $separabil[\delta(q_i, a), \delta(qj, a)] = 0$
  - dacă p şi r devin la un moment dat separabile, atunci perechile de stări (q_i, q_j) din $lista[p, r]$ devin separabile:
    Fie $(q_i, q_j) \in lista[p, r] \Rightarrow \exists a : p = \delta(q_i, a), r = \delta(q_j, a)$

$$p \ \underline{sep} \ r \Leftrightarrow \delta(q_i, a) \ \underline{sep} \ \delta(q_j, a) \Longrightarrow q_i \ \underline{sep} \ q_j$$

Algoritm pentru determinarea relaţiei ρ

- Se iniţializează tabloul *separabil* (*separabil*$[q_i, q_j] = 1$, dacă $q_i \in F$, $q_j \notin F$ sau invers)
- Pentru orice q_i, q_j $(0 \leq i < j \leq n)$ cu *separabil*$[q_i, q_j] = 0$:

Algoritm pentru determinarea relaţiei ρ

- Se iniţializează tabloul *separabil* ($separabil[q_i, q_j] = 1$, dacă $q_i \in F$, $q_j \notin F$ sau invers)
- Pentru orice q_i, q_j $(0 \leq i < j \leq n)$ cu $separabil[q_i, q_j] = 0$:
  - Dacă există $a \in \Sigma$ cu $separabil[\delta(q_i, a), \delta(q_j, a)] = 1$, atunci:
    - $separabil[q_i, q_j] = 1$
    - trebuie modificat tabloul *separabil* pentru toate perechile de stări a căror separabilitate depinde de q_i, q_j (perechile de stări din $lista[q_i, q_j]$)

Algoritm pentru determinarea relaţiei ρ

- Se iniţializează tabloul *separabil* ($separabil[q_i, q_j] = 1$, dacă $q_i \in F$, $q_j \notin F$ sau invers)
- Pentru orice q_i, q_j $(0 \leq i < j \leq n)$ cu $separabil[q_i, q_j] = 0$:
  - Dacă există $a \in \Sigma$ cu $separabil[\delta(q_i, a), \delta(q_j, a)] = 1$, atunci:
    - $separabil[q_i, q_j] = 1$
    - trebuie modificat tabloul *separabil* pentru toate perechile de stări a căror separabilitate depinde de q_i, q_j (perechile de stări din $lista[q_i, q_j]$)
  - Altfel (pentru orice $a \in \Sigma$ are loc $separabil[\delta(q_i, a), \delta(q_j, a)] = 0$):
    - pentru orice $a \in \Sigma$ cu $\delta(q_i, a) \neq \delta(q_j, a)$ adaugă (q_i, q_j) la $lista[\delta(q_i, a), \delta(q_j, a)]$

Algoritm pentru determinarea relaţiei ρ

```
//initializarea tablourilor,
se marchează perechile F × (Q − F) si (Q − F) × F
for (i=0; i<=n-1; i++)
    for (j=i+1,j<=n; j++) {
        lista[qi,qj]=∅;
        if ((qi ∈ F  && qj ∉ F) || (qi ∉ F  && qj ∈ F))
            separabil[qi,qj]=1;
        else
            separabil[qi,qj]=0;
    }
```

```
for (i=0; i<=n-1; i++)
    for (j=i+1,j<=n; j++) {
    //se selecteaza doar starile inseparabile
        if (separabil[qi,qj]==0) {
            //daca exista a astfel incat δ(qi,a) sep δ(qj,a)
            //inseamna ca qi si qj sunt separabile
            if (∃a ∈ Σ : separabil[δ(qi,a),δ(qj,a)] == 1){
                // qi si qj devin separabile si la fel toate
                // perechile de stari dependente de qi,qj
                update_separabil(qi,qj);
            }
            else {
                for (a ∈ Σ && δ(qi,a) ≠ δ(qj,a))
                    adauga (qi,qj) la lista[δ(qi,a),δ(qj,a)]
            }
        }
    }
```

Algoritm pentru determinarea relaţiei ρ

```
// qi si qj devin separabile si la fel toate
// perechile de stari dependente de qi,qj
```

$update\_separabil(qi, qj)\{$

$\quad separabil[qi, qj] = 1;$

$\quad \texttt{for} \ ((q_i', q_j') \in lista[qi, qj])\{$

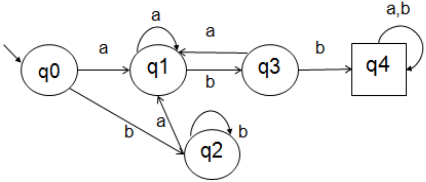
$\qquad \texttt{if} \ (separabil[q_i', q_j'] == 0)$

$\qquad\quad update\_separabil(q_i', q_j');$

$\quad \}$

$\}$

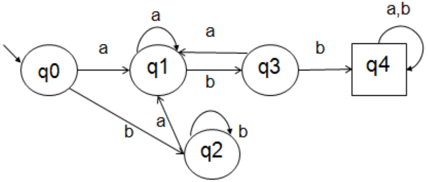
Exemplu



q1	q2	q3	q4	
0	0	0	1	q0
	0	0	1	q1
		0	1	q2
			1	q3

δ	a	b
q0	q1	q2
q1	q1	q3
q2	q1	q2
q3	q1	q4
q4	q4	q4

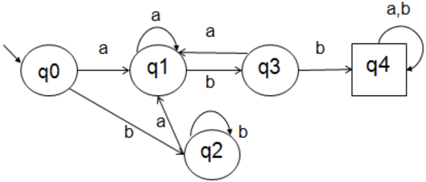
Exemplu



q1	q2	q3	q4	
0	0	(0) 1	1	q0
	0	0	1	q1
		0 (q0,q1)	1	q2
			1	q3

δ	a	b
q0	q1	q2
q1	q1	q3
q2	q1	q2
q3	q1	q4
q4	q4	q4

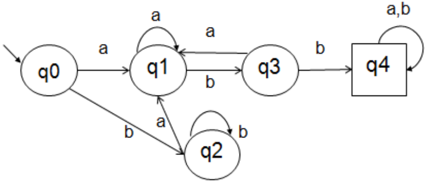
Exemplu



q1	q2	q3	q4	
0	0	1	1	q0
	0	(0) 1	1	q1
		0 (q1,q2) (q0,q1)	1	q2
			1	q3

δ	a	b
q0	q1	q2
q1	q1	q3
q2	q1	q2
q3	q1	q4
q4	q4	q4

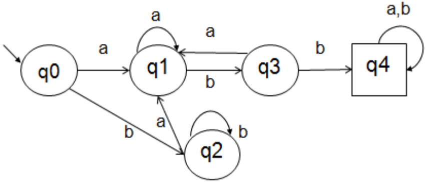
Exemplu



q1	q2	q3	q4	
(0) 1	0	1	1	q0
	(0) 1	1	1	q1
		(0)1 (q1,q2) (q0,q1)	1	q2
			1	q3

δ	a	b
q0	q1	q2
q1	q1	q3
q2	q1	q2
q3	q1	q4
q4	q4	q4

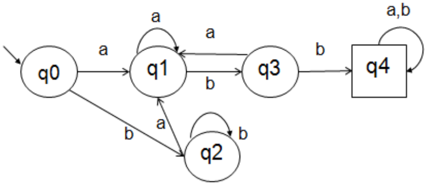
Exemplu

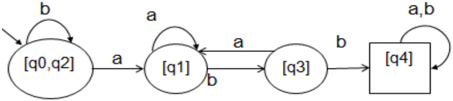


q1	q2	q3	q4	
1	0	1	1	q0
	1	1	1	q1
		1 (q1,q2) (q0,q1)	1	q2
			1	q3

δ	a	b
q0	q1	q2
q1	q1	q3
q2	q1	q2
q3	q1	q4
q4	q4	q4

Exemplu





Corectitudinea algoritmului

Teorema 4

Algoritmul se termină întotdeauna şi în final se obţine, pentru orice două stări q_i şi q_j, $0 \leq i < j \leq n$: separabil$[q_i, q_j] = 1$ ddacă q_i $\underline{sep}$ q_j

Corectitudinea algoritmului

Teorema 4

Algoritmul se termină întotdeauna şi în final se obţine, pentru orice două stări q_i şi q_j, $0 \leq i < j \leq n$: separabil$[q_i, q_j] = 1$ ddacă q_i <u>sep</u> q_j

($\Longleftarrow$) Se arată că:

$P(k)$: Pentru orice două stări q_i şi q_j ($0 \leq i < j \leq n$) separabile de către un cuvânt w cu $|w| \leq k$ ($\delta(q_i, w) \in F, \delta(q_j, w) \notin F$), are loc:

$$separabil[q_i, q_j] = 1.$$

Inducţie după $|w|$.

Corectitudinea algoritmului

Teorema 4

Algoritmul se termină întotdeauna şi în final se obţine, pentru orice două stări q_i şi q_j, $0 \leq i < j \leq n$: separabil$[q_i, q_j] = 1$ ddacă q_i $\underline{sep}$ q_j

($\Longrightarrow$) Se arată că:

pentru oricare două stări q_i, q_j ($0 \leq i < j \leq n$) pentru care *separabil*$[q_i, q_j] = 1$, are loc:

$$q_i \ \underline{sep} \ q_j.$$

Inducţie asupra momentului în care algoritmul face *separabil*$[q_i, q_j] = 1$.