

Limbaje Formale, Automate și Compilatoare

Curs 3

2020-21

Structura cursului

- 1 Automate finite cu ϵ -tranziții
- 2 Automatul determinist minimal

Curs 3

1 Automate finite cu ϵ -tranziții

2 Automatul determinist minimal

Automate finite cu ϵ -tranziii

Definiție 1

Un *automat finit cu ϵ -tranziii* este un 5-uplu $A = (Q, \Sigma, \delta, q_0, F)$, unde:

- Q, Σ, q_0 și F sunt definite ca în cazul automatelor finite deterministe
- δ este o funcție, $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow 2^Q$, numită funcția de tranziție

Observație:

- A este *automat nedeterminist*, dacă $\delta(q, \epsilon) = \emptyset, \forall q \in Q$
- A este *automat determinist*, dacă, în plus:

$$|\delta(q, a)| = 1, \forall q \in Q, \forall a \in \Sigma$$

Extensia lui δ la cuvinte

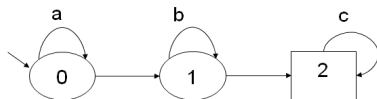
- $Cl(q)$ -mulțimea stărilor la care se poate ajunge prin ϵ -tranziții:
 - $q \in Cl(q)$
 - $q' \in Cl(q) \Rightarrow \delta(q', \epsilon) \subseteq Cl(q)$
- Dacă $S \subseteq Q$, atunci notăm:

$$Cl(S) = \bigcup_{q \in S} Cl(q)$$

- Extensia lui δ la cuvinte: $\hat{\delta} : Q \times \Sigma^* \rightarrow 2^Q$
 - 1 $\hat{\delta}(q, \epsilon) = Cl(q), \forall q \in Q;$
 - 2 $\hat{\delta}(q, ua) = Cl(\delta(\hat{\delta}(q, u), a)), \forall q \in Q, \forall u \in \Sigma^*, \forall a \in \Sigma.$

Extensia lui δ la cuvinte

- $\hat{\delta}(q, a) = Cl(\delta(Cl(q), a)), \forall q \in Q, \forall a \in \Sigma$



- În cazul automatelor cu ϵ - tranziții vom păstra notația $\hat{\delta}$ pentru extensie pentru că, în general, $\hat{\delta}(q, \epsilon) \neq \delta(q, \epsilon)$ și $\hat{\delta}(q, a) \neq \delta(q, a), a \in \Sigma$.
- $\hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q, u), v), \forall q \in Q, \forall u, v \in \Sigma^*$

Limbajul acceptat

Definiție 2

Limbajul acceptat (recunoscut) de automatul cu ϵ -tranziții

$A = (Q, \Sigma, \delta, q_0, F)$ este mulțimea :

$$L(A) = \{w \mid w \in \Sigma^*, \hat{\delta}(q_0, w) \cap F \neq \emptyset\}.$$

- Un cuvânt w este recunoscut de un automat A dacă, după citirea în întregime a cuvântului w , automatul (pornind din starea inițială q_0) poate să ajungă într-o stare finală.

Automatul determinist echivalent

Teorema 1

Pentru orice automat A cu ϵ - tranziii 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:

- $Q' = 2^Q$
- $q'_0 = Cl(q_0)$
- $\delta'(S, a) = Cl(\bigcup_{s \in S} \delta(s, a)) \quad S \in Q', a \in \Sigma$
- $S \in F' \Leftrightarrow S \cap F \neq \emptyset$

Automatul determinist echivalent

Teorema 1

Pentru orice automat A cu ϵ - tranzitii 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:

- $Q' = 2^Q$
- $q'_0 = Cl(q_0)$
- $\delta'(S, a) = Cl(\bigcup_{s \in S} \delta(s, a)) \quad S \in Q', a \in \Sigma$
- $S \in F' \Leftrightarrow S \cap F \neq \emptyset$

Au loc:

- $\delta'(q'_0, w) = \hat{\delta}(q_0, w), \forall w \in \Sigma^*$
- $L(A') = L(A)$

Automatul determinist echivalent - algoritm

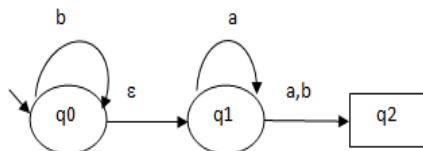
- Intrare: Automatul A (cu ϵ - tranziții) ; $Cl(S)$
- Ieșire: Automatul determinist $A' = (Q', \Sigma, \delta', q'_0, F')$, echivalent cu A.

```

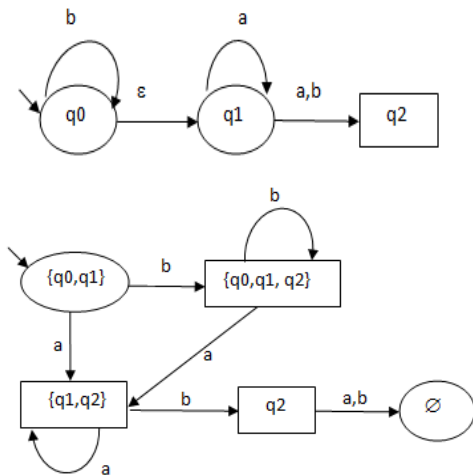
 $q'_0 = Cl(\{q_0\}); Q' = \{q'_0\};$ 
 $marcat(q'_0) = false; F' = \emptyset;$ 
if ( $q'_0 \cap F \neq \emptyset$ ) then  $F' = F' \cup \{q'_0\};$ 
while ( $\exists S \in Q' \&\&!marcat(S)$ ) {
    for ( $a \in \Sigma$ ) {
         $S' = Cl(\bigcup_{s \in S} \delta(s, a));$ 
         $\delta'(S, a) =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

- 1 Automate finite cu ϵ -tranziții
- 2 **Automatul determinist minimal**

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.

Definiție 3

Stările q_1 și 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.

Definiție 3

Stările q_1 și 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 $\delta(q_1, w) \in F$ și $\delta(q_2, w) \notin F$ (sau invers), stările q_1 și q_2 sunt *separabile* (de către w), și notăm $q_1 \underline{\text{sep}} q_2$
- $q_1 \underline{\text{sep}} q_2 \Leftrightarrow \neg q_1 \rho q_2$.

Stări inseparabile

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

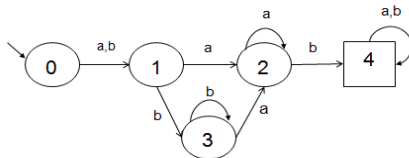
Definiție 3

Stările q_1 și 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 $\delta(q_1, w) \in F$ și $\delta(q_2, w) \notin F$ (sau invers), stările q_1 și q_2 sunt *separabile* (de către w), și notăm $q_1 \underline{\text{sep}} q_2$
- $q_1 \underline{\text{sep}} q_2 \Leftrightarrow \neg q_1 \rho q_2$.
- **Observație:** dacă $q_1 \in F$ și $q_2 \notin F$, atunci $q_1 \underline{\text{sep}} q_2$

Exemplu



Automat minimal

Observații:

- Relatia ρ este relație de echivalență.
- $\exists a \in \Sigma : \delta(p, a) \underline{sep} \delta(q, a) \implies p \underline{sep} q$.

Automat minimal

Observații:

- Relatia ρ este relație de echivalență.
- $\exists a \in \Sigma : \delta(p, a) \text{ sep } \delta(q, a) \implies p \text{ sep } q$.

Teorema 2

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

Automatul minimal

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

- Dacă $\forall q_1, q_2 \in Q : q_1 \text{ sep } q_2$, atunci A este minimal.
- Altfel, automatul minimal:

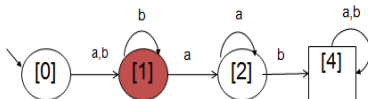
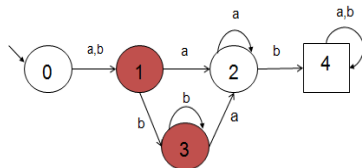
$$A_\rho = (Q/\rho, \Sigma, \delta_\rho, [q_0], F/\rho)$$

- Q/ρ - clasele de echivalență ale relației ρ :

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

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

Exemplu



Automatul minimal

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

- Q/ρ - clasele de echivalență ale relației ρ :
- $\delta_\rho([q], a) = [\delta(q, a)]$
- $[q_0]$ clasa de echivalență 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 $A = (Q, \Sigma, \delta, q_0, F)$, $Q = \{q_0, q_1, \dots, q_n\}$
- Tablou *separabil* $[q_i, q_j]$:
 - *separabil* $[q_i, q_j] = 1$ ddacă q_i sep 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)
 - Se va completa apoi *separabil* $[q_i, q_j]$, pentru $0 \leq i < j \leq n$, utilizând doar valorile calculate anterior în *separabil* și funcția δ .

Algoritm pentru determinarea relației ρ

- $lista[p, r] : (p \neq r)$
 - definită pentru perechi de stări cu $separabil[p, r] = 0$
 - $lista[p, r] = \{(q_i, q_j) | separabil[q_i, q_j] = 0 \wedge \text{exista } a \in \Sigma : p = \delta(q_i, a), r = \delta(q_j, a), (q_i, q_j) \neq (p, r)\}$
 - $lista$ se completează pe măsură ce se completează $separabil$

Algoritm pentru determinarea relației ρ

- Se inițializează 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 relației ρ

- Se inițializează 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 relației ρ

- Se inițializează 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 relației ρ

```
//initializarea tablourilor,
se marchează perechile  $F \times (Q - F)$  si  $(Q - F) \times F$ 
1.for (i=0; i<=n-1; i++)
2.    for (j=i+1, j<=n; j++) {
3.        lista[qi,qj]= $\emptyset$ ;
4.        if (( $qi \in F$  &&  $qj \notin F$ ) || ( $qi \notin F$  &&  $qj \in F$ ))
5.            separabil[qi,qj]=1;
6.        else
7.            separabil[qi,qj]=0;
8.    }
```

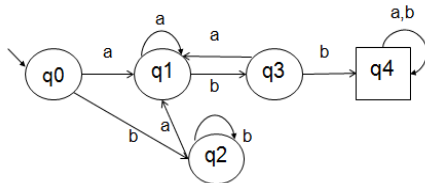
```

9. for (i=0; i<=n-1; i++)
10.   for (j=i+1, j<=n; j++) {
        //se selecteaza doar starile inseparabile
11.     if (separabil[qi,qj]==0) {
            //daca exista a astfel incat  $\delta(qi, a) \neq \delta(qj, a)$ 
            //inseamna ca qi si qj sunt separabile
12.     if ( $\exists a \in \Sigma : separabil[\delta(qi, a), \delta(qj, a)] == 1$ ) {
            // qi si qj devin separabile si la fel toate
            // perechile de stari dependente de qi,qj
13.       update_separabil(qi, qj);
14.     }
15.     else {
16.       for ( $a \in \Sigma : \delta(qi, a) \neq \delta(qj, a) \&\& (qi, qj) \neq (\delta(qi, a), \delta(qj, a))$ )
17.         adauga (qi, qj) la lista[ $\delta(qi, a), \delta(qj, a)$ ]
18.     }
19.   }
20. }
```

Algoritm pentru determinarea relației ρ

```
// qi si qj devin separabile si la fel toate
// perechile de stari dependente de qi,qj
update_separabil(qi, qj){
    separabil[qi, qj] = 1;
    for ((q'_i, q'_j) ∈ lista[qi, qj]){
        if (separabil[q'_i, q'_j] == 0)
            update_separabil(q'_i, q'_j);
    }
}
```

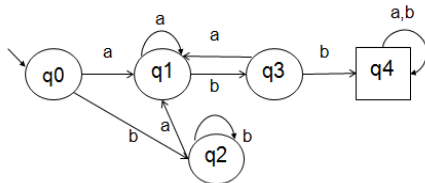
Exemplu



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

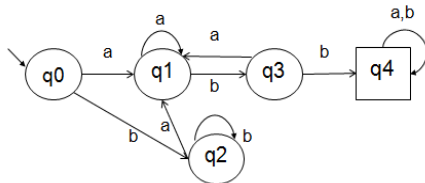
Exemplu



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

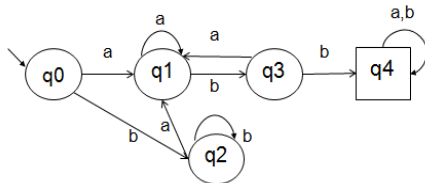
Exemplu



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

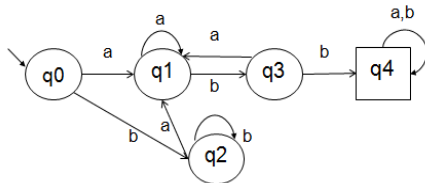
Exemplu



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

