Limbaje Formale, Automate şi Compilatoare

Curs 2

2014-15

Curs 2

Fie L, L_1, L_2 limbaje regulate: există gramaticile G, G_1, G_2 de tip 3 astfel ca $L = L(G), L_1 = L(G_1)$ şi $L_2 = L(G_2)$.

Atunci, următoarele limbaje sunt de asemenea regulate:

1. $L_1 \cup L_2$
2. $L_1 \cdot L_2$
3. L^*
4. L^R
5. $L_1 \cap L_2$
6. $L_1 \setminus L_2$

Închiderea la operaţia de iteraţie

Fie L limbaj de tip 3 (regulat).

Fie $G = (N, T, S, P)$ de tip 3, în formă normală, care genereaza L ($L = L(G)$).

Presupunem ca simbolul de start S nu apare in partea dreaptă a vreunei reguli.

Gramatica $G' = (N, T, S, P')$ unde P' consta din

- reguli $A \rightarrow aB$ din P
- reguli $A \rightarrow aS$, pentru orice regula $A \rightarrow a$ din P
- regula $S \rightarrow \epsilon$

este de tip 3 si generează L^*

Închiderea la intersecţie

Fie L_1, L_2 limbaje de tip 3 (regulate).

Fie $G_1 = (N_1, T_1, S_1, P_1)$ si $G_2 = (N_2, T_2, S_2, P_2)$ gramatici de tip 3, în formă normală, cu $L_1 = L(G_1)$, $L_2 = L(G_2)$.

Gramatica $G = (N_1 \times N_2, T_1 \cap T_2, (S_1, S_2), P)$, unde P constă din:

- $(S_1, S_2) \to \epsilon$, dacă $S_1 \to \epsilon \in P_1$ şi $S_2 \to \epsilon \in P_2$
- $(A_1, B_1) \to a(A_2, B_2)$, dacă $A_1 \to aA_2 \in P_1$ şi $B_1 \to aB_2 \in P_2$
- $(A_1, A_2) \to a$, dacă $A_1 \to a \in P_1$ şi $A_2 \to a \in P_2$

este de tip 3 şi generează limbajul $L_1 \cap L_2$

Închiderea la operaţia de oglindire

Fie L limbaj de tip 3 (regulat).

Fie $G = (N, T, S, P)$ de tip 3, în formă normală, care generează L ($L = L(G)$)

Presupunem ca simbolul de start S nu apare in partea dreaptă a vreunei reguli.

Gramatica $G' = (N, T, S', P')$ unde P' constă din

- reguli $S' \to aA$, pentru orice regulă $A \to a$ din P
- reguli $B \to aA$ pentru orice regulă $A \to aB$ din P
- regula $S \to \epsilon$
- regula $S' \to a$, pentru orice regulă $S \to a$ din P ($a \in T \cup \{\epsilon\}$)

este de tip 3 şi generează L^R

Curs 2

Lema Bar-Hillel (lema de pompare)

Lema 2.1

Fie L un limbaj de tip 3. Există un număr m astfel încât oricare ar fi cuvântul $w \in L$ cu $|w| \geq m$, acesta are o descompunere de forma $w = xyz$, unde $0 < |y| \leq m$, şi $xy^iz \in L$ oricare ar fi $i \geq 0$.

Fie $G = (N, T, S, P)$ astfel ca $L(G) = L$. Dacă $|N|$ este numărul simbolurilor din N , $m = |N| + 1$, se arată că are loc proprietatea enunţată:

Fie $w = a_1a_2 \ldots a_n, n \geq m \Rightarrow n \geq |N| + 1$

$S \Rightarrow a_1A_1 \Rightarrow a_1a_2A_2 \Rightarrow \ldots \Rightarrow a_1a_2 \ldots a_k\underline{A_k} \Rightarrow \ldots \Rightarrow a_1a_2 \ldots a_ka_{k+1} \ldots a_s\underline{A_s} \Rightarrow \ldots \Rightarrow$

$a_1a_2 \ldots a_ka_{k+1} \ldots a_sa_{s+1} \ldots a_{n-1}A_{n-1} \Rightarrow a_1a_2 \ldots a_ka_{k+1} \ldots a_sa_{s+1} \ldots a_{n-1}a_n$

$A_k = A_s$

Demonstrație

$w = a_1 a_2 \ldots a_n,\ n \geq m$

$S \Rightarrow a_1 A_1 \Rightarrow a_1 a_2 A_2 \Rightarrow \ldots \Rightarrow a_1 a_2 \ldots a_k A_k \Rightarrow \ldots \Rightarrow$
$a_1 a_2 \ldots a_k a_{k+1} \ldots a_s A_k \Rightarrow \ldots \Rightarrow a_1 a_2 \ldots a_k a_{k+1} \ldots a_s a_{s+1} \ldots a_{n-1} a_n$

Atunci:

- $S \Rightarrow^* a_1 a_2 \ldots a_k A_k$
- $A_k \Rightarrow^* a_{k+1} \ldots a_s A_k$ şi
- $A_k \Rightarrow^* a_{s+1} \ldots a_{n-1} a_n$

Fie $x = a_1 a_2 \ldots a_k$, $y = a_{k+1} \ldots a_s$ şi $z = a_{s+1} \ldots a_{n-1} a_n$

- $S \Rightarrow^* x A_k$, $A_k \Rightarrow^* y A_k$ şi $A_k \Rightarrow^* z$

Demonstraţie

Fie $x = a_1 a_2 \ldots a_k$, $y = a_{k+1} \ldots a_s$ şi $z = a_{s+1} \ldots a_{n-1} a_n$

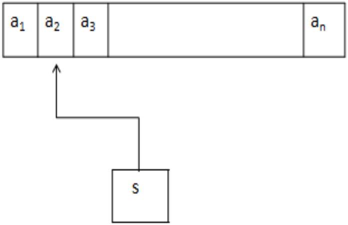
- $S \Rightarrow^* xA_k$
- $A_k \Rightarrow^* yA_k$ şi
- $A_k \Rightarrow^* z$

- Pentru $i = 0, xz \in L(G)$: $S \Rightarrow^* xA_k \Rightarrow^* xz$
- Pentru $i > 0$, $xy^i z \in L(G)$:
  $S \Rightarrow^* xA_k \Rightarrow^* xyA_k \Rightarrow^* xyyA_k \Rightarrow^* \ldots \Rightarrow^* xyy\ldots yA_k \Rightarrow^*$
  $xyy\ldots yz = xy^i z$

Curs 2

Automate finite

- Mecanism de recunoaştere (acceptare) pentru limbaje
- Limbaje de tip 3
- Mulţime finită de stări



Automate finite

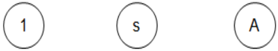
Definiţie 1

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

- Q *şi* Σ *sunt mulţimi finite, nevide, numite mulţimea stărilor respectiv alfabetul de intrare*
- $q_0 \in Q$ *este starea iniţială*
- $F \subseteq Q$ *este mulţimea stărilor finale*
- δ *este o funcţie ,* $\delta : Q \times \Sigma \to Q$*, numită funcţia de tranziţie*

Reprezentare prin diagrame(grafuri) de tranziţie

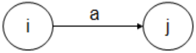
Stări:



Stare iniţială:

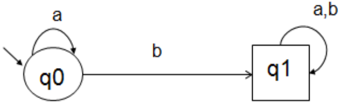
Stări finale:

Funcţia de tranziţie:



Reprezentare prin matricea de tranziţie

$$A = (\{q_0, q_1\}, \{a, b\}, \delta, q_0, \{q_1\})$$



Intrare Stare δ	a	b
q0	q0	q1
q1	q1	q1

Limbajul acceptat

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

- **Observaţii:**
  - $\hat{\delta}(q, a) = \delta(q, a), \forall q \in Q, \forall a \in \Sigma$
  - $\hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q, u), v), \forall q \in Q, \forall u, v \in \Sigma^*$

Limbajul acceptat

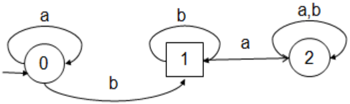
Definiţie 2

Limbajul acceptat (recunoscut) de automatul $A = (Q, \delta, \Sigma, q_0, F)$ este mulţimea :

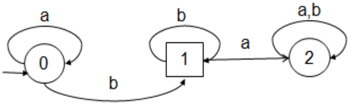
$$L(A) = \{w | w \in \Sigma^*, \hat{\delta}(q_0, w) \in F\}.$$

- Un cuvânt w este recunoscut de un automat A dacă, după citirea în întregime a cuvântului w, automatul (pornind din starea iniţială q_0) ajunge într-o stare finală.
- $\hat{\delta}(q, a) = \delta(q, a), \forall q \in Q, \forall a \in \Sigma$. Din acest motiv, $\hat{\delta}$ va fi notată de asemenea cu δ.
- Două automate A şi A' sunt echivalente, dacă $L(A) = L(A')$

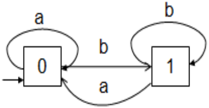
Exemple



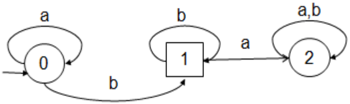
Exemple



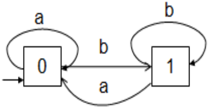
$L(A) = \{a^n b^m | n \geq 0, m \geq 1\}$



Exemple



$$L(A) = \{a^n b^m | n \geq 0, m \geq 1\}$$



$$L(A) = \{a, b\}^*$$

Exemple

Automate deterministe pentru:

- $L = \{w \in \{0,1\}^* | w \text{ conţine un număr par de } 0\}$
- $L = \{w \in \{0,1\}^* | w \text{ se termina cu } 11\}$

Curs 2

Automate finite nedeterministe

Definiţie 3

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

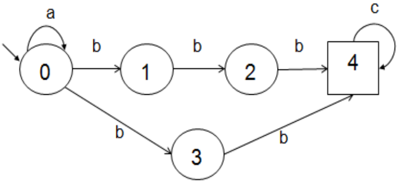
- Q, Σ, q_0 *şi* F *sunt definite ca în cazul automatelor finite deterministe*
- δ *este o funcţie ,* $\delta : Q \times \Sigma \to 2^Q$, *numită funcţia de tranziţie*

- **Observaţie:**

  A este automat determinist, dacă

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

Exemple



Intrare / Stare	a	b	c
0	{0}	{1,3}	Φ
1	Φ	{2}	Φ
2	Φ	{4}	Φ
3	Φ	{4}	Φ
4	Φ	Φ	{4}

Extensia lui δ la cuvinte

- Fie S mulţime de stări. Notăm $\delta(S, a) = \bigcup_{q \in S} \delta(q, a)$.
- Extensia lui δ la cuvinte $\hat{\delta} : Q \times \Sigma^* \to 2^Q$
  1. $\hat{\delta}(q, \epsilon) = \{q\}, \forall q \in Q$;
  2. $\hat{\delta}(q, ua) = \delta(\hat{\delta}(q, u), a), \forall q \in Q, \forall u \in \Sigma^*, \forall a \in \Sigma$.

Extensia lui δ la cuvinte

- Fie S mulţime de stări. Notăm $\delta(S, a) = \bigcup_{q \in S} \delta(q, a)$.
- Extensia lui δ la cuvinte $\hat{\delta} : Q \times \Sigma^* \to 2^Q$
  1. $\hat{\delta}(q, \epsilon) = \{q\}, \forall q \in Q$;
  2. $\hat{\delta}(q, ua) = \delta(\hat{\delta}(q, u), a)$, $\forall q \in Q, \forall u \in \Sigma^*, \forall a \in \Sigma$.
- **Observaţii:**
  - $\hat{\delta}(q, a) = \delta(q, a)$, $\forall q \in Q, \forall a \in \Sigma$
  - $\hat{\delta}(q, uv) = \hat{\delta}(\hat{\delta}(q, u), v)$, $\forall q \in Q, \forall u, v \in \Sigma^*$.

Limbajul acceptat

Definiţie 4

Limbajul acceptat (recunoscut) de automatul finit nedeterminist $A = (Q, \Sigma, \delta, q_0, F)$ este mulţimea :

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

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

Determinism = Nedeterminism

Teorema 1

Pentru orice automat nedeterminist A, există unul determinist A' echivalent.

Dacă $A = (Q, \Sigma, \delta, q_0, F)$ atunci $A' = (2^Q, \Sigma, \delta', Q_0, F')$ unde:

- $Q_0 = \{q_0\}$
- $F' = \{S | S \subseteq Q, S \cap F \neq \emptyset\}$
- $\delta'(S, a) = \bigcup_{s \in S} \delta(s, a)$ $(= \delta(S, a)), \forall S \in 2^Q$

- Pentru aplicaţii se construiesc doar stările accesibile din starea iniţială

Exemplu