Limbaje Formale, Automate şi Compilatoare

Curs 4

2016-17

Curs 4

1. Gramatici de tip 3 şi automate finite
2. Lema Bar-Hillel
3. Proprietăţi de închidere pentru clasa limbajelor de tip 3
4. Expresii regulate
5. Automatul echivalent cu o expresie regulată
   - Algoritm

Curs 4

1 Gramatici de tip 3 şi automate finite

2 Lema Bar-Hillel

3 Proprietăţi de închidere pentru clasa limbajelor de tip 3

4 Expresii regulate

5 Automatul echivalent cu o expresie regulată

- Algoritm

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	$q_0 = S$
$q \to ap$	$p \in \delta(q, a)$
$q \to a$	$f \in \delta(q, a)$
dacă $S \to \epsilon$	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	$S = q_0$
$\delta(q, a) = p$	$q \rightarrow ap$
$\delta(q, a) \in F$	$q \rightarrow a$
dacă $q_0 \in F$	se adaugă $q_0 \rightarrow \epsilon$

Curs 4

1 Gramatici de tip 3 şi automate finite

2 Lema Bar-Hillel

3 Proprietăţi de închidere pentru clasa limbajelor de tip 3

4 Expresii regulate

5 Automatul echivalent cu o expresie regulată
- Algoritm

Lema Bar-Hillel (lema de pompare)

Lema 2.1

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

Fie $A = (Q, \Sigma, \delta, q_0, F)$ astfel ca $L(A) = L$. Dacă $|Q| = n$ este numărul stărilor din N , fie $k = |Q| = n$, se arată că are loc proprietatea enunţată:

Fie $w = a_1 a_2 \dots a_m,\ m \geq k = n$

Fie $q_0 = \delta(q_0, \epsilon)$, $q_1 = \delta(q_0, a_1)$, $\dots$ $q_n = \delta(q_0, a_1 \dots a_n)$

Există două stări egale: $q_l = \delta(q_0, a_1 \dots a_l)$, $q_j = \delta(q_0, a_1 \dots a_j)$

$0 \leq l < j \leq n$.

Demonstraţie

$w = a_1 a_2 \ldots a_m$, $m \geq k = n$

Fie $x = a_1 a_2 \ldots a_l$, $y = a_{l+1} \ldots a_j$ şi $z = a_{j+1} \ldots a_{m-1} a_m$
$w = xyz$, cu $0 < |y| \leq k$.

- $q_l = \delta(q_0, a_1 \ldots a_l) = q_j = \delta(q_0, a_1 \ldots a_l a_{l+1} \ldots a_j) \Rightarrow$

$$\delta(q_0, x) = \delta(q_0, xy)$$

$xy^i z \in L(A), \forall i \geq 0$:

- $i = 0$: $\delta(q_0, xz) = \delta(\delta(q0, x), z) = \delta(\delta(q0, xy), z) = \delta(q0, xyz) \in F$ $(w = xyz \in L = L(A))$
- Presupunem că $xy^i z \in L$, pentru orice $i \leq r$ şi demonstrăm pentru $i = r + 1$

Curs 4

1. Gramatici de tip 3 şi automate finite
2. Lema Bar-Hillel
3. Proprietăţi de închidere pentru clasa limbajelor de tip 3
4. Expresii regulate
5. Automatul echivalent cu o expresie regulată
   - Algoritm

Închiderea la intersecţie

- Dacă $L_1, L_2 \in \mathcal{L}_3$, atunci $L_1 \cap L_2 \in \mathcal{L}_3$

Fie $A_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, F_1)$ şi $A_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, F_2)$ automate deterministe astfel încât $L_1 = L(A_1)$ şi $L_2 = L(A_2)$.
Automatul A (determinist) care recunoaşte $L_1 \cap L_2$:

$$A = (Q_1 \times Q_2, \Sigma_1 \cap \Sigma_2, \delta, (q_{01}, q_{02}), F_1 \times F_2)$$

$\delta((q_1, q_2), a)) = (q_1', q_2')$ ddacă

- $\delta_1(q_1, a) = q_1'$
- $\delta_2(q_2, a) = q_2'$

Închiderea la diferenţă

- Dacă $L \in \mathcal{L}_3$ atunci $\overline{L} = (\Sigma^* \setminus L) \in \mathcal{L}_3$

Fie $A = (Q, \Sigma, \delta, q_0, F)$ automat cu $L(A) = L$.

Automatul A' care recunoaşte $\overline{L} = \overline{L(A)}$:

$$A' = (Q, \Sigma, \delta, q_0, Q \setminus F)$$

Închiderea la diferenţă

- Dacă $L \in \mathcal{L}_3$ atunci $\overline{L} = (\Sigma^* \setminus L) \in \mathcal{L}_3$

Fie $A = (Q, \Sigma, \delta, q_0, F)$ automat cu $L(A) = L$.

Automatul A' care recunoaşte $\overline{L} = \overline{L(A)}$:

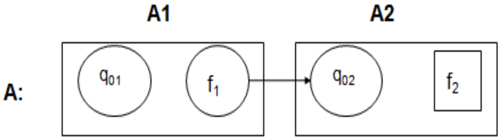
$$A' = (Q, \Sigma, \delta, q_0, Q \setminus F)$$

- Dacă $L_1, L_2 \in \mathcal{L}_3$ atunci $L_1 \setminus L_2 \in \mathcal{L}_3$: $L_1 \setminus L_2 = L_1 \cap \overline{L_2}$

Închiderea la produs

- Fie $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, \{f_1\})$ şi $A_2 = (Q_2, \Sigma, \delta_2, q_{02}, \{f_2\})$ automate cu o singură stare finală astfel încât $L_1 = L(A_1)$ şi $L_2 = L(A_2)$.
  Automatul A (cu ϵ-tranziţii) care recunoaşte $L_1 \cdot L_2$:

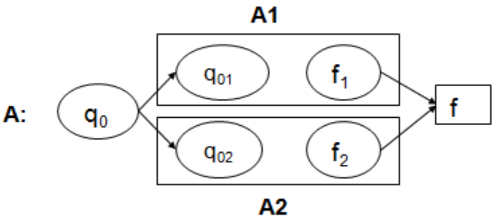
$$A = (Q_1 \cup Q_2, \Sigma, \delta, q_{01}, \{f_2\})$$



Închiderea la reuniune

- Fie $A_1 = (Q_1, \Sigma_1, \delta_1, q_{01}, \{f_1\})$ şi $A_2 = (Q_2, \Sigma_2, \delta_2, q_{02}, \{f_2\})$ automate cu o singură stare finală astfel încât $L_1 = L(A_1)$ şi $L_2 = L(A_2)$.
  Automatul A (cu ϵ-tranziţii) care recunoaşte $L_1 \cup L_2$:

$$A = (Q_1 \cup Q_2 \cup \{q_0, f\}, \Sigma_1 \cup \Sigma_2, \delta, q_0, \{f\})$$

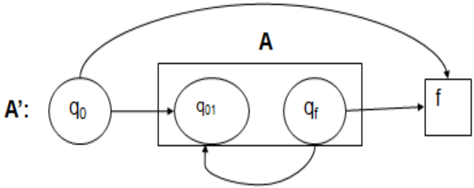


Închiderea la iteraţie

- Fie $A = (Q, \Sigma, \delta, q_{01}, \{f\})$ automat cu o singură stare finală astfel încât $L(A) = L$.

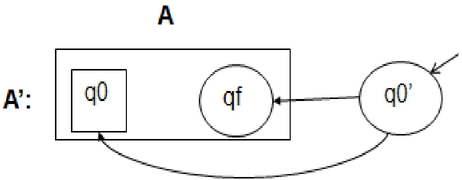
  Automatul A (cu ϵ-tranziţii) care recunoaşte L^* $(= L(A)^*)$:

$$A = (Q \cup \{q_0, f\}, \Sigma, \delta', q_0, \{f\})$$



Închiderea la operaţia de oglindire

- Fie $A = (Q, \Sigma, \delta, q_0, \{q_f\})$ automat cu o singură stare finală. Automatul A (cu ϵ-tranziţii) care recunoaşte $L(A)^R$:
  $A = (Q \cup \{q_0'\}, \Sigma, \delta', q_0', \{q_0\})$:
  - $\delta'(q_1, a) = q_2$ ddacă $\delta(q_2, a) = q_1$ (inversarea arcelor în graful de tranziţie)
  - $\delta'(q_0', \epsilon) = q_f$
  - dacă $\epsilon \in L(A)$, atunci $\delta'(q_0', \epsilon) = q_0$



Curs 4

1. Gramatici de tip 3 şi automate finite
2. Lema Bar-Hillel
3. Proprietăţi de închidere pentru clasa limbajelor de tip 3
4. **Expresii regulate**
5. Automatul echivalent cu o expresie regulată
   - Algoritm

Expresii regulate - definiţie

- Reprezentarea limbajelor de tip 3 prin expresii algebrice

Definiţie 1

Dacă Σ este un alfabet atunci o expresie regulată peste Σ se defineşte inductiv astfel:

- $\emptyset$, ϵ, *a* (*a* $\in \Sigma$) *sunt expresii regulate ce descriu respectiv limbajele* $\emptyset$, $\{\epsilon\}$, $\{a\}$.
- *Dacă* E, E_1, E_2 *sunt expresii regulate atunci:*
  - $(E_1|E_2)$ *este expresie regulată ce descrie limbajul* $L(E_1) \cup L(E2)$
  - $(E_1 \cdot E_2)$ *este expresie regulată ce descrie limbajul* $L(E_1)L(E_2)$
  - (E^*) *este expresie regulată ce descrie limbajul* $L(E)^*$

Expresii regulate - definiţie

- Reprezentarea limbajelor de tip 3 prin expresii algebrice

Definiţie 1

Dacă Σ este un alfabet atunci o expresie regulată peste Σ se defineşte inductiv astfel:

- $\emptyset$, ϵ, a ($a \in \Sigma$) *sunt expresii regulate ce descriu respectiv limbajele* $\emptyset$, $\{\epsilon\}$, $\{a\}$.
- *Dacă* E, E_1, E_2 *sunt expresii regulate atunci:*
  - $(E_1|E_2)$ *este expresie regulată ce descrie limbajul* $L(E_1) \cup L(E2)$
  - $(E_1 \cdot E_2)$ *este expresie regulată ce descrie limbajul* $L(E_1)L(E_2)$
  - (E^*) *este expresie regulată ce descrie limbajul* $L(E)^*$

- Ordinea de prioritate a operatorilor este $*$, $\cdot$, $|$

Exemple

- $(a|b)|(c|d) \longrightarrow \{a, b, c, d\}$
- $(0|1) \cdot (0|1) \longrightarrow \{00, 01, 10, 11\}$
- $a^*b^* \longrightarrow \{a^n b^k | n, k \geq 0\}$
- $(a|b)^* \longrightarrow \{a, b\}^*$
- $(0|1|2|...|9)(0|1|2...|9)^*$ descrie mulţimea întregilor fără semn
- $(a|b|c|...|z)(a|b|c|...|z|0|1|2...|9)^*$ descrie mulţimea identificatorilor

Două expresii regulate E_1, E_2 sunt echivalente, şi scriem $E_1 = E_2$ dacă $L(E_1) = L(E_2)$

Proprietăţi

- $(p|q)|r = p|(q|r)$
- $(pq)r = p(qr)$
- $p|q = q|p$
- $p \cdot \epsilon = \epsilon \cdot p = p$
- $p|\emptyset = p|p = p$
- $\emptyset \cdot p = p \cdot \emptyset = \emptyset$
- $p(q|r) = pq|pr$
- $(p|q)r = pr|qr$
- $\epsilon|pp^* = p^*$
- $\epsilon|p^*p = p^*$

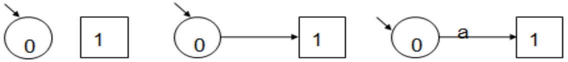
De la o expresie regulată la automatul finit

Teorema 1

Pentru orice expresie regulată E peste Σ există un automat finit (cu ϵ -tranziţii) A, astfel încât $L(A) = L(E)$.

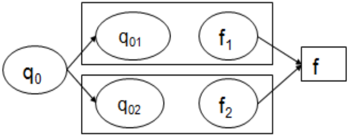
Demonstratie: inducţie structurală.

- Dacă $E \in \{\emptyset, \epsilon, a\}$ $(a \in \Sigma)$ atunci automatul corespunzător este respectiv:

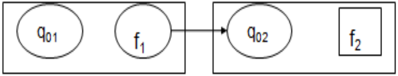


Demonstrație

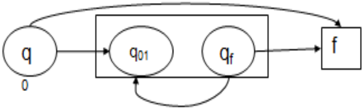
- $E = E_1 | E_2$



- $E = E_1 E_2$



- $E = E_1^*$



Reprezentarea expresiilor regulate sub formă de arbore

- **Intrare**: Expresia regulată $E = e_0 e_1 \dots e_{n-1}$
  Precedenţa operatorilor:
  prec($|$) = 1, prec($\cdot$) = 2, prec($*$) = 3 (prec(()= 0).
- **Ieşire**: Arborele asociat: t.
- **Metoda**: Se consideră două stive:
  - STIVA1 stiva operatorilor
  - STIVA2 stiva arborilor (care va conţine arborii parţiali construiţi)
  - Metoda $tree(r, tS, tD)$

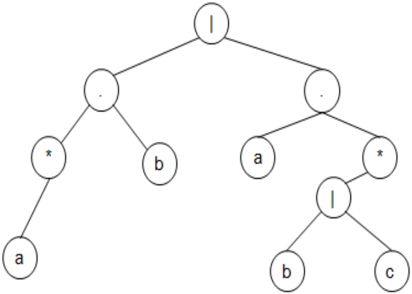
Algoritm

```
i = 0;
while(i < n) {
     c = e_i;
     switch(c) {
         case '(':  { STIVA1.push(c); break; }
         case simbol (din alfabet):  { STIVA2.push(tree(c,NULL,NULL)); break; }
         case operator:  {
              while (prec(STIVA1.top())>=prec(c))
                    build_tree();
              STIVA1.push(c); break;
         }
         case ')':  {
              do { build_tree();} while(STIVA1.top()!= '(');
              STIVA1.pop(); break;
         }
     }
     i++;
}
while(STIVA1.not_empty()) build_tree();
t = STIVA2.pop();
```

Algoritm

```
build_tree()
     op = STIVA1.pop();
     t1 = STIVA2.pop();
     switch (op) {
            case '*':  {
                 t = tree(op, t1, NULL);
                 STIVA2.push(t); break;
            }
            case'|', '·':  {
                t2 = STIVA2.pop();
                t = tree(op, t1, t2);
                STIVA2.push(t); break;
            }
}
```

Exemplu



$a^* \cdot b | a \cdot (b|c)^*$

Curs 4

1 Gramatici de tip 3 şi automate finite

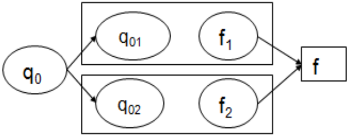
2 Lema Bar-Hillel

3 Proprietăţi de închidere pentru clasa limbajelor de tip 3

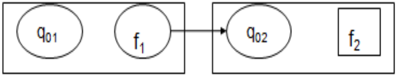
4 Expresii regulate

5 Automatul echivalent cu o expresie regulată
- Algoritm

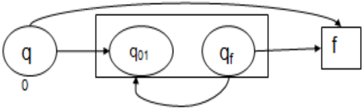
Automatul echivalent cu o expresie regulată



- $E = E_1|E_2$



- $E = E_1E_2$



- $E = E_1^*$

Observaţii

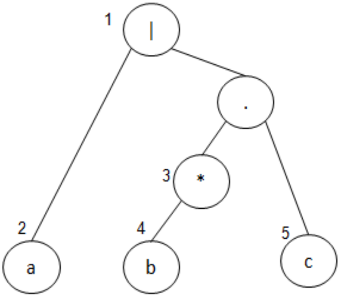
- pentru orice apariţie a unui simbol din Σ, cât şi pentru ϵ, dacă acesta apare explicit în E, este nevoie de 2 stări în automatul construit.
- fiecare din apariţiile operatorilor $|$ şi $*$ dintr-o expresie regulată E introduce două noi stări în automatul construit
- operatorul $\cdot$ nu introduce alte stări
- dacă n este numărul de simboluri din E iar m este numărul de paranteze împreună cu apariţiile simbolului $\cdot$, atunci numărul stărilor automatului echivalent cu E este $p = 2(n - m)$.

Algoritm

- **Intrare**: Expresia regulată E cu n simboluri dintre care m sunt paranteze şi apariţii ale operatorului produs;
- **Ieşire**:Automatul (cu $p = 2(n - m)$ stări) cu ϵ - tranziţii echivalent cu E
- **Metoda**:

1. Se construieşte arborele ataşat expresiei E;
2. Se parcurge arborele în preordine şi se ataşează nodurilor vizitate, exceptând pe cele etichetate cu produs , respectiv numerele $1, 2, \ldots, n - m$;

Exemplu



$E = a|b^* \cdot c$

3. Se parcurge arborele în postordine şi se ataşează fiecărui nod N o pereche de numere $(N.i, N.f)$ care reprezintă starea iniţială respectiv finală a automatului echivalent cu expresia corespunzătoare subarborelui cu rădăcina N, astfel:

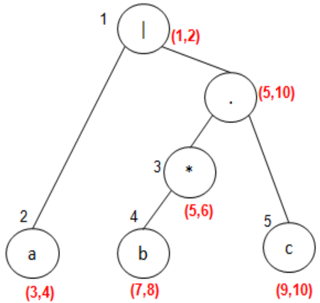
   - Dacă nodul are numărul k (de la pasul 2) atunci:

$$N.i = 2k - 1, N.f = 2k;$$

   - Dacă nodul este etichetat cu produs şi S este fiul stâng al lui N, iar D fiul drept, atunci:

$$N.i = S.i \text{ iar } N.f = D.f$$

Exemplu



$E = a|b^* \cdot c$

4. Se parcurge din nou arborele obţinut în postordine.

   Dacă N este nodul curent iar S si D sunt fii sai, atunci, în funcţie de eticheta lui N, se execută următoarele:

4. Se parcurge din nou arborele obţinut în postordine.

   Dacă N este nodul curent iar S si D sunt fii sai, atunci, în funcţie de eticheta lui N, se execută următoarele:

   - Dacă N este etichetat cu *a* (deci este frunza):

$$\delta(N.i, a) = N.f$$

4. Se parcurge din nou arborele obţinut în postordine.
   Dacă N este nodul curent iar S si D sunt fii sai, atunci, în funcţie de eticheta lui N, se execută următoarele:

   - Dacă N este etichetat cu *a* (deci este frunza):

$$\delta(N.i, a) = N.f$$

   - Dacă N este etichetat cu $|$:

$$\delta(N.i, \epsilon) = \{S.i, D.i\},$$

$$\delta(S.f, \epsilon) = N.f, \ \delta(D.f, \epsilon) = N.f$$

4. Se parcurge din nou arborele obţinut în postordine.

   Dacă N este nodul curent iar S si D sunt fii sai, atunci, în funcţie de eticheta lui N, se execută următoarele:

   - Dacă N este etichetat cu *a* (deci este frunza):

     $$\delta(N.i, a) = N.f$$

   - Dacă N este etichetat cu $|$:

     $$\delta(N.i, \epsilon) = \{S.i, D.i\},$$

     $$\delta(S.f, \epsilon) = N.f, \ \delta(D.f, \epsilon) = N.f$$

   - Dacă N este etichetat cu $\cdot$:

     $$\delta(S.f, \epsilon) = D.i$$

4. Se parcurge din nou arborele obţinut în postordine.

   Dacă N este nodul curent iar S si D sunt fii sai, atunci, în funcţie de eticheta lui N, se execută următoarele:

   - Dacă N este etichetat cu *a* (deci este frunza):

   $$\delta(N.i, a) = N.f$$

   - Dacă N este etichetat cu $|$:

   $$\delta(N.i, \epsilon) = \{S.i, D.i\},$$

   $$\delta(S.f, \epsilon) = N.f,\ \delta(D.f, \epsilon) = N.f$$

   - Dacă N este etichetat cu $\cdot$:

   $$\delta(S.f, \epsilon) = D.i$$

   - Dacă N este etichetat cu $*$ (D nu există în acest caz):

   $$\delta(N.i, \epsilon) = \{S.i, N.f\},$$

   $$\delta(S.f, \epsilon) = \{S.i, N.f\}$$

4. Se parcurge din nou arborele obţinut în postordine.
   Dacă N este nodul curent iar S si D sunt fii sai, atunci, în funcţie de eticheta lui N, se execută următoarele:

   - Dacă N este etichetat cu a (deci este frunza):

   $$\delta(N.i, a) = N.f$$

   - Dacă N este etichetat cu $|$:

   $$\delta(N.i, \epsilon) = \{S.i, D.i\},$$

   $$\delta(S.f, \epsilon) = N.f,\ \delta(D.f, \epsilon) = N.f$$

   - Dacă N este etichetat cu $\cdot$:

   $$\delta(S.f, \epsilon) = D.i$$

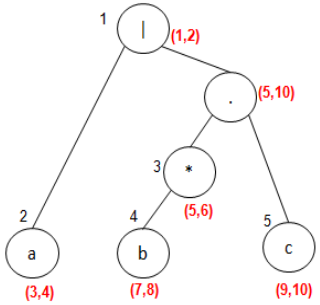
   - Dacă N este etichetat cu $*$ (D nu există în acest caz):

   $$\delta(N.i, \epsilon) = \{S.i, N.f\},$$

   $$\delta(S.f, \epsilon) = \{S.i, N.f\}$$

5. Starea iniţială a automatului este $N.i$, starea finală $N.f$, unde N este nodul rădăcină;

Exemplu



$E = a|b^* \cdot c$

Exemplu

δ	a	b	c	ϵ
1	$\emptyset$	$\emptyset$	$\emptyset$	$\{3,5\}$
2	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
3	4	$\emptyset$	$\emptyset$	$\emptyset$
4	$\emptyset$	$\emptyset$	$\emptyset$	$\{2\}$
5	$\emptyset$	$\emptyset$	$\emptyset$	$\{6,7\}$
6	$\emptyset$	$\emptyset$	$\emptyset$	$\{9\}$
7	$\emptyset$	8	$\emptyset$	$\emptyset$
8	$\emptyset$	$\emptyset$	$\emptyset$	$\{6,7\}$
9	$\emptyset$	$\emptyset$	10	$\emptyset$
10	$\emptyset$	$\emptyset$	$\emptyset$	$\{2\}$

Corectitudinea algoritmului

Teorema 2

Algoritmul descris este corect: automatul cu ϵ - tranziţii obţinut este echivalent cu expresia regulată E.

Demonstraţie:

- Modul în care au fost alese perechile (i, f) de stări pentru fiecare nod al arborelui construit corespunde construcţiilor din teorema 1.
- Deasemenea, tranziţiile care se definesc în pasul 5 al algoritmului urmăresc construcţia din teorema 1.

Automatul obţinut este echivalent cu expresia dată la intrare.