Limbaje Formale, Automate şi Compilatoare

Curs 4

2019-19

Curs 4

1. Corectitudinea algoritmului pentru determinarea relaţiei ρ
2. Gramatici de tip 3 şi automate finite
3. Proprietăţi de închidere pentru clasa limbajelor de tip 3
4. Expresii regulate

Curs 4

1. Corectitudinea algoritmului pentru determinarea relaţiei ρ
2. Gramatici de tip 3 şi automate finite
3. Proprietăţi de închidere pentru clasa limbajelor de tip 3
4. Expresii regulate

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)&& (qi,qj) ≠ (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){
      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);
      }
}
```

Corectitudinea algoritmului

Teorema 1

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 sep q_j

Corectitudinea algoritmului

Teorema 1

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 sep 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 1

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 $separabil[q_i, q_j] = 1$.

Curs 4

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 (nedeterminist) 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	$S = q_0$
$p \in \delta(q, a)$	$q \to ap$
$\delta(q, a) \cap F \neq \emptyset$	$q \to a$
dacă $q_0 \in F$	se adaugă $q_0 \to \epsilon$

Curs 4

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 $\overline{L} = \overline{L(A)}$:

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

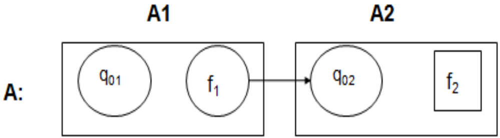
- Dacă $L_1, L_2 \in \mathcal{L}_3$ atunci $L_1 \setminus L_2 \in \mathcal{L}_3 : L_1 \setminus L_2 = L_1 \cap \overline{L_2}$

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 $L_1 \cdot L_2$:

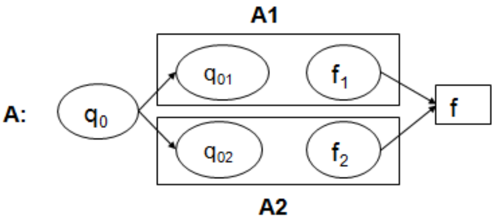
$$A = (Q_1 \cup Q_2, \Sigma, \delta, q_{01}, \{f_2\})$$



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 $L_1 \cup L_2$:

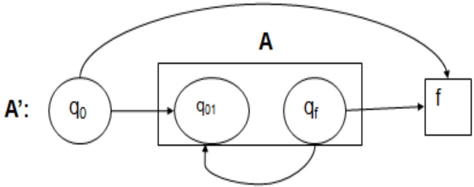
$$A = (Q_1 \cup Q_2 \cup \{q_0, f\}, \Sigma_1 \cup \Sigma_2, \delta, q_0, \{f\})$$



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^* $(= L(A)^*)$:

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



Curs 4

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

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