Limbaje regulate
Curs de pregatire, Lot 2006

Florin Manea

Facultatea de Matematică şi Informatică, Universitatea din Bucureşti

Str. Academiei 14, 70109, Bucureşti, România

E-mail: flmanea@funinf.cs.unibuc.ro

Ierarhia lui Chomsky

- Limbaje Regulate / Gramatici Regulate / Automate finite

Ierarhia lui Chomsky

- Limbaje Regulate / Gramatici Regulate / Automate finite
- Limbaje Independente de Context (Context-Free) / Gramatici Independente de Context / Automate push-down

Ierarhia lui Chomsky

- Limbaje Regulate / Gramatici Regulate / Automate finite
- Limbaje Independente de Context (Context-Free) / Gramatici Independente de Context / Automate push-down
- Limbaje Dependente de Context (Context-Sensitive) / Gramatici Dependente de Context / Automate liniar mărginite

Ierarhia lui Chomsky

- Limbaje Regulate / Gramatici Regulate / Automate finite
- Limbaje Independente de Context (Context-Free) / Gramatici Independente de Context / Automate push-down
- Limbaje Dependente de Context (Context-Sensitive) / Gramatici Dependente de Context / Automate liniar mărginite
- Limbaje recursiv enumerabile / Gramatici oarecare / Maşini Turing

Gramatici

- $G = (N, T, S, P)$
- T mulţime finită de simboluri, numite **terminale**
- N mulţime finită de simboluri, numite **neterminale**
- $S \in N$, numit **simbol de start**
- P multime finită de perechi de tipul $x \rightarrow y$, numite **producţii**, unde:
  – x este un şir de simboluri neterminale si terminale, care conţine cel puţin un neterminal,
  – y este un şir oarecare de simboluri terminale şi neterminale.

Limbajul generat de o gramatică

- $w \Rightarrow w'$ (w derivează într-un pas în w') daca şi numai dacă $w = \alpha x \beta$, $w' = \alpha y \beta$ şi $x \rightarrow y \in P$.
- $w \Rightarrow^* w'$ (w derivează în 0 sau mai mulţi paşi în w') dacă şi numai dacă există şirul de derivări: $w \Rightarrow w_1 \Rightarrow \ldots \Rightarrow w_n = w', n \geq 0$.
- Limbajul generat de gramatică G se defineşte ca fiind: $L(G) = \{w \mid S \Rightarrow^* w,$ şi w e format numai din terminale$\}$

Gramatici regulate

- Producţiile din gramaticile regulate au forma: $A \to aB$ sau $A \to a$, unde $A, B \in N$ şi $a \in T$
- Limbajele generate de gramaticile regulate se numesc limbaje regulate
- **Exemplu:** Gramatica $G = (N, T, S, P)$, unde:
  $N = \{S, B\}$, $T = \{a, b\}$, şi
  $P = \{S \to aS, S \to a, S \to aB, B \to bB, B \to b\}$
  generează limbajul $L(G) = \{a^n b^m \mid n \geq 1, m \geq 0\}$

Automate finite nedeterministe

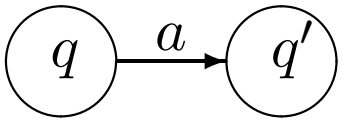
- $A = (Q, V, q_0, F, \delta)$
- Q mulţime finită, numită **mulţimea stărilor**
- $q_0 \in Q$, numită **stare iniţială**
- $F \subseteq Q$, numită **mulţimea stărilor finale**
- V mulţime finită de simboluri, numite **simboluri de intrare**
- $\delta : Q \times V \rightarrow \mathcal{P}(Q)$ **funcţia de tranziţie**
  $\delta(q, a) = \{q_1, \ldots, q_t\}, t \geq 0$ inseamnă că: dacă automatul se află în starea q şi primeşte la intrare simbolul a, poate sa ajungă în oricare din stările $q_1, \ldots, q_t$

Automate finite deterministe

- Dacă pentru orice simbol a şi orice stare q avem: $|\delta(q,a)| = 1$, automatul se numeşte determinist. In acest caz: $\delta : Q \times V \rightarrow Q$
- Extensia funcţiei de tranziţie la cuvinte se face recursiv: $\delta(q, aw) = \cup_{q' \in \delta(q,a)} \delta(q', w)$.
- Limbajul acceptat de automatul nedeterminist A este: $L(A) = \{w \in V^* \mid \delta(q_0, w) \cap F \neq \emptyset\}$
- Limbajul acceptat de automatul determinist A este: $L(A) = \{w \in V^* \mid \delta(q_0, w) \in F\}$

Automate finite: Observaţii şi Exemple

- Dacă $q' \in \delta(q, a)$ reprezentăm:



- $A = (Q, V, q_0, F, \delta)$
- $Q = \{q_0, q_a, q_b, q_e\}$, $V = \{a, b\}$, $F = \{q_a, q_b\}$
- $\delta(q_0, a) = q_a$, $\delta(q_a, a) = q_a$, $\delta(q_a, b) = q_b$, $\delta(q_b, b) = q_b$, şi $\delta(q, x) = q_e$, pentru orice altă pereche $q \in Q$, $x \in V$.
- A este un automat finit determinist si accepta: $L(A) = \{a^n b^m \mid n \geq 1, m \geq 0\}$

Automate finite: Exemple

- $A = (Q, V, q_0, F, \delta)$
- $Q = \{q_0, q_1, q_2, q_3, q_e\}$
- $V = \{a, b\}$
- $F = \{q_1, q_2\}$
- $\delta(q_0, a) = \{q_1, q_2\}$, $\delta(q_1, a) = q_1$, $\delta(q_2, b) = q_2$, şi $\delta(q, x) = q_e$, pentru orice altă pereche $q \in Q$, $x \in V$.
- A este un automat finit nedeterminist si acceptă: $L(A) = \{a^n \mid n \geq 1\} \cup \{ab^n \mid n \geq 1\}$

Automate finite: Exemple

- Fie P un şir cu simboluri din V, $P = p[1] \dots p[n]$. Notăm cu $P_k = p[1] \dots p[k]$.
- $A = (Q, V, q_0, F, \delta)$
- $Q = \{0, 1, \dots, n\}$
- $F = \{n\}$
- $\delta(i, a) = max\{k \mid 0 \leq k \leq n, P_k$ e sufix al lui $P_i a\}$
- A este un automat finit determinist si acceptă: $L(A) = \{w \mid w$ se termină cu $P\}$

Rezultate

- Clasa limbajelor generate de gramaticile regulate coincide cu clasa limbajelor acceptate de automatele finite nedeterministe si coincide cu clasa limbajelor acceptate de automatele finite deterministe. Aceste limbaje se numesc limbaje regulate.
- Există limbaje care nu sunt regulate. $L = \{a^n b^n \mid n \geq 1\}$.
- Limbajele regulate pot fi descrise prin expresii regulate.
- Notaţii:
  pentru un limbaj L, $L^* = \{w \mid w = w_1 \dots w_n, w_i \in L, n \geq 0\}$
  λ este şirul cu 0 simboluri.

Expresii regulate

- $e = \emptyset$, $L(e) = \emptyset$

Expresii regulate

- $e = \emptyset$, $L(e) = \emptyset$
- $e = \lambda$, $L(e) = \{\lambda\}$

Expresii regulate

- $e = \emptyset$, $L(e) = \emptyset$
- $e = \lambda$, $L(e) = \{\lambda\}$
- $e = a$, $L(e) = \{a\}$

Expresii regulate

- $e = \emptyset$, $L(e) = \emptyset$
- $e = \lambda$, $L(e) = \{\lambda\}$
- $e = a$, $L(e) = \{a\}$
- α, β sunt expresii regulate rezultă că $e = \alpha + \beta$ e expresie regulată, şi $L(e) = L(\alpha) \cup L(\beta)$

Expresii regulate

- $e = \emptyset$, $L(e) = \emptyset$
- $e = \lambda$, $L(e) = \{\lambda\}$
- $e = a$, $L(e) = \{a\}$
- α, β sunt expresii regulate rezultă că $e = \alpha + \beta$ e expresie regulată, şi $L(e) = L(\alpha) \cup L(\beta)$
- α, β sunt expresii regulate rezultă că $e = \alpha.\beta$ e expresie regulată, şi $L(e) = L(\alpha)L(\beta)$

Expresii regulate

- $e = \emptyset$, $L(e) = \emptyset$
- $e = \lambda$, $L(e) = \{\lambda\}$
- $e = a$, $L(e) = \{a\}$
- α, β sunt expresii regulate rezultă că $e = \alpha + \beta$ e expresie regulată, şi $L(e) = L(\alpha) \cup L(\beta)$
- α, β sunt expresii regulate rezultă că $e = \alpha.\beta$ e expresie regulată, şi $L(e) = L(\alpha)L(\beta)$
- α este expresie regulată rezultă că $e_1 = (\alpha)$ şi $e_2 = (\alpha)^*$ sunt expresii regulate; $L(e_1) = L(\alpha)$ şi $L(e_2) = L(\alpha)^*$

Expresii Regulate

- Fie $A = (Q, V, q_0, F, \delta)$ automat finit determinist, cu $Q = \{0, 1, \ldots, n\}$. Construim expresia e, unde $L(e) = L(A)$

Expresii Regulate

- Fie $A = (Q, V, q_0, F, \delta)$ automat finit determinist, cu $Q = \{0, 1, \ldots, n\}$. Construim expresia e, unde $L(e) = L(A)$
- $R^0_{i,j} = \{a_1, \ldots, a_t \mid \delta(q_i, a_k) = q_j, 1 \leq k \leq t\}$,
  $r^0_{i,j} = a_1 + \ldots + a_k$;
  $R^t_{i,j} = \{w \mid \delta(q_i, w) = q_j$ iar tranziţia din se face prin stări mai mici cu $t\}$;
  $R^{t+1}_{i,j} = R^t_{i,j} \cup R^t_{i,t}(R^t_{t,t})^* R^t_{t,j}$
  $r^{t+1}_{i,j} = r^t_{i,j} + r^t_{i,t}.(r^t_{t,t})^*.r^t_{t,j}$.

Expresii Regulate

- Fie $A = (Q, V, q_0, F, \delta)$ automat finit determinist, cu $Q = \{0, 1, \ldots, n\}$. Construim expresia e, unde $L(e) = L(A)$
- $R^0_{i,j} = \{a_1, \ldots, a_t \mid \delta(q_i, a_k) = q_j, 1 \leq k \leq t\}$,
  $r^0_{i,j} = a_1 + \ldots + a_k$;
  $R^t_{i,j} = \{w \mid \delta(q_i, w) = q_j$ iar tranziţia din se face prin stări mai mici cu $t\}$;
  $R^{t+1}_{i,j} = R^t_{i,j} \cup R^t_{i,t}(R^t_{t,t})^* R^t_{t,j}$
  $r^{t+1}_{i,j} = r^t_{i,j} + r^t_{i,t}.(r^t_{t,t})^*.r^t_{t,j}$.
- $L(A) = \cup_{j \in F} R^{n+1}_{0,j}$; $e = r^{n+1}_{0,j_1} + \ldots + r^{n+1}_{0,j_p}, F = \{j_1, \ldots, j_p\}$.

Expresii Regulate: Exemple

- $\{a^n b^m \mid n \geq 1, m \geq 0\} = L(aa^*b+)$.
- $\{a^n \mid n \geq 1\} \cup \{ab^n \mid n \geq 1\} = L(aa^* + abb^*)$
- $\{w \mid$
  w se termină cu P şi are simboluri din $\{a_1, a_2, \ldots, a_n\}\} =$
  $L((a_1 + a_2 + \ldots + a_n)^* P)$

Automate finite: Proprietăţi

- Dacă L este regulat, atunci $L^C = \{w \mid w \notin L\}$ e regulat. (într-un automat care acceptă L starile nefinale devin finale, iar cele finale devine nefinale)
- Dacă L_1 şi L_2 sunt regulate, atunci $L_1 \cap L_2$ este regulat. (se construieşte un automat care are ca stări perechi de stări din automatele care acceptă cele două limbaje, şi se simulează un calcul simultan in acestea)
- Dacă L_1 şi L_2 sunt regulate, atunci $L_1 \cup L_2$ este regulat.

Automate finite: Proprietăţi

- Stări accesibile: stările la care se poate ajunge din starea iniţială (se determină prin parcurgere a grafului asociat automatului).
  Automatul nu acceptă nici un cuvânt ddacă nu există stări finale accesibile.
- Stări coaccesibile: stările de la care se poate ajunge în stări finale (se determină prin parcurgere a grafului asociat automatului)
- Pentru un limbaj regulat L există un automat cu număr minim de stări (automat minimal). Oricare două automate care acceptă limbajul regulat L, cu număr minim de stări, sunt isomorfe.

Minimzarea unui automat finit

- Se dă automatul $A = (Q, V, q_0, F, \delta)$. Să se găsească un automat cu număr minim de stări care acceptă $L(A)$.
- Idee: se elimină stările neaccesibile; se construieşte echivalenţa comportamentală între stările din Q: $q \equiv s \Leftrightarrow [\forall w, \delta(q, w) \in F \Leftrightarrow \delta(s, w) \in F]$
- Automatul minimal va avea ca stări clasele de echivalenţă obţinute faţă de relaţia de mai sus, tranziţiile fiind aceleaşi.
- Complexitatea algoritmului “simplu" de minimzare: $O(|V||Q|^2)$.
- Algoritmul lui Hopcroft: $O(|V||Q| \log |Q|)$

Algoritmul de calcul pentru clasele de echivalenţă

1. for $p \in F$ and $q \in Q \setminus F$ mark the pair (p, q);
2. for $p, q \in Q$ construct an empty list;
3. for each unmarked pair (p, q) do
4. if $\exists a \in V$ such that $(\delta(p, a), \delta(q, a))$ is marked
5. mark (p, q)
6. recursively mark all the pairs on the list of (p, q), and on the lists of the pairs marked at this step
7. else
8. for all $a \in V$
9. put (p, q) on the list of $(\delta(p, a), \delta(q, a))$

Aplicaţie:

- Să se verifice dacă două automate A_1, A_2 acceptă acelaşi limbaj?

1. Se minimizează ambele automate, şi se verifică dacă se obţin automate identice
2. Se consideră ambele automate ca un nou pseudo-automat, şi se aplică algoritmul de minimizare asupra acestuia. Dacă stările iniţiale ale celor două automate vor fi echivalente, atunci cele două automate acceptau acelaşi
3. Se calculează automatul pentru $L(A_1) \cap L(A_2)^C$ şi automatul pentru $L(A_2) \cap L(A_1)^C$. Dacă (şi numai dacă) nici unul nu acceptă vreun cuvânt atunci cele două automate acceptă aceleaşi limbaje.