

Limbaje Formale, Automate și Compilatoare

Curs 6

2020-21

Curs 6

- 1 Eliminarea regulilor de ștergere și redenumirilor din gramatici de tip 2
- 2 Forma normală Chomsky
- 3 Problema recunoașterii: algoritmul Cocke Younger Kasami
- 4 Automate pushdown
- 5 Legătura dintre automatele pushdown și limbajele de tip 2
- 6 Automate pushdown deterministe

Curs 6

- 1 Eliminarea regulilor de ștergere și redenumirilor din gramatici de tip 2
- 2 Forma normală Chomsky
- 3 Problema recunoașterii: algoritmul Cocke Younger Kasami
- 4 Automate pushdown
- 5 Legătura dintre automatele pushdown și limbajele de tip 2
- 6 Automate pushdown deterministe

Eliminarea regulilor de ștergere

- Intrare: $G = (N, T, S, P)$
- Ieșire: $G' = (N, T, S, P')$, $L(G') = L(G)$, P' nu conține reguli de ștergere (reguli de forma $A \rightarrow \epsilon$)

$N_0 = \{A | A \in N, A \rightarrow \epsilon \in P\}; i = 0;$

do {

$i = i + 1;$

$N_i = N_{i-1} \cup \{X | X \in N, \exists X \rightarrow \alpha \in P, \alpha \in N_{i-1}^*\};$

} while $N_i \neq N_{i-1};$

$N_\epsilon = N_i;$

Eliminarea regulilor de ștergere

- Intrare: $G = (N, T, S, P)$
- Ieșire: $G' = (N, T, S, P')$, $L(G') = L(G)$, P' nu conține reguli de ștergere (reguli de forma $A \rightarrow \epsilon$)

$N_0 = \{A | A \in N, A \rightarrow \epsilon \in P\}; i = 0;$

do {

$i = i + 1;$

$N_i = N_{i-1} \cup \{X | X \in N, \exists X \rightarrow \alpha \in P, \alpha \in N_{i-1}^*\};$

} while $N_i \neq N_{i-1};$

$N_\epsilon = N_i;$

Are loc:

- $N_0 \subseteq N_1 \subseteq \dots \subseteq N_i \subseteq N_{i+1} \subseteq \dots \subseteq N_\epsilon \subseteq N$
- $A \in N_\epsilon \iff A \Rightarrow^+ \epsilon$

Eliminarea regulilor de ștergere

P' se obține din P astfel:

- în fiecare regulă $A \rightarrow \alpha \in P$ se pun în evidență simbolurile din N_ϵ ce apar în α :

$$\alpha = \alpha_1 X_1 \alpha_2 X_2 \dots \alpha_n X_n \alpha_{n+1}, \quad X_i \in N_\epsilon$$

- se înlocuiește fiecare regulă de acest fel cu mulțimea de reguli de forma

$$A \rightarrow \alpha_1 Y_1 \alpha_2 Y_2 \dots \alpha_n Y_n \alpha_{n+1} \text{ unde } Y_i = X_i \text{ sau } Y_i = \epsilon$$

în toate modurile posibile (2^n)

- se elimină toate regulile de ștergere
- pentru a obține cuvântul nul (dacă S este în N_ϵ) se adaugă S' simbol de start nou și regulile $S' \rightarrow S, S' \rightarrow \epsilon$

Exemplu

$G = (\{S, A, B, C\}, \{a, b, c\}, S, P)$, unde P :

- $S \rightarrow aAbC \mid BC$
- $A \rightarrow aA \mid aB$
- $B \rightarrow bB \mid C$
- $C \rightarrow cC \mid \epsilon$

$G' = (\{S', S, A, B, C\}, \{a, b, c\}, S', P')$ unde P' :

- $S' \rightarrow S \mid \epsilon$
- $S \rightarrow aAbC \mid aAb \mid B \mid C$
- $A \rightarrow aA \mid aB \mid a$
- $B \rightarrow bB \mid b \mid C$
- $C \rightarrow cC \mid c$

Eliminarea redenumirilor ($A \rightarrow B, A, B \in N$)

- Intrare: $G = (N, T, S, P)$
- Ieșire: $G' = (N, T, S, P'), L(G') = L(G), P'$ nu conține redenumiri

```

for( $A \in N$ ) {
   $N_0 = \{A\}; i = 0;$ 
  do {
     $i = i + 1;$ 
     $N_i = N_{i-1} \cup \{C \mid C \in N, \exists B \rightarrow C \in P, B \in N_{i-1}\};$ 
  } while  $N_i \neq N_{i-1};$ 
   $N_A = N_i; // N_A = \{X \in N \mid A \Rightarrow^* X\}$ 
}
 $P' = \{X \rightarrow \alpha \in P \mid \alpha \notin N\}$ 
for( $X \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n \in P'$ )
  for( $A \in N \ \&\& \ X \in N_A, X \neq A$ )
     $P' = P' \cup \{A \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n\}$ 

```


Exemplu

$G = (\{x, y, z\}, \{a, b, c\}, x, P)$, unde P :

- $x \rightarrow y|ax|a$
- $y \rightarrow z|by|b$
- $z \rightarrow cz|c$

$N_x = \{x, y, z\}$, $N_y = \{y, z\}$, $N_z = \{z\}$

Gramatica echivalentă fără redenumiri $G' = (\{x, y, z\}, \{a, b, c\}, x, P')$
unde P' :

- $x \rightarrow ax|a|by|b|cz|c$
- $y \rightarrow by|b|cz|c$
- $z \rightarrow cz|c$

Curs 6

- 1 Eliminarea regulilor de ștergere și redenumirilor din gramatici de tip 2
- 2 Forma normală Chomsky**
- 3 Problema recunoașterii: algoritmul Cocke Younger Kasami
- 4 Automate pushdown
- 5 Legătura dintre automatele pushdown și limbajele de tip 2
- 6 Automate pushdown deterministe

Forma normală Chomsky

Definiție 1

O gramatică este în *formă normală Chomsky* dacă regulile sale au forma:

$A \rightarrow BC, A \rightarrow a$ (și eventual $S \rightarrow \epsilon$) ($A, B, C \in N$ și $a \in T$).

Teorema 1

Orice limbaj independent de context poate fi generat de o gramatică în formă normală Chomsky.

Demonstrație

- Se elimină regulile de ștergere și redenumirile

Demonstrație

- Se elimină regulile de ștergere și redenumirile
- Se elimină regulile care nu sunt în formă normală Chomsky:
Dacă $A \rightarrow x_1 x_2 \dots x_n$, $n > 1$ este o astfel de regulă atunci o înlocuim cu $A \rightarrow Y_1 Y_2 \dots Y_n$ unde:
 - $Y_i = x_i$, dacă $x_i \in N$ (neterminalii rămân la fel)
 - $Y_i = x_a$ dacă $x_i = a \in T$ (x_a este neterminal nou) și se adaugă regula $x_a \rightarrow a$

Demonstrație

- Se elimină regulile de ștergere și redenumirile
- Se elimină regulile care nu sunt în formă normală Chomsky:
Dacă $A \rightarrow x_1 x_2 \dots x_n$, $n > 1$ este o astfel de regulă atunci o înlocuim cu $A \rightarrow Y_1 Y_2 \dots Y_n$ unde:
 - $Y_i = x_i$, dacă $x_i \in N$ (neterminalii rămân la fel)
 - $Y_i = x_a$ dacă $x_i = a \in T$ (x_a este neterminal nou) și se adaugă regula $x_a \rightarrow a$
- O regulă de forma $A \rightarrow Y_1 Y_2 \dots Y_n$, dacă $n > 2$, o înlocuim cu:
 - $A \rightarrow Y_1 Z_1$
 - $Z_1 \rightarrow Y_2 Z_2$
 -
 - $Z_{n-3} \rightarrow Y_{n-2} Z_{n-2}$
 - $Z_{n-2} \rightarrow Y_{n-1} Y_n$,
 unde Z_1, Z_2, \dots, Z_{n-2} sunt neterminali noi.

Exemplu

$G = (\{S, A\}, \{a, b, c\}, S, P)$, unde P :

- $S \rightarrow aSb \mid cAc$
- $A \rightarrow cA \mid c$

Gramatica echivalentă în formă normală Chomsky

$G = (\{S, A, x_a, x_b, Z_1, Z_2\}, \{a, b, c\}, S, P')$, unde P' :

- $S \rightarrow x_a Z_1 \mid x_c Z_2$
- $Z_1 \rightarrow S x_b$
- $Z_2 \rightarrow A x_c$
- $A \rightarrow x_c A \mid c$
- $x_a \rightarrow a$
- $x_b \rightarrow b$
- $x_c \rightarrow c$

Curs 6

- 1 Eliminarea regulilor de ștergere și redenumirilor din gramatici de tip 2
- 2 Forma normală Chomsky
- 3 Problema recunoașterii: algoritmul Cocke Younger Kasami**
- 4 Automate pushdown
- 5 Legătura dintre automatele pushdown și limbajele de tip 2
- 6 Automate pushdown deterministe

Algoritmul Cocke Younger Kasami (CYK)

- Problema recunoașterii în gramatici de tip 2: dată o gramatică de tip 2 și un cuvânt w , să se decidă dacă $w \in L(G)$
- Problema recunoașterii în gramatici în formă normală Chomsky se poate rezolva cu algoritmul CYK în timp $O(n^3)$.
- Dacă $w = a_1 a_2 \dots a_n$ atunci se construiesc mulțimile

$$V_{ij} = \{A \mid A \Rightarrow^+ a_i a_{i+1} \dots a_{i+j-1}\}$$

inductiv pentru $j = 1, \dots, n$

$$w \in L(G) \Leftrightarrow S \in V_{1n}$$

Algoritmul Cocke Younger Kasami

- Pentru $j = 1$:

- $V_{i1} = \{A | A \Rightarrow^+ a_i\} = \{A | \exists A \rightarrow a_i \in P\}$

- Pentru $j > 1$, V_{ij} :

- Dacă $A \Rightarrow^+ a_i a_{i+1} \dots a_{i+j-1}$:

$$A \Rightarrow BC \Rightarrow^+ a_i a_{i+1} \dots a_{i+j-1} \text{ și}$$

$$B \Rightarrow^+ a_i a_{i+1} \dots a_{i+k-1} \quad (B \in V_{ik})$$

$$C \Rightarrow^+ a_{i+k} a_{i+k+1} \dots a_{i+j-1} \quad (C \in V_{i+k, j-k})$$

$$\text{unde } 1 \leq i \leq n+1-j, 1 \leq k \leq j-1$$

- $V_{ij} = \bigcup_{k=1}^{j-1} \{A | A \rightarrow BC \in P, B \in V_{ik}, C \in V_{i+k, j-k}\}$

Algoritmul Cocke Younger Kasami

- Notăție:

$$\{A | A \rightarrow BC \in P, B \in V_{ik}, C \in V_{i+k \ j-k}\} = V_{ik} \circ V_{i+k \ j-k}$$

- Atunci:

pentru $2 \leq j \leq n, 1 \leq i \leq n+1-j$:

$$V_{ij} = \bigcup_{k=1}^{j-1} (V_{ik} \circ V_{i+k \ j-k})$$

Algoritmul Cocke Younger Kasami

- Intrare: $G = (N, T, S, P)$ în formă normală Chomsky, $w = a_1 a_2 \dots a_n$
- Ieșire: $w \in L(G)$?

```

for(i=1; i<=n; i++)
     $V_{i1} = \{A | \exists A \rightarrow a_i \in P\};$ 
for(j=2; j<=n; j++)
    for (i=1; i<=n+1-j; i++){
         $V_{ij} = \emptyset;$ 
        for(k=1; k<=j-1; k++)
             $V_{ij} = V_{ij} \cup (V_{ik} \circ V_{i+k \ j-k});$ 
    }
if( $S \in V_{1n}$ )  $w \in L(G)$  else  $w \notin L(G)$ 

```

Exemplu

$G = (\{S, X, Y, Z\}, \{a, b, c\}, S, P)$, unde P :

- $S \rightarrow XY$
- $X \rightarrow XY|a$
- $Y \rightarrow YZ|a|b$
- $Z \rightarrow c$

$w = abc$

Exemplu

$G = (\{S, X, Y, Z\}, \{a, b, c\}, S, P)$, unde P :

- $S \rightarrow XY$
- $X \rightarrow XY|a$
- $Y \rightarrow YZ|a|b$
- $Z \rightarrow c$

$w = abc$

$V_{11} = \{X, Y\}$	$V_{12} = \{S, X\}$	$V_{13} = \{S, X\}$
$V_{21} = \{Y\}$	$V_{22} = \{Y\}$	
$V_{31} = \{Z\}$		

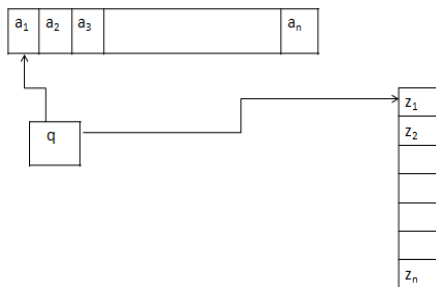
$S \in V_{13} \Leftrightarrow abc \in L(G)$

Curs 6

- 1 Eliminarea regulilor de ștergere și redenumirilor din gramatici de tip 2
- 2 Forma normală Chomsky
- 3 Problema recunoașterii: algoritmul Cocke Younger Kasami
- 4 Automate pushdown**
- 5 Legătura dintre automatele pushdown și limbajele de tip 2
- 6 Automate pushdown deterministe

Automate pushdown

- Automat finit + memorie pushdown (stiva)
- Model fizic:



Automate pushdown-definiție

Definiție 2

Un automat pushdown este un 7-uplu: $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$:

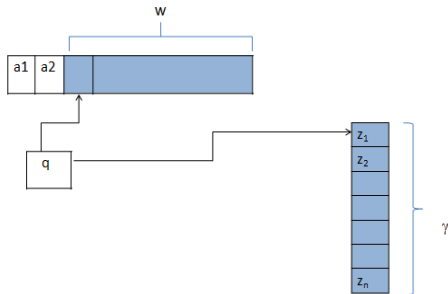
- *Q este mulțimea (finită) a stărilor*
- *Σ este alfabetul de intrare*
- *Γ este alfabetul memoriei pushdown (stivei)*
- *$q_0 \in Q$ este starea inițială*
- *$z_0 \in \Gamma$ este simbolul inițial din stivă*
- *$F \subseteq Q$ este mulțimea stărilor finale*
- *$\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$*

Modelul este nedeterminist

Configurația unui automat pushdown

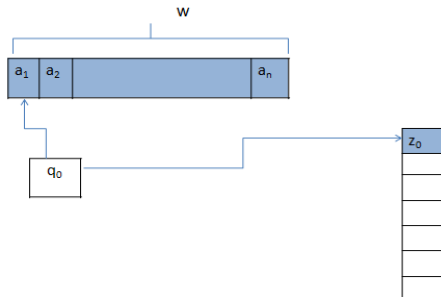
Configurație: $(q, w, \gamma) \in Q \times \Sigma^* \times \Gamma^*$

1 : γ (primul simbol din γ) reprezintă vârful stivei



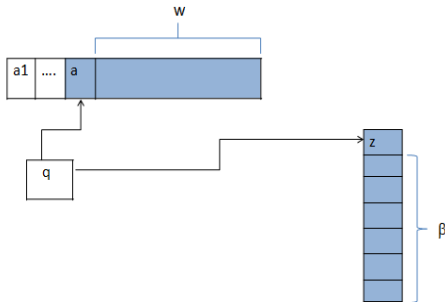
Automate pushdown

Configurație inițială: $(q_0, w, z_0) \in Q \times \Sigma^* \times \Gamma^*$



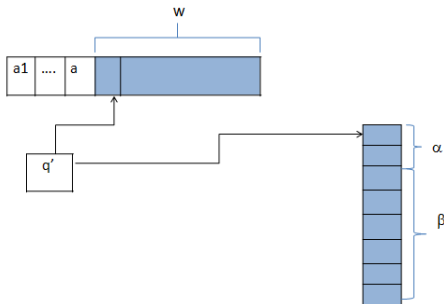
Relația de tranziție între configurații

- Configurația curentă $(q, aw, z\beta)$ și $(q', \alpha) \in \delta(q, a, z)$
 $(q, q' \in Q, a \in \Sigma \cup \{\epsilon\}, z \in \Gamma, \alpha, \beta \in \Gamma^*)$



Relația de tranziție între configurații

- $(q, aw, z\beta) \vdash (q', w, \alpha\beta)$



Relația de tranziție între configurații

Fie $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ un automat pushdown.

- Relația de tranziție între configurații:

$(q, aw, z\beta) \vdash (q', w, \alpha\beta)$ dacă $(q', \alpha) \in \delta(q, a, z)$

$(q, q' \in Q, a \in \Sigma \cup \{\epsilon\}, z \in \Gamma, \alpha, \beta \in \Gamma^*)$

- Calcul: închiderea reflexivă și tranzitivă a relației de mai sus: dacă C_1, \dots, C_n configurații astfel încât:

$$C_1 \vdash C_2 \vdash \dots \vdash C_n$$

se scrie: $C_1 \vdash^+ C_n$ dacă $n \geq 2$, $C_1 \vdash^* C_n$, dacă $n \geq 1$

Limbaajul recunoscut

Prin stări finale (dacă $F \neq \emptyset$)

$$L(M) = \{w \in \Sigma^* \mid (q_0, w, z_0) \vdash^* (q, \epsilon, \gamma), q \in F, \gamma \in \Gamma^*\}$$

Prin golirea stivei (dacă $F = \emptyset$)

$$L_\epsilon(M) = \{w \in \Sigma^* \mid (q_0, w, z_0) \vdash^* (q, \epsilon, \epsilon), q \in Q\}$$

Exemplu

Automat care recunoaște limbajul $\{a^n b^n | n \geq 1\}$:

$$M = (\{q_0, q_1, q_2\}, \{a, b\}, \{a, z\}, \delta, q_0, z, \{q_2\})$$

$$1 \quad \delta(q_0, a, z) = \{(q_0, az)\}$$

$$2 \quad \delta(q_0, a, a) = \{(q_0, aa)\}$$

$$3 \quad \delta(q_0, b, a) = \{(q_1, \epsilon)\}$$

$$4 \quad \delta(q_1, b, a) = \{(q_1, \epsilon)\}$$

$$5 \quad \delta(q_1, \epsilon, z) = \{(q_2, \epsilon)\}$$

Exemple

- Un automat pushdown ce recunoaște limbajul $\{waw^R \mid w \in \{0, 1\}^*\}$

Exemple

- Un automat pushdown ce recunoaște limbajul $\{waw^R \mid w \in \{0, 1\}^*\}$
 - Fiecare 0 sau 1 citit se introduce în stivă
 - a la intrare produce pregătirea scoaterii a câte un simbol din stiva dacă el coincide cu cel din intrare

Exemple

- Un automat pushdown ce recunoaște limbajul $\{waw^R \mid w \in \{0, 1\}^*\}$
 - Fiecare 0 sau 1 citit se introduce în stivă
 - a la intrare produce pregătirea scoaterii a câte un simbol din stiva dacă el coincide cu cel din intrare

$$M = (\{q_0, q_1, q_2\}, \{0, 1, a\}, \{0, 1, z\}, \delta, q_0, z, \{q_2\})$$

- 1 $\delta(q_0, i, z) = \{(q_0, iz)\}, (i \in \{0, 1\})$
- 2 $\delta(q_0, i, j) = \{(q_0, ij)\}, (i, j \in \{0, 1\})$
- 3 $\delta(q_0, a, i) = \{(q_1, i)\}$
- 4 $\delta(q_1, i, i) = \{(q_1, \epsilon)\}$
- 5 $\delta(q_1, \epsilon, z) = \{(q_2, \epsilon)\}$

Exemple

- Un automat pushdown ce recunoaște limbajul $\{waw^R \mid w \in \{0, 1\}^*\}$
 - Fiecare 0 sau 1 citit se introduce în stivă
 - a la intrare produce pregătirea scoaterii a câte un simbol din stiva dacă el coincide cu cel din intrare

$$M = (\{q_0, q_1, q_2\}, \{0, 1, a\}, \{0, 1, z\}, \delta, q_0, z, \{q_2\})$$

- 1 $\delta(q_0, i, z) = \{(q_0, iz)\}, (i \in \{0, 1\})$
- 2 $\delta(q_0, i, j) = \{(q_0, ij)\}, (i, j \in \{0, 1\})$
- 3 $\delta(q_0, a, i) = \{(q_1, i)\}$
- 4 $\delta(q_1, i, i) = \{(q_1, \epsilon)\}$
- 5 $\delta(q_1, \epsilon, z) = \{(q_2, \epsilon)\}$

- Un automat pushdown ce recunoaște limbajul $\{ww^R \mid w \in \{0, 1\}^*\}$?

Exemple

- Un automat pushdown ce recunoaște limbajul $\{waw^R \mid w \in \{0, 1\}^*\}$
 - Fiecare 0 sau 1 citit se introduce în stivă
 - a la intrare produce pregătirea scoaterii a câte un simbol din stiva dacă el coincide cu cel din intrare

$$M = (\{q_0, q_1, q_2\}, \{0, 1, a\}, \{0, 1, z\}, \delta, q_0, z, \{q_2\})$$

- 1 $\delta(q_0, i, z) = \{(q_0, iz)\}, (i \in \{0, 1\})$
 - 2 $\delta(q_0, i, j) = \{(q_0, ij)\}, (i, j \in \{0, 1\})$
 - 3 $\delta(q_0, a, i) = \{(q_1, i)\}$
 - 4 $\delta(q_1, i, i) = \{(q_1, \epsilon)\}$
 - 5 $\delta(q_1, \epsilon, z) = \{(q_2, \epsilon)\}$
- Un automat pushdown ce recunoaște limbajul $\{ww^R \mid w \in \{0, 1\}^*\}$?
 - Un automat pushdown ce recunoaște limbajul $\{ww \mid w \in \{0, 1\}^*\}$?

Echivalența definițiilor privind recunoașterea

Teorema 2

Pentru orice automat pushdown M cu $F = \emptyset$, există un automat pushdown M' cu stări finale astfel ca $L(M') = L_{\epsilon}(M)$.

Teorema 3

Pentru orice automat pushdown M cu $F \neq \emptyset$, există un automat pushdown M' cu $F = \emptyset$ astfel ca $L_{\epsilon}(M') = L(M)$.

Curs 6

- 1 Eliminarea regulilor de ștergere și redenumirilor din gramatici de tip 2
- 2 Forma normală Chomsky
- 3 Problema recunoașterii: algoritmul Cocke Younger Kasami
- 4 Automate pushdown
- 5 Legătura dintre automatele pushdown și limbajele de tip 2**
- 6 Automate pushdown deterministe

Automatul pushdown echivalent cu o gramatică de tip 2

Teorema 4

Pentru orice gramatică G există un automat pushdown M fără stări finale astfel încât $L_{\epsilon}(M) = L(G)$

Automatul pushdown echivalent cu o gramatică de tip 2

Teorema 4

Pentru orice gramatică G există un automat pushdown M fără stări finale astfel încât $L_{\epsilon}(M) = L(G)$

- Fie $G = (N, T, S, P)$
- Construim $M = (\{q\}, T, N \cup T, \delta, q, S, \emptyset)$ unde:
 - ① $\delta(q, \epsilon, A) = \{(q, \beta) \mid A \rightarrow \beta \in P\}, \forall A \in N$
 - ② $\delta(q, a, a) = \{(q, \epsilon)\}, \forall a \in T$
 - ③ $\delta(q, x, y) = \emptyset$, în restul cazurilor
- $w \in L(G) \Leftrightarrow S \Rightarrow^+ w \Leftrightarrow (q, w, S) \vdash^+ (q, \epsilon, \epsilon) \Leftrightarrow w \in L_{\epsilon}(M)$
- M simulează derivările extrem stângi din G

Exemplu

- $G = (\{x\}, \{a, b\}, x, \{x \rightarrow axb, x \rightarrow ab\})$
- Automatul pushdown echivalent:

$$M = (\{q\}, \{a, b\}, \{a, b, x\}, \delta, q, x, \emptyset)$$

- 1 $\delta(q, \epsilon, x) = \{(q, axb), (q, ab)\}$
- 2 $\delta(q, a, a) = \{(q, \epsilon)\}$
- 3 $\delta(q, b, b) = \{(q, \epsilon)\}$

Gramatica echivalentă cu un automat pushdown

Teorema 5

Pentru orice automat pushdown M există o gramatică G astfel încât
$$L(G) = L_{\epsilon}(M)$$

Gramatica echivalentă cu un automat pushdown

Teorema 5

Pentru orice automat pushdown M există o gramatică G astfel încât $L(G) = L_{\epsilon}(M)$

- Fie $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, \emptyset)$
- Construim $G = (N, \Sigma, S, P)$ astfel:
 - $N = \{[qzp] \mid p, q \in Q, z \in \Gamma\} \cup \{S\}$
 - P conține toate regulile de forma:
 - $S \rightarrow [q_0 z_0 q], \forall q \in Q$
 - dacă $(p, \epsilon) \in \delta(q, a, z)$, atunci:

$$[qzp] \rightarrow a$$
 - dacă $(p, z_1 z_2 \dots z_m) \in \delta(q, a, z)$, atunci, pentru orice secvență de stări $q_1, \dots, q_m \in Q$:

$$[qzq_m] \rightarrow a[pz_1 q_1][q_1 z_2 q_2] \dots [q_{m-1} z_m q_m]$$
- Are loc: $[qzp] \Rightarrow^+ w \Leftrightarrow (q, w, z) \vdash^+ (p, \epsilon, \epsilon)$

Curs 6

- 1 Eliminarea regulilor de ștergere și redenumirilor din gramatici de tip 2
- 2 Forma normală Chomsky
- 3 Problema recunoașterii: algoritmul Cocke Younger Kasami
- 4 Automate pushdown
- 5 Legătura dintre automatele pushdown și limbajele de tip 2
- 6 Automate pushdown deterministe**

Automate pushdown deterministe

Definiție 3

Automatul pushdown $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ este determinist dacă funcția de tranziție $\delta : Q \times (\Gamma \cup \{\epsilon\}) \times \Gamma \longrightarrow 2^{Q \times \Gamma^*}$ îndeplinește condițiile:

- 1 $|\delta(q, a, z)| = 1, \forall a \in \Sigma \cup \{\epsilon\}, \forall q \in Q, \forall z \in \Gamma$
- 2 Dacă $\delta(q, \epsilon, z) \neq \emptyset$ atunci $\delta(q, a, z) = \emptyset, \forall a \in \Sigma$

Un automat pushdown determinist poate avea ϵ -tranziii

Automate pushdown deterministe

Definiție 3

Automatul pushdown $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ este determinist dacă funcția de tranziție $\delta : Q \times (\Gamma \cup \{\epsilon\}) \times \Gamma \longrightarrow 2^{Q \times \Gamma^*}$ îndeplinește condițiile:

- 1 $|\delta(q, a, z)| = 1, \forall a \in \Sigma \cup \{\epsilon\}, \forall q \in Q, \forall z \in \Gamma$
- 2 Dacă $\delta(q, \epsilon, z) \neq \emptyset$ atunci $\delta(q, a, z) = \emptyset, \forall a \in \Sigma$

Un automat pushdown determinist poate avea ϵ -tranziii

$M = (\{q_0, q_1, q_2\}, \{0, 1, a\}, \{0, 1, z\}, \delta, q_0, z, \{q_2\})$

- 1 $\delta(q_0, i, z) = \{(q_0, iz)\}, (i \in \{0, 1\})$
- 2 $\delta(q_0, i, j) = \{(q_0, ij)\}, (i, j \in \{0, 1\})$
- 3 $\delta(q_0, a, i) = \{(q_1, i)\}$
- 4 $\delta(q_1, i, i) = \{(q_1, \epsilon)\}$
- 5 $\delta(q_1, \epsilon, z) = \{(q_2, \epsilon)\}$

$L(M) = \{waw^R \mid w \in \{0, 1\}^*\}$

\mathcal{L}_{2DET} - Limbaje de tip 2 deterministe

$$\mathcal{L}_{2DET} = \{L \mid \exists M \text{ automat pushdown determinist astfel ca } L = L(M)\}.$$

- Clasa \mathcal{L}_{2DET} este o clasă proprie a clasei de limbaje \mathcal{L}_2 ($\mathcal{L}_{2DET} \subset \mathcal{L}_2$).
- $\{ww^R \mid w \in \{0, 1\}^*\} \in \mathcal{L}_2 \setminus \mathcal{L}_{2DET}$

\mathcal{L}_{2DET} - Limbaje de tip 2 deterministe

$\mathcal{L}_{2DET} = \{L \mid \exists M \text{ automat pushdown determinist astfel ca } L = L(M)\}.$

- Clasa \mathcal{L}_{2DET} este o clasă proprie a clasei de limbaje \mathcal{L}_2 ($\mathcal{L}_{2DET} \subset \mathcal{L}_2$).
- $\{ww^R \mid w \in \{0,1\}^*\} \in \mathcal{L}_2 \setminus \mathcal{L}_{2DET}$
 $M = (\{q_0, q_1, q_2\}, \{0,1\}, \{0,1,z\}, \delta, q_0, z, \{q_2\})$
 - 1 $\delta(q_0, i, z) = \{(q_0, iz)\}, (i \in \{0,1\})$
 - 2 $\delta(q_0, i, j) = \{(q_0, ij)\}, (i, j \in \{0,1\}, i \neq j)$
 - 3 $\delta(q_0, i, i) = \{(q_0, ii), (q_1, \epsilon)\}$
 - 4 $\delta(q_1, i, i) = \{(q_1, \epsilon)\}$
 - 5 $\delta(q_1, \epsilon, z) = \{(q_2, \epsilon)\}$

\mathcal{L}_{2DET} - Limbaje de tip 2 deterministe

Definiție 4

O gramatică G este deterministă dacă:

- Orice regulă este de forma $A \rightarrow a\alpha$, unde $a \in T$ iar $\alpha \in (N \cup T)^*$
- Pentru orice $A \in N$, dacă $A \rightarrow a\alpha$, $A \rightarrow b\alpha'$ sunt reguli, atunci $a \neq b$

Pentru orice gramatică deterministă G există un automat pushdown determinist M astfel ca $L(G) = L(M)$