

Analiza descendenta - prima parte

April 14, 2024

Automat finit - reamintire

Automat finit (finite automaton, finite state acceptor):

$$A = (T, Q, R, q_0, F)$$

- ▶ Q set nevid - setul starilor interne
- ▶ $(T \cup Q, R)$ sistem de rescriere; $T \cap Q = \emptyset$
- ▶ $q_0 \in Q$ - starea initiala
- ▶ $F \subseteq Q$ - stari finale
- ▶ fiecare element din R are forma $qt \rightarrow q'$, $q, q' \in Q, t \in T$

$$L(A) = \{\tau \in T^* \mid q_0\tau \Rightarrow^* q, q \in F\}$$

Automat stiva - definitie sistem de rescriere -reamintire

Automat stiva

$$A = (T, Q, R, q_0, F, S, s_0)$$

, unde:

- ▶ Q set nevid - setul starilor interne
- ▶ $(T \cup Q \cup S, R)$ sistem de rescriere; $T \cap Q = \emptyset$
- ▶ $q_0 \in Q$ - starea initiala
- ▶ $s_0 \in S \cup \{\varepsilon\}$ - simboluri stiva, s_0 continutul initial al stivei
- ▶ $F \subseteq Q$ - stari finale
- ▶ fiecare element din R are forma $\sigma q t \tau \rightarrow \sigma' q' \tau$,
 $\sigma, \sigma' \in S^*$, $q, q' \in Q$, $t \in T \cup \varepsilon$, $\tau \in T^*$

Daca automatul e la configuratia $s_1 \dots s_n q \tau$ intr-o derivare, automatul e in starea q , τ este partea necitita din input, s_1, \dots, s_n este continutul pe stiva, s_n in varf.

Limbaj acceptat

Daca automatul e la configuratia $s_1 \dots s_n q \tau$ intr-o derivare, automatul e in starea q , τ este partea necitita din input, s_1, \dots, s_n este continutul pe stiva, s_n in varf.

$$L(A) = \{\tau \mid s_0 q_0 \tau \Rightarrow^* q, q \in F, \tau \in T^*\}$$

CFG - PDA

Pentru fiecare gramatica independenta de context G exista un automat stiva A a.i. $L(A)=L(G)$.

Parsing

- ▶ Rolul parsarii: reconstruirea derivarii prin care o CFG poate genera un input string dat.
- ▶ Echivalent cu construirea arborelui de parsare care reprezintă derivarea
- ▶ Directii:
 - ▶ Top-Down - constructia incepe de la radacina; derivarea stanga
 - ▶ Bottom-up - constructia incepe de la frunze; mai greu de construit manual, dar pot fi generate

Rezumat

Recursive descent parsing

Predictive parsing

Structuri ajutatoare: FIRST, FOLLOW

Definire Gramatici LL(k)

Algoritmul LL(K)

Exemplu aplicare LL(k)

Exemplu aplicare LL(3)

1. Recursive descent - top down parse

Arborele de parsare e construit:

- ▶ de la simbolul de start
- ▶ de la stanga la dreapta
- ▶ se incearca regulile in ordinea in care apar
- ▶ revenire si incercare alternative

$$E \rightarrow T \mid T + E$$

$$T \rightarrow \text{int} \mid \text{int} * T \mid (E)$$

Parse tree pt: (int)
 \uparrow

E
 |
 T

$$E \rightarrow T \mid T + E$$

$$T \rightarrow \text{int} \mid \text{int} * T \mid (E)$$

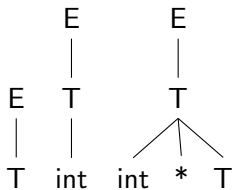
Parse tree pt: (int)
 \uparrow



$$E \rightarrow T \mid T + E$$

$$T \rightarrow \text{int} \mid \text{int} * T \mid (E)$$

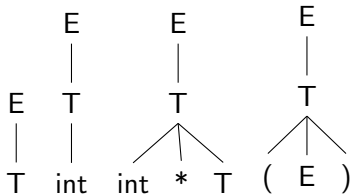
Parse tree pt: (int)
 \uparrow



$$E \rightarrow T \mid T + E$$

$$T \rightarrow int \mid int * T \mid (E)$$

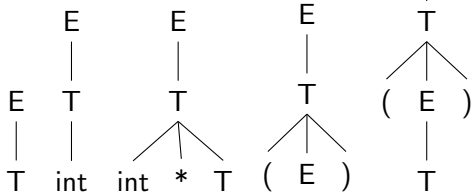
Parse tree pt: $(\quad int \quad)$
 \uparrow



$$E \rightarrow T \mid T + E$$

$$T \rightarrow \text{int} \mid \text{int} * T \mid (E)$$

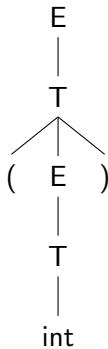
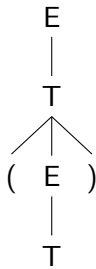
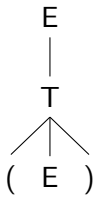
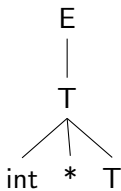
Parse tree pt: (int)
 ↑



$$E \rightarrow T \mid T + E$$

$$T \rightarrow \text{int} \mid \text{int} * T \mid (E)$$

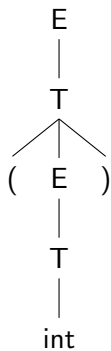
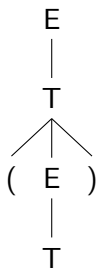
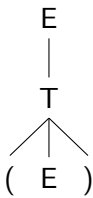
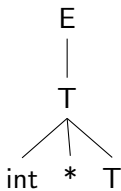
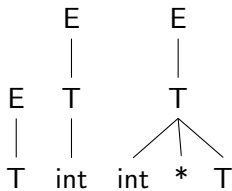
Parse tree pt: (int)
 \uparrow



$$E \rightarrow T \mid T + E$$

$$T \rightarrow \text{int} \mid \text{int} * T \mid (E)$$

Parse tree pt: (int)
 ↑



Recursive-descent parser - neștiind care dintre producțiile alternative pt un nonterminal trebuie aplicată, există posibilitatea de eșec

- Predictive parser: dat fiind sirul de intrare a (primul din sirul ramas) și nonterminalul A care trebuie expandat, am putea determina care producție alternativă e cea care derivatează stringul ramas după a
idee: alternativă corectă trebuie detectată uitându-ne înainte la k simboluri din stringul care trebuie derivate: $LL(1)$ și $LR(1)$

2. Parser predictiv¹

Fie $G = (T, N, P, Z)$ o CFG si automatul stiva

$A = (T, \{q\}, R, q, \{q\}, V, Z)$ cu $V = T \cup N$ si R :

(alfabet, stari, productii, stare initiala, stari finale, alfabet stiva, continut initial stiva)

$$\{tgt \rightarrow q | t \in T\} \cup \\ \{Xq \rightarrow x_n \dots x_1 q | X \rightarrow x_1 x_2 \dots x_n \in P, n \geq 0, X \in N, X_i \in V\}$$

Automatul accepta un sir din $L(G)$ prin

- ▶ construirea unei **derivari cea mai din stanga** a aceluia sir si
- ▶ compararea simbolurilor generate (de la stanga la dreapta) si incarcate pe stiva cu simbolurile care apar in sir.

¹Letia& Chifu 4.2

exemplu 1

Fie $G_1 = (T, N, S, P)$

- ▶ $T = \{+, (,), i\}$, $N = \{S, F\}$
- ▶ cu productiile P
 - ▶ $S \rightarrow F$
 - ▶ $S \rightarrow (S + F)$
 - ▶ $F \rightarrow i$

Care e automatul pentru analiza descendenta?

Care e derivarea stanga pentru $(i + i)$?

Automatul accepta $(i + i)$?

exemplu 2

Fie $G_1 = (T, N, E, P)$

- ▶ $T = \{+, *, (,), i\}$, $N = \{E, T, F\}$
- ▶ cu productiile P
 - ▶ $(1, 2) E \rightarrow T \mid E + T$
 - ▶ $(3, 4) T \rightarrow F \mid T * F$
 - ▶ $(5, 6) F \rightarrow i \mid (E)$

Automatul stiva construit pentru analiza descendenta:

- ▶ $T = \{+, *, (,), i\}$, $Q = \{q\}$,
 $q_0 = q$, $F = \{q\}$, $S = \{+, -, *, (,), i, E, T, F\}$, $s_0 = E$
- ▶ cu productiile R
 1. $Eq \rightarrow Tq$, $Eq \rightarrow T + Eq$,
 2. $Tq \rightarrow Fq$, $Tq \rightarrow F * Tq$,
 3. $Fq \rightarrow iq$, $Fq \rightarrow)E(q$,
 4. $+q+ \rightarrow q$, $*q* \rightarrow q$, $(q(\rightarrow q,)q) \rightarrow q$, $iqi \rightarrow q\}$

Derivarea gasita: $i+i*i$

stiva	stare	intrare	derivarea cea mai din stanga
E	q	$i + i * i$	E
T+E	q	$i + i * i$	E+T
T+T	q	$i + i * i$	T+T
T+F	q	$i + i * i$	F+T
T+i	q	$i + i * i$	i+T
T+	q	$+ i * i$	
T	q	$i * i$	
F*T	q	$i * i$	i+T*F
F*F	q	$i * i$	i+F*F
F*i	q	$i * i$	i+i*F
F*	q	$* i$	
F	q	i	
i	q	i	i+i*i
	q		

Exemplul 3

Fie $G_1 = (T, N, E, P)$

- ▶ $T = \{a, b, c\}$, $N = \{Z, X, Y\}$
- ▶ cu productiile P
 - ▶ (1) $Z \rightarrow X$
 - ▶ (2,3) $X \rightarrow Y|bYa$
 - ▶ (4,5) $Y \rightarrow c|ca$

Automatul $(\{a, b, c\}, \{q\}, R, q, \{q\}, \{a, b, c, X, Y, Z\}, Z)$:

- ▶ $aq a \rightarrow q$
- ▶ $bq b \rightarrow q$
- ▶ $cq c \rightarrow q$
- ▶ $Zq \rightarrow Xq$
- ▶ $Xq \rightarrow Yq$
- ▶ $Xq \rightarrow aYbq$
- ▶ $Yq \rightarrow cq$
- ▶ $Yq \rightarrow acq$

bcaa?

3. No backtracking

Analiza descendenta sau predictiva - traseaza derivarea de la simbolul de start la propozitie, prezicand simbolurile care trebuie sa fie prezente.

- ▶ stiva precizeaza sirul din V^* utilizat pentru derivarea restului sirului de la intrare
- ▶ **automat stiva determinist**: pentru gramatici $LL(k)$

Asumptii si structuri ajutatoare: CFG

Presupunem ca CFG (T, N, P, Z) contin

- ▶ $Z \rightarrow S$ singura in care apare Z - daca nu exista o introducem
- ▶ fiecare propozitie se termina cu $\#$ - indica finalul propozitiei
- ▶ productia i are forma

$$X_i \rightarrow \chi_i, \text{ unde } \chi_i = x_{i,1}x_{i,2}\dots x_{i,m}$$

- ▶ $k : \omega$ primele $\min(k, |\omega| + 1)$ simboluri din $\omega\#$

$$k : \omega = \begin{cases} \omega\#, & \text{daca } |\omega| < k \\ \alpha, & \text{daca } \omega = \alpha\gamma \text{ si } |\alpha| = k \end{cases}$$

- ▶ $FIRST_k(\omega)$ setul tuturor capetelor $k : \omega$ terminale ale sirurilor derivabile din ω

$$FIRST_k(\omega) = \{\tau | \exists \nu \in T^* \text{ a.i. } \omega \Rightarrow^* \nu, \tau = k : \nu\}$$

- ▶ $EFF_k(\omega)$ (ε - free first, primul fara ε) - toate sirurile din $FIRST_k(\omega)$ pentru care nu s-a aplicat nicio productie ε in ultimul pas din derivarea cea mai din dreapta

$$EFF_k(\omega) = \{\tau \in FIRST_k(\omega) | \nexists A \in N, \nu \in T^* \text{ a.i. } \omega \Rightarrow^R A\tau\nu \Rightarrow \tau\nu\}$$

- ▶ $FOLLOW(\omega)$ capete k terminale care ar putea urma lui ω ;
 $FOLLOW_k(Z) = \{\#\}$

$$FOLLOW_k(\omega) = \{\tau | \exists \nu \in T^* \text{ a.i. } Z \Rightarrow^* \mu\omega\nu, \tau \in FIRST_k(\nu)\}$$

Exemplu de valori FIRST, FOLLOW pt $k = 1$

- ▶ $T = \{id, *, +, (,)\}$, $N = \{E, E', T, T', F\}$
- ▶ cu productiile P
 - ▶ $Z \rightarrow E$
 - ▶ $E \rightarrow TE'$
 - ▶ $E' \rightarrow +TE' | \varepsilon$
 - ▶ $T \rightarrow FT'$
 - ▶ $T' \rightarrow *FT' | \varepsilon$
 - ▶ $F \rightarrow (E) | id$

simbol	$FIRST_1(X)$	$FOLLOW_1(X)$
E	$\{(, id\}$	$\{), \#\}$
E'	$\{+, \varepsilon\}$	$\{), \#\}$
T	$\{(, id\}$	$\{+, \#,)\}$
T'	$\{*, \varepsilon\}$	$\{+, \#,)\}$
F	$\{(, id\}$	$\{*, +, \#,)\}$

Exemplu

$$\begin{aligned} E &\Rightarrow TE' \Rightarrow FT'E' \Rightarrow (E)T'E' \Rightarrow^+ (id) * FT'E' \Rightarrow \\ &(id) * F * T' + TE' \Rightarrow (id) * id * id + id \end{aligned}$$

Exemplu de valori FIRST, FOLLOW pt $k = 1$

- ▶ $T = \{id, *, +, (,)\}$, $N = \{E, E', T, T', F\}$
- ▶ cu productiile P
 - ▶ $Z \rightarrow E$
 - ▶ $E \rightarrow TE'$
 - ▶ $E' \rightarrow +TE' | \varepsilon$
 - ▶ $T \rightarrow FT'$
 - ▶ $T' \rightarrow *FT' | \varepsilon$
 - ▶ $F \rightarrow (E) | id$

simbol	$FIRST_1(X)$	$FOLLOW_1(X)$
E	$\{(, id\}$	$\{), \#\}$
E'	$\{+, \varepsilon\}$	$\{), \#\}$
T	$\{(, id\}$	$\{+, \#,)\}$
T'	$\{*, \varepsilon\}$	$\{+, \#,)\}$
F	$\{(, id\}$	$\{*, +, \#,)\}$

Exemplu

$$\begin{aligned} E &\Rightarrow TE' \Rightarrow FT'E' \Rightarrow (E)T'E' \Rightarrow^+ (id) * FT'E' \Rightarrow \\ &(id) * F * T' + TE' \Rightarrow (id) * id * id + id \end{aligned}$$

Gramatici $LL(k)$

O gramatică independentă de context $G = (T, N, P, Z)$ este $LL(k)$ pentru un $k \geq 0$ dacă pentru derivări arbitrare

$$Z \Rightarrow^L \mu X \chi \Rightarrow \mu \nu \chi \Rightarrow^* \mu \gamma$$

$$Z \Rightarrow^L \mu X \chi \Rightarrow \mu \omega \chi \Rightarrow^* \mu \gamma'$$

unde $\mu, \gamma, \gamma' \in T^*, \nu, \chi, \omega \in V^*, X \in N$

avem următoarea proprietate: $k : \gamma = k : \gamma'$ implică $\nu = \omega$

Fie gramatica $G = (\{i, (, +,)\}, \{Z, E, F\}, P, Z)$ cu productiile

► $Z \rightarrow E$

► $E \rightarrow F$

► $E \rightarrow (E + F)$

► $F \rightarrow i$

$$Z \Rightarrow E \Rightarrow (E+F) \Rightarrow (F+F) \Rightarrow^* (i+i)$$

$$Z \Rightarrow E \Rightarrow (E+F) \Rightarrow ((E+F)+F) \Rightarrow^* ((i+i)+i)$$

$$Z \Rightarrow^L \mu X \chi \Rightarrow \mu \nu \chi \Rightarrow^* \mu \gamma$$

$$Z \Rightarrow^L \mu X \chi \Rightarrow \mu \omega \chi \Rightarrow^* \mu \gamma'$$

unde $\mu, \gamma, \gamma' \in T^*, \nu, \chi, \omega \in V^*, X \in N$

Exemplul 3 - reluat

Fie $G_1 = (T, N, E, P)$

▶ $T = \{a, b, c\}$, $N = \{Z, X, Y\}$

▶ cu productiile P

▶ (1) $Z \rightarrow X$

▶ (2,3) $X \rightarrow Y|bYa$

▶ (4,5) $Y \rightarrow c|ca$

1) $Z \Rightarrow X \Rightarrow Y \Rightarrow c$

2) $Z \Rightarrow X \Rightarrow Y \Rightarrow ca$

3) $Z \Rightarrow X \Rightarrow bYa \Rightarrow bca$

4) $Z \Rightarrow X \Rightarrow bYa \Rightarrow bcaa$

Pentru 1 si 2, s-au aplicat $Y \rightarrow c$ si $Y \rightarrow ca$ dar $1:c=1:ca$.

Pentru 3 si 4, s-au aplicat $Y \rightarrow c$ si $Y \rightarrow ca$ dar $1:ca=1:caa$, la fel si $2:ca = 2:caa$.

Situatie

$$[X_p \rightarrow \mu.\nu; \Omega]$$

$$\mu = x_{p,1} \dots x_{p,j}, \nu = x_{p,j+1} \dots x_{p,n_p},$$

$$|\mu| = j, |\nu| = n_p - j$$

Punctul nu face parte din vocabular. Marcheaza pozitia curenta a analizei in partea dreapta a productiei

ex: $q_7 = [X \rightarrow b.Ya; \#]$

Algoritmul LL(k)

Fie $G = (T, N, P, Z)$. Pt automatul stiva se determina Q si tranzitiile R :

1. $Q = \{q_0\}$ si $R = \emptyset$ cu $q_0 = [Z \rightarrow .S, \{\#\}]$

Obs: $FOLLOW_k(Z) = \{\#\}$. q_0 starea initiala si a stivei.

Automatul se opreste daca aceasta stare se intalneste din nou, stiva este vida, simbolul de intrare urmator este $\#$.

Algoritmul LL(k)

Fie $G = (T, N, P, Z)$. Pt automatul stiva se determina Q si tranzitiile R :

1. $Q = \{q_0\}$ si $R = \emptyset$ cu $q_0 = [Z \rightarrow .S, \{\#\}]$

Obs: $FOLLOW_k(Z) = \{\#\}$. q_0 starea initiala si a stivei.

Automatul se opreste daca aceasta stare se intalneste din nou, stiva este vida, simbolul de intrare urmator este $\#$.

2. fie $q = [X \rightarrow \mu.\nu; \Omega]$ un element al lui Q care inca nu a fost tratat

Algoritmul LL(k)

Fie $G = (T, N, P, Z)$. Pt automatul stiva se determina Q si tranzitiile R :

1. $Q = \{q_0\}$ si $R = \emptyset$ cu $q_0 = [Z \rightarrow .S, \{\#\}]$

Obs: $FOLLOW_k(Z) = \{\#\}$. q_0 starea initiala si a stivei.

Automatul se opreste daca aceasta stare se intalneste din nou, stiva este vida, simbolul de intrare urmator este $\#$.

2. fie $q = [X \rightarrow \mu.\nu; \Omega]$ un element al lui Q care inca nu a fost tratat
3. Daca $\nu = \varepsilon$ atunci se include $q\varepsilon \rightarrow \varepsilon$ in R .

Algoritmul LL(k)

Fie $G = (T, N, P, Z)$. Pt automatul stiva se determina Q si tranzitiile R :

1. $Q = \{q_0\}$ si $R = \emptyset$ cu $q_0 = [Z \rightarrow .S, \{\#\}]$

Obs: $FOLLOW_k(Z) = \{\#\}$. q_0 starea initiala si a stivei.

Automatul se opreste daca aceasta stare se intalneste din nou, stiva este vida, simbolul de intrare urmator este $\#$.

2. fie $q = [X \rightarrow \mu.\nu; \Omega]$ un element al lui Q care inca nu a fost tratat
3. Daca $\nu = \varepsilon$ atunci se include $q\varepsilon \rightarrow \varepsilon$ in R .
4. Daca $\nu = t\gamma$, $t \in T$ si $\gamma \in V^*$, fie $q' = [X \rightarrow \mu t.\gamma; \Omega]$.
Adauga q' in Q si $qt \rightarrow q'$ in R .

Algoritmul LL(k)

Fie $G = (T, N, P, Z)$. Pt automatul stiva se determina Q si tranzitiile R :

1. $Q = \{q_0\}$ si $R = \emptyset$ cu $q_0 = [Z \rightarrow .S, \{\#\}]$
Obs: $FOLLOW_k(Z) = \{\#\}$. q_0 starea initiala si a stivei.
Automatul se opreste daca aceasta stare se intalneste din nou, stiva este vida, simbolul de intrare urmator este $\#$.
2. fie $q = [X \rightarrow \mu.\nu; \Omega]$ un element al lui Q care inca nu a fost tratat
3. Daca $\nu = \varepsilon$ atunci se include $q\varepsilon \rightarrow \varepsilon$ in R .
4. Daca $\nu = t\gamma$, $t \in T$ si $\gamma \in V^*$, fie $q' = [X \rightarrow \mu t.\gamma; \Omega]$.
Aaduga q' in Q si $qt \rightarrow q'$ in R .
5. Daca $\nu = Y\gamma$, $Y \in N$ si $\gamma \in V^*$,
 - ▶ fie $q' = [X \rightarrow \mu Y.\gamma; \Omega]$
 - ▶ si $H = \{[Y \rightarrow \cdot\beta_i; FIRST_k(\gamma\Omega)] \mid Y \rightarrow \beta_i \in P\}$.
 - ▶ actualizeaza $Q = Q \cup \{q'\} \cup H$
 - ▶ si $R = R \cup \{q\tau_i \rightarrow q'h_i\tau_i \mid h_i \in H, \tau_i \in FIRST_k(\beta_i\gamma\Omega)\}$
6. daca toate starile din q au fost analizate, stop. Altfel continua cu 2.

Construirea automatului se termina datorita numarului finit de situatii.

Automatul rezultat este determinist daca si numai daca G este o gramatica $LL(k)$.

Exemplu de construire (gramatica e CFG dar si regulata)

- ▶ $Z \rightarrow S$
- ▶ $S \rightarrow 0S$
- ▶ $S \rightarrow 1$

k=1

		stari noi	tranzitii noi
			$q_0 = [Z \rightarrow .S; \{\#\}]$
q_0	5	$q' = [Z \rightarrow S.; \#] = q_1$ $H = \{[S \rightarrow .0S; \#] = q_2,$ $[S \rightarrow .1; \#] = q_3\}$	$q_0 \tau? \rightarrow q_1 h? \tau?$ $q_0 0 \rightarrow q_1 q_2 0$ $q_0 1 \rightarrow q_1 q_3 1$
q_1	3	-	$q_1 \varepsilon \rightarrow \varepsilon$
q_2	4	$q' = [S \rightarrow 0.S; \#] = q_4$	$q_2 0 \rightarrow q_4$
q_3	4	$q' = [S \rightarrow 1.; \#] = q_5$	$q_3 1 \rightarrow q_5$
q_4	5	$q' = [S \rightarrow 0S.; \#] = q_6$ H = la fel cu analiza pt q_0	? $q_4 0 \rightarrow q_6 q_2 0$ $q_4 1 \rightarrow q_6 q_3 1$
q_5	3	-	$q_5 \varepsilon \rightarrow \varepsilon$
q_6	3	-	$q_6 \varepsilon \rightarrow \varepsilon$

derivare

Care e derivarea pt $001\#$?

$$q_0 q_0 001\# \Rightarrow ?$$

Incercare cu $k = 1$; $Z \rightarrow S$, $S \rightarrow 0S1$, $S \rightarrow 01$

		stari noi	No	tranzitii noi
		$q_0 = [Z \rightarrow .S; \{\#\}]$		
q_0	5	$q' = [Z \rightarrow S.; \#] = q_1$ $H = \{[S \rightarrow .0S1; \#] = q_2,$ $[S \rightarrow .01; \#] = q_3\}$	1 2	$q_0 \tau? \rightarrow q_1 h? \tau?$ $\tau \in FIRST_1(0S1\#);$ $q_0 0 \rightarrow q_1 q_2 0$ $\tau \in FIRST_1(01\#);$ $q_0 0 \rightarrow q_1 q_3 0$
q_1	3	-	3	$q_1 \epsilon \rightarrow \epsilon$
q_2	4	$q' = [S \rightarrow 0.S1; \#] = q_4$	4	$q_2 0 \rightarrow q_4$
q_3	4	$q' = [S \rightarrow 0.1; \#] = q_5$	5	$q_3 0 \rightarrow q_5$
q_4	5	$q' = [S \rightarrow 0S.1; \#] = q_6$ $H = \{[S \rightarrow .0S1; FIRST_1(1\#)]$ $= q_7,$ $[S \rightarrow .01; FIRST_1(1\#)] = q_8\}$	6 7	$\tau \in FIRST_1(0S11\#);$ $q_4 0 \rightarrow q_6 q_7 0$ $\tau \in FIRST_1(011\#);$ $q_4 0 \rightarrow q_6 q_8 0$
q_5	4	$[S \rightarrow 01.; \#] = q_9$	8	$q_5 1 \rightarrow q_9$
q_6	4	$[S \rightarrow 0S1.; \#] = q_{10}$	9	$q_6 1 \rightarrow q_{10}$
q_9	3		10	$q_9 \epsilon \rightarrow \epsilon$
	...			
	3			
			

$k=1$: automat nedeterminist: $q_00 \rightarrow q_1q_20$ si $q_00 \rightarrow q_1q_30$

Derivare 01: cu un lookahead de 1 nu stim pe care productie sa o aplicam

$$q_0q_001\# \xRightarrow{1} q_0q_1q_201\# \xRightarrow{4} q_0q_1q_41\# \quad \text{deadend}$$

$$q_0q_001\# \xRightarrow{2} q_0q_1q_301\# \xRightarrow{5} q_0q_1q_51\# \xRightarrow{8} q_0q_1q_9\# \xRightarrow{10} q_0q_1\varepsilon\# \Rightarrow q_0\#$$

Inercare cu $k = 2$; $Z \rightarrow S$, $S \rightarrow 0S1$, $S \rightarrow 01$

		stari noi	No	tranzitii noi
		$q_0 = [Z \rightarrow .S; \{\#\}]$		
q_0	5	$q' = [Z \rightarrow S.; \#] = q_1$ $H = \{[S \rightarrow .0S1; \#] = q_2,$ $[S \rightarrow .01; \#] = q_3\}$	1 2	$q_0\tau? \rightarrow q_1h?\tau?$ $\tau \in FIRST_2(0S1\#);$ $q_000 \rightarrow q_1q_200$ $\tau \in FIRST_2(01\#);$ $q_001 \rightarrow q_1q_301$
q_1	3	-	3	$q_1\varepsilon \rightarrow \varepsilon$
q_2	4	$q' = [S \rightarrow 0.S1; \#] = q_4$	4	$q_20 \rightarrow q_4$
q_3	4	$q' = [S \rightarrow 0.1; \#] = q_5$	5	$q_30 \rightarrow q_5$
q_4	5	$q' = [S \rightarrow 0S.1; \#] = q_6$ $H = \{[S \rightarrow .0S1; FIRST_2(1\#)]$ $= q_7,$ $[S \rightarrow .01; FIRST_2(1\#)] = q_8\}$	6 7	$\tau \in FIRST_2(0S11\#);$ $q_400 \rightarrow q_6q_700$ $\tau \in FIRST_2(011\#);$ $q_401 \rightarrow q_6q_801$
q_5	4	$[S \rightarrow 01.; \#] = q_9$	8	$q_51 \rightarrow q_9$
q_6	4	$[S \rightarrow 0S1.; \#] = q_{10}$	9	$q_61 \rightarrow q_{10}$
q_7	4	$[S \rightarrow 0.S1; 1\#] = q_{11}$	10	$q_70 \rightarrow q_{11}$
q_8	4	$[S \rightarrow 0.1; 1\#] = q_{12}$	11	$q_80 \rightarrow q_{12}$
q_9	3		10	$q_9\varepsilon \rightarrow \varepsilon$
q_{10}	3		10	$q_{10}\varepsilon \rightarrow \varepsilon$
q_{11}	5	$q' = [S \rightarrow 0S.1; \{1\#}] = q_{13}$ $H = \{[S \rightarrow .0S1; FIRST_2(11\#)]$ $= q_{14},$ $[S \rightarrow .01; FIRST_2(11\#)] = q_{15}\}$	6 7	$\tau \in FIRST_2(0S111\#);$ $q_{11}00 \rightarrow q_{13}q_{14}00$ $\tau \in FIRST_2(0111\#);$ $q_{11}01 \rightarrow q_{13}q_{15}01$

Derivare $Z \Rightarrow 0011$

Stiva	Stare	Intrare	Derivarea cea mai din stanga
q_0	q_0	0011#	Z
$q_0 q_1$	q_2	0011#	S
$q_0 q_1$	q_4	011#	0S1
$q_0 q_1 q_6$	q_8	011#	0011
$q_0 q_1 q_6$	q_{12}	11#	
....			

Gramatica $LL(3)$

- ▶ $Z \rightarrow X$
- ▶ $X \rightarrow Y|bYa$
- ▶ $Y \rightarrow c|ca$

		stari noi		tranzitii noi
		$q_0 = [Z \rightarrow .X; \#]$		
q_0	5	$q' = [Z \rightarrow X.; \#] = q_1$ $H = \{[X \rightarrow .Y; \#] = q_2,$ $[X \rightarrow .bYa; \#] = q_3\}$	1 2 3	$\tau \in FIRST_3(Y\#) = \{c\#, ca\#$ $q_0c\# \rightarrow q_1q_2c\#$ $q_0ca\# \rightarrow q_1q_2ca\#$ $\tau \in FIRST_3(bYa) = \{bca\}$ $q_0bca \rightarrow q_1q_3bca$
q_2	5	$q' = [X \rightarrow Y.; \#] = q_4$ $H = \{[Y \rightarrow .c; \#] = q_5,$ $Y \rightarrow .ca; \#] = q_6\}$ 	4 5	$\tau \in FIRST_3(c\#) = \{c\#$ $q_2c\# \rightarrow q_4q_5c\#$ $q_2ca\# \rightarrow q_4q_6ca\#$

Gramatica $LL(3)$

- ▶ $Z \rightarrow X$
- ▶ $X \rightarrow Y|bYa$
- ▶ $Y \rightarrow c|ca$

$$q_0 = [Z \rightarrow \bullet X; \#]$$

$$q_1 = [Z \rightarrow X \bullet; \#]$$

$$q_2 = [X \rightarrow \bullet Y; \#]$$

$$q_3 = [X \rightarrow \bullet bYa; \#]$$

$$q_4 = [X \rightarrow Y \bullet; \#]$$

$$q_5 = [Y \rightarrow \bullet c; \#]$$

$$q_6 = [Y \rightarrow \bullet ca; \#]$$

$$q_7 = [X \rightarrow b \bullet Ya; \#]$$

$$q_8 = [Y \rightarrow c \bullet; \#]$$

$$q_9 = [Y \rightarrow c \bullet a; \#]$$

$$q_{10} = [X \rightarrow bY \bullet a; \#]$$

$$q_{11} = [Y \rightarrow \bullet c; a\#]$$

$$q_{12} = [Y \rightarrow \bullet ca; a\#]$$

$$q_{13} = [Y \rightarrow ca \bullet; \#]$$

$$q_{14} = [X \rightarrow bYa \bullet; \#]$$

$$q_{15} = [Y \rightarrow c \bullet; a\#]$$

$$q_{16} = [Y \rightarrow c \bullet a; a\#]$$

$$q_{17} = [Y \rightarrow ca \bullet; a\#]$$

$q_0 = [Z \rightarrow \bullet X; \#]$	$q_9 = [Y \rightarrow c \bullet a; \#]$
$q_1 = [Z \rightarrow X \bullet; \#]$	$q_{10} = [X \rightarrow bY \bullet a; \#]$
$q_2 = [X \rightarrow \bullet Y; \#]$	$q_{11} = [Y \rightarrow \bullet c; a\#]$
$q_3 = [X \rightarrow \bullet bYa; \#]$	$q_{12} = [Y \rightarrow \bullet ca; a\#]$
$q_4 = [X \rightarrow Y \bullet; \#]$	$q_{13} = [Y \rightarrow ca \bullet; \#]$
$q_5 = [Y \rightarrow \bullet c; \#]$	$q_{14} = [X \rightarrow bYa \bullet; \#]$
$q_6 = [Y \rightarrow \bullet ca; \#]$	$q_{15} = [Y \rightarrow c \bullet; a\#]$
$q_7 = [X \rightarrow b \bullet Ya; \#]$	$q_{16} = [Y \rightarrow c \bullet a; a\#]$
$q_8 = [Y \rightarrow c \bullet; \#]$	$q_{17} = [Y \rightarrow ca \bullet; a\#]$

$$\begin{aligned}
R = \{ & q_0c\# \rightarrow q_1q_2c\#, & q_7ca\# \rightarrow q_{10}q_{11}ca\# \\
& q_0ca\# \rightarrow q_1q_2ca\#, & q_7caa \rightarrow q_{10}q_{12}caa, \\
& q_0bca \rightarrow q_1q_3bca, & q_8 \rightarrow \epsilon, \\
& q_1 \rightarrow \epsilon, & q_9a \rightarrow q_{13}, \\
& q_2c\# \rightarrow q_4q_5c\#, & q_{10}a \rightarrow q_{14}, \\
& q_2ca\# \rightarrow q_4q_6ca\#, & q_{11}c \rightarrow q_{15}, \\
& q_3b \rightarrow q_7, & q_{12}c \rightarrow q_{16}, q_{13} \rightarrow \epsilon, \\
& q_4 \rightarrow \epsilon, & q_{14} \rightarrow \epsilon, \\
& q_5c \rightarrow q_8, & q_{15} \rightarrow \epsilon, \\
& q_6c \rightarrow q_9, & q_{16}a \rightarrow q_{17}, q_{17} \rightarrow \epsilon \}
\end{aligned}$$

aceeasi gramatica dar cu $k = 2$

$$q_7ca \rightarrow q_{10}q_{11}ca$$

$$q_7ca \rightarrow q_{10}q_{12}ca$$

Cu $k = 3$

$$q_7ca\# \rightarrow q_{10}q_{11}ca\#$$

$$q_7caa \rightarrow q_{10}q_{12}caa$$

unde pt $k = 3$

- ▶ $q_7 = [X \rightarrow b.Ya; \#]$
- ▶ $q_{10} = [X \rightarrow bY.a; \#]$
- ▶ $q_{11} = [Y \rightarrow .c; a\#]$
- ▶ $q_{12} = [Y \rightarrow .ca; a\#]$

Derivare $Z \Rightarrow X \Rightarrow bYa \Rightarrow bcaa$

Stiva	Stare	Intrare	Derivarea cea mai din stanga
q_0	q_0	$bcaa\#$	Z
q_0q_1	q_3	$bcaa\#$	X
q_0q_1	q_7	$caa\#$	bYa
$q_0q_1q_{10}$	q_{12}	$caa\#$	$bcaa$
$q_0q_1q_{10}$	q_{16}	$aa\#$	
$q_0q_1q_{10}$	q_{17}	$a\#$	
q_0q_1	q_{10}	$a\#$	
q_0q_1	q_{14}	$\#$	
q_0	q_1	$\#$	
	q_0	$\#$	

- ▶ La **tranzitiile de stivuire** sunt examinate simbolurile dinainte (lookaheads symbols).
- ▶ Aceste tranzitii corespund **intrarii intr-o productie noua**
- ▶ **Citirea simbolurilor terminale si decizia de terminare a productiei printr-o tranzitie de destivuire** se realizeaza fara inspectarea simbolurilor dinainte

Rezumat

Recursive descent parsing

Predictive parsing

Structuri ajutatoare: FIRST, FOLLOW

Definire Gramatici LL(k)

Algoritmul LL(K)

Exemplu aplicare LL(k)

Exemplu aplicare LL(3)

$$Z \Rightarrow X \Rightarrow ca$$

$$Z \Rightarrow X \Rightarrow bca$$