

Gramatichi LR(k)

- analizoare LR(k)
 - analiza sintactica ascendentă
 - secv. de intrare este citita de la stanga spre dreapta
 - se folosesc derivari de dreapta
- metoda: deplasare - reducere

Analiza sintactica ascendenta

Exemplu:

Cum “arata” analiza sintactica ascendenta pentru gramatica:

$$S \rightarrow AB \quad (1)$$

$$A \rightarrow a \quad (2)$$

$$B \rightarrow b \quad (3)$$

si intrarea: ab ?

Gramatica LR(K)

Analizoarele sintactice LR(k) lucreaza cu
gramatica imbogatita:

$$G' = (N \cup \{S'\}, \Sigma, P \cup \{S' \rightarrow S\}, S')$$
$$(S' \notin N)$$

Gramatica LR(K)

O gramatica $G = (N, \Sigma, P, S)$

este de tip $LR(k)$ pentru $k \geq 0$

daca din:

- $S' \xrightarrow{*_{dr}} \alpha A w \Rightarrow_{dr} \alpha \beta w$
- $S' \xrightarrow{*_{dr}} \gamma B x \Rightarrow_{dr} \alpha \beta y$
- $\text{FIRST}_k(w) = \text{FIRST}_k(y)$

rezulta ca:

- $A = B$
- $x = y$
- $\alpha = \gamma$

Gramatici LR(K) - terminologie

Prefix viabil

Fie: $S =^* \Rightarrow_{dr} \alpha A w \Rightarrow_{dr} \alpha \beta w$

Orice prefix al lui $\alpha\beta$ se numeste prefix viabil

Element de analiza LR(k)

se defineste ca fiind: $[A \rightarrow \alpha.\beta, u]$

unde $A \rightarrow \alpha\beta \in P$ si $u \in \Sigma^k$ u-predictie

Element de analiza valid

$[A \rightarrow \alpha.\beta, u]$ valid pentru prefixul viabil $\gamma\alpha$ daca:

- $S =^* \Rightarrow_{dr} \gamma A w \Rightarrow_{dr} \gamma \alpha \beta w$
- $u = \text{FIRST}_k(w)$

Analizor sintactic LR(K)

Vom studia:

- LR(0)
- SLR
- LR(1)
- LALR

Pasi in analiza LR(k):

- gramatica imbogatita
- constructia colectiei canonice
- constructia tabelului de analiza
- analiza: → automat

Vom lucra astfel:

la multimea cuvintelor de analizat se adauga la sfarsit \$

\$ - marcator de sfarsit de cuvant

Colectia canonica LR(K)

$C = \{I_i - \text{elementele de analiza pentru un prefix viabil}\}$

- in I_0 avem un prim element de analiza
- am cel putin un element in I_j *(pentru fiecare)*
=> adaug altele: functia *Closure*
- am o multime I_j *(pentru fiecare)*
=> construiesc multimile *goto*(I_i, X)

Observatie: I_i corespunde unei stari a automatului

Notatie: E – multimea elementelor de analiza

Constructia colectiei canonice LR(k)

$C = \{ I_i\text{-elementele de analiza pentru un prefix viabil} \}$

in I_0 avem: $[S' \rightarrow .S, \dots]$

- $I_0 = Closure ([S' \rightarrow .S, \dots])$

- $C = \{ I_0 \}$

- repeta

pentru toti I_i din C , $X \in (N \cup \Sigma)$ executa

$$C = C \cup goto(I_i, X)$$

sf. pentru

pana cand C nu se mai modifica

K=0: **LR(0)**

Gramatica imbogatita

- se adauga S'
 - nou simbol de start
 - $S' \rightarrow S$
-

Colectia canonica:

- In I_0 avem: [$S' \rightarrow .S$]
- ...

Functia *Closure* LR(0)

Notam: \mathcal{E} – multimea elementelor de analiza

- $\text{Closure} : \text{Part}(\mathcal{E}) \rightarrow \text{Part}(\mathcal{E})$
- Fie: $e \in \mathcal{E}$
daca $e = [A \rightarrow \alpha . B\beta]$
atunci $\forall B \rightarrow \delta \in P: [B \rightarrow . \delta] \in \text{Closure}(e)$

Functia *goto* LR(0)

- $goto : \text{Part}(E) \times (N \cup \Sigma) \rightarrow \text{Part}(E)$
- $goto(I, X) = \text{Closure}(\{[A \rightarrow \alpha X. \beta] \mid [A \rightarrow \alpha. X\beta] \in I\})$

Tabelul de analiza LR(0)

		goto $N \cup \Sigma$
actiune		
I_0	deplasare (s)	
I_1	reducere (nr. r.p.)	
:	acceptare (acc)	
	eroare	

Tabelul de analiza LR(0)

$T(I_i, \text{actiune}) =$

– s (shift, deplasare)

daca: $[A \rightarrow \alpha.\beta] \in I_i, \beta \not\sim \epsilon$

si: $T(I_i, X) = I_j$, daca $I_j = goto(I_i, X)$

– L (reducere cu r.p. nr. L)

daca $[A \rightarrow \alpha.] \in I_i$

$A \rightarrow \alpha \in P$: regula de prod. cu numarul L

si: $T(I_i, X)$ nu se completeaza

– acc daca: $[S' \rightarrow S.] \in I_i$

Toate celelalte cazuri se considera eroare .

Automatul LR(0) – model matematic

- configuratie:
 (α, β, Π)
(stiva_de_lucru, banda_de_intrare, banda_de_iesire)
- pe stiva: prefixe viabile, stari ale analizorului
- config. initiala: $(\$0, w \$, \varepsilon)$
- config. finala: $(\$0S I_{acc}, \$, \Pi)$

Automatul LR(0) – model matematic

Tranzitii

- **deplasare:**

$$(\$, \gamma s_k, a_i..a_n \$, \Pi) \vdash (\$, \gamma s_k a_i s_m, a_{i+1}..a_n \$, \Pi)$$

daca: $T(s_k, \text{actiune}) = s$ si $T(s_k, a_i) = s_m$

- **reducere:**

$$(\$, \gamma s_{p-1} X_p s_p ... X_k s_k, a_i..a_n \$, \Pi) \vdash (\$, \gamma s_{p-1} A s_m, a_i..a_n \$, L\Pi)$$

daca: $T(s_k, \text{action}) = L$

si: $A \rightarrow X_p ... X_k$ – r.p. cu nr. L

$T(s_{p-1}, \text{actiune}) = s$

$T(s_{p-1}, A) = s_m$

- **acceptare:**

$$(\$, 0S s_{\text{acc}}, \$, \Pi) \vdash \text{acc.}$$

- **eroare:**

orice alta situatie

Gramatica LR(0)

O gramatica este LR(0)
daca tabelul de analiza nu contine conflicte.

Gramatica data prin urmatoarele r.p. este LR(0) ?

$$S \rightarrow Ax$$

$$S \rightarrow By$$

$$A \rightarrow a$$

$$B \rightarrow a$$

K=1: SLR, LR(1), LALR

Analiza sintactica SLR

- SLR = Simple LR
- element de analiza SLR:
 $[A \rightarrow \alpha.\beta, u]$
 $u = FOLLOW_1(A)$
 $|u| = 1$
- constructia colectiei canonice ($\sim LR(0)$)
 - $[A \rightarrow \alpha.\beta, u]$, $u = FOLLOW_1(A)$
- SLR tine cont de predictie numai pentru reducere

Analiza sintactica SLR

- constructia tabelului de analiza SLR
 - actiunea de reducere depinde de predictia u
 \Rightarrow reducerea va avea o coloana pentru fiecare $a \in \Sigma$
 - tabelul: linii: elementele colectiei canonice
coloane: $N \cup \Sigma \cup \{\$\}$
celula: $s_{\text{stare}}, r_{\text{nr.r.p}}, \text{acc}$
- analizorul ~ analizorul pt. LR(0)
 - automat: configuratii si tranzitii

Analiza sintactica SLR

In tabelul de analiza SLR vom avea:

actiune: **reducere**

+

deplasare

(**goto**)

$X \in \Sigma \cup \{\$\}$

$X \in N \cup \Sigma$

linii: elementele colectiei canonice

coloane: $N \cup \Sigma \cup \{\$\}$

Gramatica SLR

O gramatica este SLR
daca tabelul de analiza nu contine conflicte.

Gramatica data prin urmatoarele r.p. este SLR ?

$S \rightarrow A$	1
$S \rightarrow xb$	2
$A \rightarrow aAb$	3
$A \rightarrow B$	4
$B \rightarrow x$	5

- Gramatica data prin urmatoarele r.p. este SLR ?

$S \rightarrow A$	1
$S \rightarrow xb$	2
$A \rightarrow aAb$	3
$A \rightarrow B$	4
$B \rightarrow x$	5

Analizor sintactic LR(1)

- imbogatirea gramaticii
- constructia colectiei canonice element de analiza LR(1):
 - $[A \rightarrow \alpha.\beta, u], |u| = 1$
- constructia tabelului de analiza
- analiza: automat

Colectia canonica LR(1)

- elem. initial
 $[S' \rightarrow .S, \$]$
- *Closure*
 $[A \rightarrow \alpha.B\beta, a] \Rightarrow [B \rightarrow .\gamma, b] \in Closure([A \rightarrow \alpha.B\beta, a])$
 $B \rightarrow \gamma \quad \forall b \in FIRST_1(\beta a)$
- *goto*
 $goto(I, X) =$
 $Closure(\{[A \rightarrow \alpha X. \beta, a] \mid [A \rightarrow \alpha X \beta, a] \in I\})$

Tabelul LR(1)

		shift + reduce
stare		$N \cup \Sigma \cup \$$
I_0		
I_{acc}	I_1	
	I_2	
	:	
	:	

Construirea tab. de analiza LR(1)

- $[A \rightarrow \alpha.X\beta, b] \in I_i$: $goto(I_i, X) = I_j \leqslant$ functia $goto$
 $action(I_i, X) = s_j$
- $[A \rightarrow \alpha. . , a] \in I_i$ $action(I_i, a) = r_L$
 $L -$ nr. reg. de productie: $A \rightarrow \alpha$
 $A \diamond S'$
- $[S' \rightarrow S. . , \$] \in I_i$ $action(I_i, \$) = acc$

Obs: o gram. este LR(...) daca tabelul de analiza nu contine conflicte; si reciproc

Analizorul LR(1)

pe baza tabelului de analiza
→ similar LR(0), SLR

Analizor sintactic LALR

- $[A \xrightarrow{\text{nucleu}} \alpha.\beta, a]$
 - colectia canonica LR(1)
 - fuzioneaza elementele de analiza cu nuclee identice si care nu creeaza conflicte
 - predictia: reunirea predictiilor
-
- tabelul LALR & analiza : similar LR(1)

LR (1 –uri)

- Conflict:

$[A \rightarrow \alpha_1.a\alpha_2, u]$ $[B \rightarrow \beta_1. \quad , \quad a]$	deplasare-reducere
$[A \rightarrow \alpha_1. \quad , \quad a]$ $[B \rightarrow \beta_1. \quad , \quad a]$	reducere-reducere



Vezi și resursa bibliografică:

S. MOTOGNA

Metode de proiectare a compilatoarelor, 2006