

Curs 8

Analiză sintactică

LR(k)

Termeni

- Predicție – vezi LL(1)
- Manșa = simboluri din vârful stivei de lucru care formează (în ordine) pdp
- Analizor de tip ***deplasare - reducere***:
- **deplasează** simboluri pentru a forma manșa
- când s-a format partea dreaptă a unei producții - **reduce** la partea stângă corespunzătoare

LR(k)

- L = left - secvența este procesată de la stânga la dreapta
- R = right - folosirea derivărilor de dreapta
- k = lungimea predicției
- Gramatică îmbogățită
- $G = (N, \Sigma, P, S)$
- $G' = (N \cup \{S'\}, \Sigma, P \cup \{S' \rightarrow S\}, S'), S' \notin N$

S' nu apare în partea dreaptă
a nici unei producții

- **Definiție:** Dacă într-o gramatică $G = (N, \Sigma, P, S)$ avem
 $S \xRightarrow{*}_{dr} \alpha Aw \Rightarrow_{dr} \alpha \beta w$, unde $\alpha \in (N \cup \Sigma)^*$, $A \in N$, $w \in \Sigma^*$, atunci
 orice prefix al secvenței $\alpha \beta$ se numește ***prefix viabil*** în G .
- **Definiție:** ***Elementul de analiză LR(k)*** se definește ca fiind $[A \rightarrow \alpha.\beta, u]$, unde $A \rightarrow \alpha\beta$ e o producție a gramaticii, $u \in \Sigma^k$ și descrie stadiul în care, având în vedere producția $A \rightarrow \alpha\beta$, a fost detectat α (α e în vârful stivei) și se așteaptă să fie detectat β .
- **Definiție:** Elementul de analiză LR(k) este ***valid pentru prefixul viabil*** $\gamma\alpha$ dacă:

$$S \xRightarrow{*}_{dr} \gamma Aw \Rightarrow_{dr} \gamma \alpha \beta w$$

$$u = \text{FIRST}_k(w)$$

Definiția O gramatică $G = (N, \Sigma, P, S)$ este de tip **LR(k)** pentru $k \geq 0$ dacă din:

$$1. S' \xRightarrow{*}_{dr} \alpha Aw \Rightarrow_{dr} \alpha \beta w;$$

$$2. S' \xRightarrow{*}_{dr} \gamma Bx \Rightarrow_{dr} \alpha \beta y;$$

$$3. FIRST_k(w) = FIRST_k(y)$$

rezultă că $\alpha = \beta, A = B, x = y$.

- $[A \rightarrow \alpha\beta., u]$ - s-a detectat pdp - se poate face reducere
- $[A \rightarrow \alpha.\beta, u]$ - deplasare

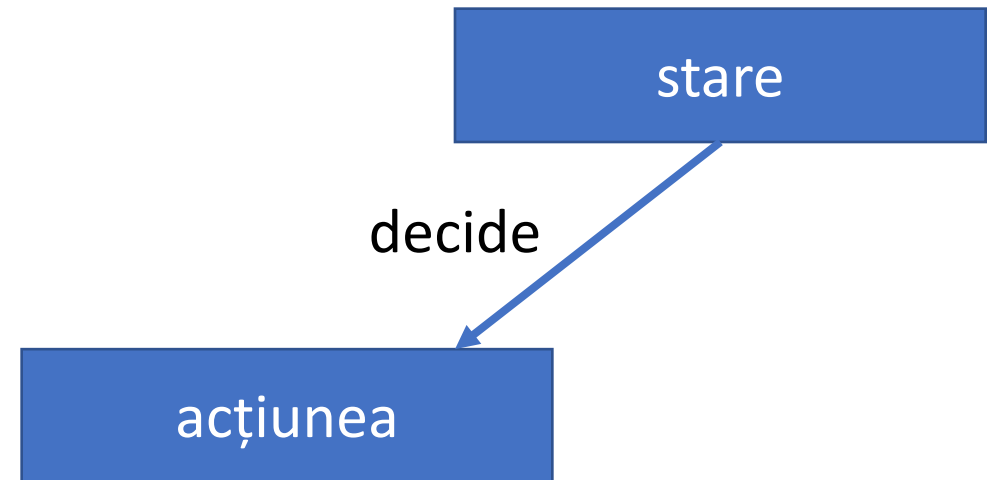
\Rightarrow Stiva de lucru:

$\$s_{init}X_1s_1 \dots X_ms_m$

unde: $\$$ - marcaj stivă vidă

$X_i \in N \cup \Sigma$

s_i - stări



Principiul LR(k)

- starea curentă
- simbolul curent
- predicția

Determină în mod unic:

- acțiunea care se aplică
- tranziția în altă stare

=> Tabel LR(k) – 2 părți: parte de acțiune + parte de goto

Stări

Ce conține o stare?

- Elemente de analiză
- Închidere - *closure*

Câte stări sunt? Cum se trecere dintr-o stare în altă stare

- *goto*

- $[A \rightarrow \alpha.B\beta, u]$ valabil pentru prefixul viabil $\gamma\alpha \Rightarrow$

$$S \xRightarrow{*}_{dr} \gamma Aw \Rightarrow_{dr} \gamma\alpha B\beta w$$

$$u = FIRST_k(w)$$

- $B \rightarrow \delta \in P \Rightarrow S \xRightarrow{*} \gamma Aw \Rightarrow_{dr} \gamma\alpha B\beta w \Rightarrow_{dr} \gamma\alpha\delta\beta w.$

$\Rightarrow [B \rightarrow .\delta, u]$ valabil pentru prefixul viabil $\gamma\alpha$

Analiză sintactică LR(k): LR(0), SLR, LR(1), LALR

- definirea elementului de analiză
- construirea mulțimii de stări
- construirea tabelului de analiză
- Analiza secvenței de baza tranzițiilor între configurații

Analizor sintactic LR(0)

- Predicție de lungime 0 (se ignoră predicția)

1. Element de analiză LR(0): $[A \rightarrow \alpha.\beta]$

2. Construirea mulțimii de stări

- Ce conține o stare – *closure_LR0*
- Cum se trece dintr-o stare în altă stare – *goto_LR0*
- Construirea mulțimii de stări – *ColCan_LR0*

Colecție canonică

Algorithm *Closure*

INPUT: I-element de analiză; G' - gramatica îmbogățită

OUTPUT: $C = \text{closure}(I)$;

$C := \{I\}$;

repeat

for $\forall [A \rightarrow \alpha.B\beta] \in C$ **do**

for $\forall B \rightarrow \gamma \in P$ **do**

if $[B \rightarrow \cdot\gamma] \notin C$ **then**

$C = C \cup [B \rightarrow \cdot\gamma]$

end if

end for

end for

until C nu se mai modifică

Funcția *goto*

$$\text{goto} : P(\mathcal{E}_0) \times (N \cup \Sigma) \rightarrow P(\mathcal{E}_0)$$

unde \mathcal{E}_0 = mulțimea de elemente LR(0)

$$\text{goto}(s, X) = \text{closure}(\{[A \rightarrow \alpha X.\beta] \mid [A \rightarrow \alpha.X\beta] \in s\})$$

Algorithm *ColCan_LR(0)*

INPUT: G' - gramatica îmbogățită

OUTPUT: C - colecția canonică de stări

$C := \emptyset;$

$s_0 := closure(\{[S' \rightarrow .S]\})$

$C := C \cup \{s_0\};$

repeat

for $\forall s \in C$ **do**

for $\forall X \in N \cup \Sigma$ **do**

if $goto(s, X) \neq \emptyset$ and $goto(s, X) \notin C$ **then**

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

end if

end for

end for

until C nu se mai modifică

3. Construirea tabelului de analiză

- câte o linie pentru fiecare stare
- 2 părți:
 - De acțiune: o coloană (pentru o anumită stare acțiunea este unică deoarece nu se ține cont de predicție)
 - De goto: câte o coloană pentru fiecare simbol al gramaticii $X \in N \cup \Sigma$

Reguli tabel LR(0)

1. dacă $[A \rightarrow \alpha.\beta] \in s_i$ atunci acțiune(s_i)=**shift**
2. dacă $[A \rightarrow \beta.] \in s_i$ și $A \neq S'$ atunci acțiune(s_i)=**reduce l**, unde l - numărul producției $A \rightarrow \beta$
3. dacă $[S' \rightarrow S.] \in s_i$ atunci acțiune(s_i)=**acc**
4. dacă $\text{goto}(s_i, X) = s_j$ atunci **goto(s_i, X) = s_j**
5. toate celelalte valori = **eroare**

Observații

- 1) starea inițială a analizorului = starea care conține $[S' \rightarrow .S]$
- 2) Din starea de acceptare nu există deplasare: dacă s e stare de acceptare,

$$\text{goto}(s, X) = \emptyset, \forall X \in N \cup \Sigma.$$

- 3) Dacă într-o anumită stare s acțiunea este de reducere, atunci

$$\text{goto}(s, X) = \emptyset, \forall X \in N \cup \Sigma.$$

- 4) Argument G' : fie $G = (\{S\}, \{a, b, c\}, \{S \rightarrow aSbS, S \rightarrow c\}, S)$

stările $[S \rightarrow aSbS.]$ și $[S \rightarrow c.]$ – acceptare / reducere ?

Observații (cont)

- 5) O gramatică nu este de tip LR(0) dacă tabelul de analiză conține conflicte:
- conflict deplasare - reducere: când o stare include elemente de analiză de forma $[A \rightarrow \alpha.\beta]$ și $[B \rightarrow \gamma.]$, ceea ce ar implica două acțiuni diferite pentru această stare
 - conflict reducere - reducere: când o stare include elemente de analiză de forma $[A \rightarrow \alpha\beta.]$ și $[B \rightarrow \gamma.]$, în care acțiunea este de reducere, dar se creează conflict relativ la producția cu care se face reducerea

4. Analiza secvenței de baza tranzițiilor între configurații

- INPUT:

- Gramatica limbajului $G' = (N \cup \{S'\}, \Sigma, P \cup \{S' \rightarrow S\}, S')$
- Tabel de analiză LR(0)
- Secvența de analizat $w = a_1 \dots a_n$

- OUTPUT:

Dacă ($w \in L(G)$) **atunci șir de producții**
altfel **locția erorii**

Configurații LR(0)

(α, β, π)

Unde:

- α = stiva de lucru
- β = stiva de intrare
- π = banda de ieșire (rezultat)

Configurația inițială:
 $(\$s_0, w\$, \varepsilon)$

Configurația finală:
 $(\$s_{acc}, \$, \pi)$

Tranziții

1. Deplasare

dacă $\text{actiune}(s_m) = \text{shift}$ AND $\text{head}(\beta) = a_i$ AND $\text{goto}(s_m, a_i) = S_j$ **atunci**

$$(\$s_0 x_1 \dots x_m s_m, a_i \dots a_n \$, \pi) \vdash (\$s_0 x_1 \dots x_m s_m a_i s_j, a_{i+1} \dots a_n \$, \pi)$$

2. Reducere

dacă $\text{actiune}(s_m) = \text{reduce } t$ AND $(t) A \rightarrow x_{m-p+1} \dots x_m$ AND $\text{goto}(s_{m-p}, A) = s_j$ **atunci**

$$(\$s_0 \dots x_m s_m, a_i \dots a_n \$, \pi) \vdash (\$s_0 \dots x_{m-p} s_{m-p} A s_j, a_i \dots a_n \$, t \pi)$$

3. Acceptare

dacă $\text{actiune}(s_m) = \text{accept}$ **atunci** $(\$s_i, \$, \pi) = \text{acc}$

3. Eroare - altfel

Algoritm de analiză sintactică LR(0)

- [AICI](#)