18 Martie 2019

Curs 5

Prop: Fie $G=(N,\Sigma,S,P)$ o gram. $LL(k)$ tare, $k \geq 1$ dat; M tabela de analiză sintactică k-predictivă și alg. k-predictiv bazat pe M. Atunci:

$$(w\$, S\$, \lambda) \vdash^{*} (\$, \$, \pi) \vdash accept \Leftrightarrow S \overset{\pi}{\underset{s}{\Rightarrow}} w$$

(alg. k-predictiv pt. gram $LL(k)$ tare G este valid pt. G)

Dem: Fie $G' = (N \cup \{S'\}, \Sigma \cup \{\$\}, S', P \cup \{S \to S\$\})$ extensia lui G.

i. Arătăm că dacă $(xy\$, S\$, \lambda) \vdash^{n} (y\$, \alpha\$, \pi)$ atunci $S \overset{\pi}{\underset{s}{\Rightarrow}} x\alpha$

$n=1$. Rez. că $x=\lambda$, $\pi = i$, $i: S \to \alpha$. Evident $S \overset{\pi}{\underset{s}{\Rightarrow}} x\alpha$

$n \to n+1$. Avem: $(xy\$, S\$, \lambda) \vdash^{n} (x'y\$, \alpha'\$, \pi')$
$\vdash (y\$, \alpha\$, \pi)$

Avem 2 cazuri:

a) $x' = a \in \Sigma$, $\alpha' = a\alpha$, $\pi = \pi'$.

Rez. că $x = x''a$ și $(x''ay\$, S\$, \lambda) \vdash^{n} (ay\$, a\alpha\$, \pi)$

Cf. cu ip. de inducție:

$S \overset{\pi'}{\underset{s}{\Rightarrow}} x''a\alpha$, adică $S \overset{\pi}{\underset{s}{\Rightarrow}} x\alpha$

b) $x' = \lambda$; $\alpha' = A\beta$, $\pi = \pi' i$; $i: A \to \gamma$, $\alpha = \gamma\beta$.

Avem $(xy\$, S\$, \lambda) \vdash^{n} (y\$, A\beta\$, \pi')$

Cf. ip. de inducție:

$S \overset{\pi'}{\underset{s}{\Rightarrow}} xA\beta \overset{i}{\underset{s}{\Rightarrow}} x\gamma\beta = x\alpha$, deci $S \overset{\pi}{\underset{s}{\Rightarrow}} x\alpha$

Dacă $(w\$, S\$, \lambda) \vdash^{*} (\$, \$, \pi)$ at. cf. i pt. $x=\lambda$, $\alpha = \lambda$, rez. că $S \overset{\pi}{\underset{s}{\Rightarrow}} w$

ii. Fie $S \overset{\pi}{\underset{s}{\Rightarrow}} xA\alpha$ și $First'_k(\beta) \subseteq First'_k(A\alpha)$, $\beta \in (N \cup \Sigma)^*$, $x \in \Sigma^*$. Atunci:

$(x\beta\$, S\$, \lambda) \vdash^{*} (\beta\$, A\alpha\$, \pi)$

Inducție după $n = |\pi|$.

Deoarece $First'_k(\beta) \le First'_k(A\alpha)$ rez. că $First'_k(x\beta) \le$ $First'_k(xA\alpha)$

$First'_k(x\beta\$) \le First'_k(xA\alpha\$) \le First'_k(xA\alpha\, Follow'_k(S))$

rez. că: $(x\beta\$, S\$, \lambda) \vdash (x\beta\$, xA\alpha\$, i) \vdash^{*} (\beta\$, A\alpha\$, \pi)$

$n \to n+1$: Fie $S \overset{\pi'}{\underset{s}{\Rightarrow}} x_1 A_1 \alpha_1 \overset{j}{\underset{s}{\Rightarrow}} x_1 x_2 A \alpha_2 \alpha_1$

unde $\pi = \pi' j$, $j: A_1 \to x_2 A \alpha_2$, $x_1 x_2 = x$, $\alpha_2 \alpha_1 = \alpha$, $|\pi'| = n$. Fie β a.î. $First'_k(\beta) \le First'_k(A\alpha)$.

Atunci $First'_k(x_2\beta) \le First'_k(x_2 A \alpha_2 \alpha_1) \le$
$\le First'_k(A_1\alpha_1)$.

Cf. ip. de inducție: $(x_1 x_2 \beta\$, S\$, \lambda) \vdash (x_2\beta\$, A_1\alpha_1\$, \pi')$

Deoarece $j: A_1 \to x_2 A \alpha_2$ și $First'_k(x_2\beta\$) \le First'_k(x_2 A \alpha_2\, Follow'_k(A_1))$, $(x_2\beta\$, A_1\alpha_1\$, \pi') \vdash (x_2\beta\$, x_2 A \alpha_2 \alpha_1\$, \pi' j) \vdash^{|x_2|} (\beta\$, A\alpha\$, \pi)$

Fie $w \in L(G)$, $S \overset{*}{\underset{s}{\Rightarrow}} w$. Punem în evidență ultimul pas al deriv.: $S \overset{\pi}{\underset{s}{\Rightarrow}} xA\alpha \overset{i}{\Rightarrow} x\gamma\alpha = w$,

Fie $i: A \to \gamma \in P$, $\gamma \in \Sigma^*$

Cf. $\overline{II}$ pt. $\beta = \lambda$: $(x\gamma\alpha\$, S\$, \lambda) \vdash^{*} (\gamma\alpha\$, A\alpha\$, \pi) \vdash (\gamma\alpha\$, \gamma\alpha\$, \pi i) \vdash^{|\gamma\alpha|} (\$, \$, \pi i) \vdash$ accept

ALGORITM RECURSIV DESCENDENT

Fie $G = (N, \Sigma, S, P)$ gr. LL(1)

Pp. că funcția scan() care returnează tokenul curent.

- pt. fiecare neterminal se implementează o fct.:

```
main() {
    token = scan();
    S();
    if (token != eof)   error("eof expected");
```

- pt. fiecare neterminal $A \in N$:

```
void A() {
    if (token ∈ First(α1 Follow(A))) {
```

```
        printf("A → α1");
        parse(α1);
}
elseif (token ∈ First(α2 Follow(A))) {
        printf("A → α2");
        parse(α2);
}
  ⋮
elseif (token ∈ First(αn Follow(A))) {
        printf("A → αn");
        parse(αn);
}
else error("se aşteaptă un token diferit");
}
```

- parse(Aα) = check(Aα) = { A(); check(α); }

- parse(aα) = { token = scan(); check(α); }, $a \in \Sigma$

- check(aα) = { if (a == token) token = scan(); else error("a expected"); check(α); }

~~check(α)~~ ≠

ALGORITMUL EARLEY îmbunătăţit

1968. Earley, pt. gram. fără λ-producţii

2002. T. AYCOCK, HORSPOOL

Algoritm pt. orice gram. indep de context $O(n^3)$

- pt. gram. deterministe: $O(n^2)$
- pt. gram. LR(k): $O(n^2)$

Fie $G = (N, \Sigma, S, P)$ gic.

$w \in \Sigma^*$, $w = a_1 a_2 \dots a_n$ şirul analizat

Se construiesc mulţimile $S_0, S_1, \dots, S_n$ care conţin configuraţii de forma:

$A \to \alpha \cdot \beta, j$ $\quad A \to \alpha\beta \in P$

• marchează cât din șirul $\alpha\beta$ a fost analizat $j \in \{0, 1, \dots, n\}$

O configurație se adaugă la o mulțime $S_i \iff \nexists$ în S_i.

G se extinde la $G' = (S \cup \{S'\}, \Sigma, S', P \cup \{S' \to S\})$

Configurația inițială este $S' \to .S, 0$ care este inclusă în S_0

<u>Operații</u>:

- <u>Scanare</u>: dacă $[A \to \alpha . a\beta, j] \in S_i$ și dacă $a = a_{i+1}, i \le n-1$ atunci adaugă $[A \to \alpha a . \beta, j]$ la S_{i+1}.
- <u>Predicție</u>: dacă $A \to \alpha . B\beta, j \in S_i$ atunci pt. $\forall B \to \gamma \in P$ adaugă $B \to . \gamma, i$ la S_i

  În plus, dacă $B \overset{*}{\Rightarrow} \lambda$, atunci adaugă $A \to \alpha B . \beta, j$ la S_i
- <u>Completare</u>: dacă $A \to \alpha ., j \in S_i$ și dacă există $B \to \beta . A\gamma, k \in S_j$, atunci adaugă $B \to \beta A . \gamma, k$ la S_i

<u>Prop</u>: $w \in L(G) \iff S' \to S., 0 \in S_n$

<u>Ex</u>:
$E' \to E$
$E \to E + E$
$E \to n$
$w = n + n$

S_0:
$E' \to . E, 0$
$E \to . E + E, 0$
$E \to . n, 0$

S_1, $w_1 = 'n'$:
$E \to n., 0$
$E' \to E., 0$
$E \to E. + E, 0$

S_2 $\quad w_2 = '+'$

$E \to E+.E,\ 0$

$E \to .E+E,\ 2$

$E \to .n,\ 2$

S_3 $\quad w_3 = 'n'$

$E \to n.,\ 2$

$E \to E+E.,\ 0$

$E \to E.+E,\ 2$

$E' \to E.,\ 0$

$E \to E.+E,\ 0$

METODE DE ANALIZĂ BOTTOM-UP

$S \to aAB$

$A \to b \mid Ab$

$B \to bba$

$w = abbba$

d - deplasare
r - reducere

$abbba \xrightarrow{d} a|bbba \xrightarrow{d} ab|bba \xrightarrow{r} aA|bba \xrightarrow{d} aAb|ba \xrightarrow{r} aAA|ba \xrightarrow{d} aAAb|a \xrightarrow{r} aAAA|a \xrightarrow{d} aAAAa|$

$\xrightarrow{r} aAA|a \ \dots \ S$

19. Martie 2019

Seminar 3

[redacted] Fie $G = (N, \Sigma, S, P)$ g.i.c. Să se construiască un translator stivă T_G. Producțiile numerotate $1, \dots, |P|$.

$$\tau(T_G) = \{(w, \pi) \mid w \in L(G),\ S \xRightarrow[s]{\pi} w\}$$

$$T_G = (\{q\}, \Sigma, \{1, \dots, |P|\}, N \cup \Sigma, \delta, q, S, \emptyset)$$

↳ lucrăm cu vidarea stivei, nu cu stări fin.

$(q, w, S, \lambda) \vdash$

$\delta(q, \lambda, A) = \{(q, \alpha, i) \mid i: A \to \alpha\}$

$\delta(q, a, a) = \{(q, \lambda, \lambda)\},\ \forall\, a \in \Sigma$

EX 1) $G: S \to \overset{1}{aSbS} \mid \overset{2}{bSaS} \mid \overset{3}{\lambda}$

$L(G) = \{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$