LABORATOR NR 6

Analiza sintactică de tip precedență

Gramatici de precedență simplă

a) Dacă gramatica G este LR(k), determinarea capătului din dreapta al părții reductibile se face cu ajutorul unei funcții asupra următoarelor *k* simboluri din porțiunea neexplorată a cuvântului de intrare. Adică, dacă este dată derivarea $S \overset{*}{\Rightarrow} \alpha A w \Rightarrow \alpha\beta w$, atunci determinarea lui β se face cu ajutorul unei funcții asupra cuvântului $PRIM_k(\beta)$. Conceptul de gramatică de precedență este legat de existența unor relații (numite de precedență) între simbolurile lui $N \cup \Sigma$, care permit izolarea părții reductibile.

Dacă $S \underset{d}{\overset{*}{\Rightarrow}} \alpha A w \underset{d}{\Rightarrow} \alpha\beta w = \alpha\beta' yzw'$ cu $\beta = \beta' y$ şi $w = zw'$, atunci izolarea capătului din dreapta a lui β rezultă din:

1) toate simbolurile consecutive ale lui β satisfac o anumită relație
2) ultimul simbol al lui β şi primul simbol al lui *w,* adică perechea (*y,z*), satisfac o relație.

Deci izolarea părții reductibile depinde numai de relația în care se găsesc două simboluri succesive. Relațiile, trei la număr, sunt notate de obicei cu $< \cdot, \dot{=}, \cdot >$. În cazul precedenței simple, întro derivare dreaptă de forma: $S \underset{d}{\overset{*}{\Rightarrow}} \alpha X_{k+1} A a x \underset{d}{\Rightarrow} \alpha X_{k+1} X_k ... X_1 a x$ porțiunea $X_k ..... X_2 X_1$ este delimitată prin cele trei relații astfel:

- la dreapta: $X_1 \cdot > a_1$
- la stânga: $X_{k+1} < \cdot X_k$

- în interior: $X_{i+1} = X_i$ pentru $i = 1,....., k-1$.

De asemenea, între simbolurile consecutive din αX_{k+1}, avem relația $<\cdot$ *sau* $\doteq$. Facem observația că toate derivările sunt la dreapta. Reducerea se face după schema: se parcurge forma derivațională de la stânga la dreapta până când întâlnim pentru prima dată relația $\cdot>$, apoi se parcurge secvența înapoi până când se întâlneşte pentru prima dată relația $<\cdot$. Porțiunea dintre $<\cdot$ şi $\cdot>$ este partea reductibilă. Dacă gramatica este unic invertibilă (adică $(A \to \beta) \in P$ şi $(B \to \beta) \in P$ implică A=B) atunci reducerea este unic determinată şi procesul se repetă până când se obține, prin reducere, S sau până când nu mai este posibilă nici o reducere.

Definiție. Fiind dată o GIC $G = (N, \Sigma, P, S)$ se numesc relații de precedență Wirth-Weber relațiile $<\cdot, \doteq, \cdot>$ definite peste $X, Y \in N \cup \Sigma$ şi $a \in \Sigma$ astfel:

(1) $X <\cdot Y$ dacă există producția $C \to \beta_1 X B \beta_2$ şi $B \overset{+}{\Rightarrow} Y\alpha$

(2) $X \doteq Y$ dacă există producția $C \to \beta_1 XY \beta_2$

(3) $X \cdot> a$ dacă există producția $C \to \beta_1 AB \beta_2$ şi $A \overset{+}{\Rightarrow} \alpha X, B \overset{+}{\Rightarrow} a\beta$

La acestea se adaugă relațiile ce implică marcajul \$, de început şi sfârşit al cuvântului de intrare:

(4) $\$ <\cdot X$ dacă $S \overset{+}{\Rightarrow} X\alpha$

(5) $X \cdot> \$$ dacă $S \overset{+}{\Rightarrow} \alpha X$.

Definiție. O GIC $G = (N, \Sigma, P, S)$ este proprie dacă:

(1) nu are simboluri inutile

(2) nu există cicluri, adică derivări de forma $A \overset{+}{\Rightarrow} A$

(3) nu are ε-producții în afara, eventual, a producției $S \to \varepsilon$ în care caz S nu apare în membrul drept al nici unei producții.

Definiție. O GIC proprie, fără ε-producții, în care între oricare două simboluri există cel mult o relație de precedență Wirth-Weber se numeşte gramatică de *precedență*. Dacă, în plus, gramatica este şi unic invertibilă, ea se numeşte de *precedență simplă*.

Lemă. *Fie* $G = (N, \Sigma, P, S)$ *o gramatică proprie fără* ε*-producții .*

(1) Dacă $X <\cdot Y \ sau \ X \doteq Y \ şi \ (Y \to A\alpha) \in P \quad atunci \ X <\cdot A$.

(2) Dacă $X <\cdot a \ sau \ X \doteq a \ sau \ X \doteq a \quad şi \ (X \to \alpha Y) \in P \quad atunci \ Y <\cdot A$.

Teoremă. Fie $G = (N, \Sigma, P, S)$ *o gramatică proprie fără* ε*-producții şi derivarea:*

$$\$S\$ \overset{*}{\Rightarrow} X_p X_{p-1} \dots X_{k+1} A a_1 \dots . a_q \Rightarrow X_p X_{p-1} \dots X_{k+1} X_k \dots . X_1 a_1 \dots . a_q$$

Atunci:

(1) $pentru \ k < i < p \ avem \ X_{i+1} <\cdot X_i \ sau \ X_{i+1} \doteq X_i$

(2) $X_{k+1} <\cdot X_k$

(3) $pentru \ 1 \le i \le k-1, \ X_{i+1} \doteq X_i$

(4) $X_1 \cdot> a_1$.

b) Analiza sintactică a gramaticilor de precedență simplă

Plecând de la relațiile de precedență trebuie stabilit mai întâi când are loc o trecere a unui simbol de pe banda de intrare pe banda pushdown şi când are loc o operație de reducere.

Algoritm

Intrare : o gramatică de precedență simplă $G = (N, \Sigma, P, S)$ cu producțiile numerotate de la 1 la p.

Ieşire : funcțiile f şi g.

Metoda : Simbolul $\$ \notin (N \cup \Sigma)$ marchează sfârşitul benzii pushdown şi ultimul simbol de pe banda de intrare. Funcția f depinde de simbolul din vârful benzii pushdown şi de cel curent de pe banda de intrare, ea este definită astfel:

a) $f(X,a) = trecere$, dacă $X <\cdot a$ sau $X \doteq a$

b) $f(X,a) = reducere$, dacă $X \cdot> a$

c) $f(S,\$) = acceptare$

d) $f(X,a) = eroare$ în alte cazuri.

Regula c) este prioritară față de regulile a) şi b) când X=S şi a=$.

Funcția *g* nu depinde de conținutul benzii de intrare; ea depinde numai de simbolurile din vârful benzii pushdown ce formează partea reductibilă, plus un sibbol la stânga:

$a')$ $g(X_{k+1}X_k....X_1, \varepsilon) = i$ dacă $X_{k+1} <\cdot X_k$, $X_{j+1} \doteq X_j$ pentru $1 \le j < k$ şi $A \to X_k...X_1$ este producția cu numărul i .

$b')$ $g(\alpha, \varepsilon) = eroare$ în alte cazuri.

Deci f este definită pe $(N \cup \Sigma \cup \{\$\}) \times (\Sigma \cup \{\$\})$ iar g este definită pe $(N \cup \Sigma \cup \{\$\})^*$.

Algoritmul de analiză foloseşte o bandă pushdown pe care $ arată capătul din dreapta şi o bandă de ieşire. Deci, o configurație este de forma: $(\$X_1...X_m, a_1...a_q\$, i_1...i_n)$ unde:

- $\$X_1...X_m$ reprezintă conținutul benzii pushdown cu X_m în vârf;
- $a_1...a_q$ este porțiunea din banda de intrare rămasă de citit, iar a_1 este simbolul curent;

-$i_1 ... i_n$ indică secvența regulilor de producție utilizate pentru a reduce cuvântul inițial la $X_1 ... X_m a_1 ... a_q$.

Pentru ca legătura dintre transformarea configurațiilor şi funcțiile *f* şi *g* să fie mau clară, vom considera că fincțiile *f* şi *g* sunt extinse astfel:

$f : V^* \times (\Sigma \cup \{\$\})^* \rightarrow \{trecere, reducere, eroare, acceptare\}$

$g : V^* \times (\Sigma \cup \{\$\})^* \rightarrow \{1, 2, ..., p, eroare\}$,

unde $V = N \cup \Sigma \cup \{\$\}$.

Trecerea de la o configurație la alta se defineşte astfel:

(1) dacă $f(\alpha, aw) = trecere$ atunci $(\alpha, aw, \pi) | - (\alpha a, w, \pi)$;

(2) dacă $f(\alpha, w) = reducere,\quad g(\alpha, w) = i,\quad A \rightarrow \beta$ este producția cu numarul *i* atunci $(\alpha, w, \pi) | - (\gamma A, w, \pi i), \alpha = \gamma\beta$;

(3) dacă $f(\alpha, aw) = acceptare$ atunci $(\alpha, w, \pi) | - acceptare$

(4) $(\alpha, w, \pi) | - eroare$ în alte cazuri.

Dacă $(\$, w\$, \varepsilon) \overset{*}{|-} (\$S, \$, \pi) | - acceptare$, atunci $S \underset{d}{\overset{\pi}{\Rightarrow}} w$.

Algoritm.

Intrare : matricea de precedență asociată gramaticii G şi $w = x_1 ... x_n$;

Ieşirea : analiza sintactică la dreapta a lui *w* dacă $w \in L(G)$ şi "eroare" în caz contrar;

Metoda : algoritmul utilizează o bandă pushdown *bp* al cărei vârf este indicat de *vârf,* o bandă de intrare *bi* pe care se depune simbolul curent din *w* şi o bandă de ieşire *be*, pe care se depune rezultatul analizei sintactice;

$k \leftarrow 1$

$vârf \leftarrow 1$

$bp(vârf) \leftarrow '\$'$

$bi \leftarrow x_k$

```
ie ← 0
ind ← true
while ind do
begin
 if not bp(vârf) ·> bi
    then
    begin
     vârf ← vârf + 1
      bp(vârf) ← bi
       k ← k + 1
        bi ← x_k
     end
     else
     begin
          if vârf=2 and bp(vârf)='$' and  bi='$'
          then
          begin
           afişează conținutul benzii be
           ind  ← false
          end
 i ← vârf
 while bp(i-1) ≐ bp(i) do
    i ← i-1
găseşte producția p a cărei parte dreaptă este
          be(i)be(i+1)….be(vârf)
if o asemenea producție nu există
```

```
        then
        begin
          write('w ∈ L(G)')
          ind ← false
        end
        else
        begin
           vârf ← i
           bp(vârf) ← partea stângă a producției p
           ie ← ie + 1
           be(ie) ← p
        end
   end
end
```

Gramatici de (m,n)-precedență

Considerăm în loc de perechi de simboluri (X,Y) între care se definesc relațiile $<\cdot, \doteq$ şi $\cdot>$, perechi de simboluri (α, β), cu $|\alpha| = m, |\beta| = n,\ m, n \in N$.

Definiție. Fie $G = (N, \Sigma, P, S)$ o gramatică proprie, fără ε-producții. Definim relațiile de (*m,n*)-precedență pe $(N \cup \Sigma \cup \{\$\})^m \times (N \cup \Sigma \cup \{\$\})^n$ astfel:

fie $\$^m S \$^n \overset{*}{\Rightarrow} X_p X_{p-1} ... X_{k+1} A a_1 ... a_q \Rightarrow X_p X_{p-1} ... X_{k+1} X_k ... X_1 a_1 ... a_q$ un şir de derivații la dreapta; atunci:

(1) $\alpha <\cdot \beta$ unde α constă din ultimele *m* simboluri ale lui $X_p X_{p-1} ... X_{k+1}$ şi

(a) fie β constă din primele *n* simboluri ale lui $X_k ... X_1 a_1 .... a_q$

(b) fie $X_k \in \Sigma$ şi $\beta \in PRIM_n(X_k ... X_1 a_1 ... a_q)$

(2) $\alpha = \beta$ pentru $1 \le j < k$, unde α constă din ultimele m simboluri ale lui $X_p X_{p-1} ... X_{j+1}$ şi

(a) fie β constă din primele n simboluri ale lui $X_j X_{j-1} ... X_1 a_1 .... a_q$

(b) fie $X_j \in \Sigma$ şi $\beta \in PTIM_n\left(X_j X_{j-1} ... X_1 a_1 ... a_q\right)$

(c) $X_m X_{m-1} ... X_1 \cdot > a_1 .... a_n$.

Algoritm –construieşte relațiile de (m,n)-precedență

Intrare : gramatica proprie $G = (N, \Sigma, P, S)$ fără ε-producții

Ieşire : relațiile de (m,n)-precedență

Metoda : Se construieşte mai întâi mulțimea S a subşirurilor de lungime $m+n$ care apar în cuvântul $\alpha\beta u$ astfel încât $\$^m S \$^n \overset{*}{\Rightarrow} \alpha A w \Rightarrow \alpha\beta w$ şi $u = PRIM_n(w)$. Paşii sunt următorii:

<u>Pasul 1.</u> Fie S $= \left\{\$^m S \$^{n-1}, \$^{m-1} S \$^n\right\}$; cele două elemente din S se consideră nemarcate.

<u>Pasul 2.</u> Dacă δ este element nemarcat din S , el se marchează după efectuarea operațiilor:

(a) dacă δ nu se scrie sub forma $\alpha A x$ *cu* $|x| \le n, x \in (\Sigma \cup \{\$\})^*$ atunci nu se efectuează nimic;

(b) dacă $\delta = \alpha A x, |x| \le n, A \in N, x \in (\Sigma \cup \{\$\})^*$ se adaugă la S toate şirurile γ cu proprietatea că există $(A \to \beta) \in P$ şi γ este un subşir de lungime $m+n$ al lui $\alpha\beta x$. Gramatica G fiind proprie, avem $|\alpha\beta x| \ge m+n$. Şirurile adăugate la S se consideră nemarcate.

<u>Pasul 3.</u> Repetă pasul (2) până când toate şirurile din S sunt marcate. Plecând de la S se construiesc relațiile de (m,n)-precedență:

Pasul 4. Pentru fiecare şir $\alpha A\gamma \in$ S astfel încât $|\alpha| = m$ şi pentru fiecare producție $A \to \beta$ luăm $\alpha <\cdot \delta$, unde δ constă din primele *n* simboluri din $\beta\gamma$ sau $\gamma \in PRIM_n(\beta\gamma)$ dacă β începe cu un terminal.

Pasul 5. Pentru fiecare şir $\alpha A\gamma \in$ S astfel încât $|\alpha| = m$ şi pentru fiecare producție $A \to \beta_1 XY\beta_2$ luăm $\delta_1 \doteq \delta_2$, unde δ_1 constă din ultimele *n* simboluri ale $\alpha\beta_1 X$ iar δ_2 din primele *n* simboluri ale lui $Y\beta_2\gamma$ sau $\delta_2 = Yw$ dacă $Y \in \Sigma$ şi $w \in PRIM_{n-1}(\beta_2\gamma)$;

Pasul 6. Pentru fiecare şir $\alpha Aw \in$ S astfel încât $|w| = n$ şi pentru fiecare producție $A \to \beta$ luăm $\delta \cdot> w$, unde δ constă din ultimele *m* simboluri ale lui $\alpha\beta$.

Gramatici de precedență slabă

Condiția ca relațiile de precedență definite pentru o GIC să fie disjuncte este destul de tare. Dacă se impune numai condiția de disjuncție care asigură izolarea corectă a capătului din dreapta al părții reductibile, adică $\cdot> \cap \left(<\cdot \cup \doteq \right) = \varnothing$, se obține o gramatică de precedență slabă.

Definiție. O gramatică proprie, fără ε-producții, G, se numeşte de *precedență slabă* dacă:

(1) $\cdot> \cap \left(<\cdot \cup \doteq \right) = \varnothing$

(2) dacă $A \to \alpha X\beta$ şi $B \to \beta$ sunt producții ale lui G, atunci nici una din relațiile $X <\cdot B$, $X \doteq B$ nu este validă.

Teoremă. Fie $G=(N,\Sigma,P,S)$ *o gramatică de precedență slabă,* $(B\rightarrow\beta)\in P$ *şi* $\$S\$\overset{*}{\Rightarrow}\gamma Cw\Rightarrow\delta X\beta w$. *Dacă* $(A\rightarrow\alpha X\beta)\in P$, *atunci ultima producție folosită în derivarea de mai sus nu este* $B\rightarrow\beta$.

TEME PROPUSE

1. Calculul relațiilor Wirth-Weber
2. Să se verifice dacă o gramatică este de precedență simplă.
3. Implementați algoritmul de analiză sintactică pentru gramatici de precedență simplă.
4. Implementați algoritmul pentru calculul relațiilor de m-n precedență