MI-SP-SP-21 (MI-SYP)

Formální a atributovaný překlad řízený LR analyzátorem.

Formální překlad:

- překladové gramatiky → máme i výstupní symboly
- vstupní a výstupní gramatiky (společně vytvoří překladovou gramatiku)

Postfixová překladová gramatika $PG = (N, T, D, R, S)$

$$A \to \alpha\beta \quad \alpha \in (N \cup T)^*,\ \beta \in D^*$$

→ výstupní symboly jsou vždy na konci pravidla (poslední)

- místo pravého rozkladu můžeme rovnou překládat

druhá možnost → tabulka redukční pravidlo → výstup (substituce)

LR(k) překladové gramatiky

- rozšíření na výstup i při přesunu

$$A \to \alpha x a \beta \quad x \in D^*,\ a \in T,\ \alpha, \beta \in (N \cup T \cup D)^*$$

R-translation gramatika - D jsou na konci nebo vždy těsně před T

+ zároveň neobsahuje překladový konflikt

překladové položky jedné množiny

$$\Rightarrow \begin{array}{l} [A \to \alpha . a\beta,\ \overset{\text{výstup}}{x},\ u]^{\text{výhled}} \\ [B \to \gamma . b\delta,\ y,\ v] \end{array} \quad \text{pokud } x \neq y \text{ a } FIRST_k(a\beta u) \cap FIRST_k(b\delta v) \neq \emptyset$$

- může nastat pouze u přesunu, u redukce jsou pravidla jasná

(to se stane pokud $a = b \in T$ - tj na stejný T mám na výstupu různé D)

Překladové gramatiky s LR(k) vstupní gramatikou

- snažíme se gramatiku transformovat na LR(k) překladovou

- gramatiky jsou ekvivalentní, pokud $Z(PG_1) = Z(PG_2)$ (~~tj $T_1 = T_2$ a $D_1 = D_2$~~)

- jestliže charakteristické gramatiky G_1 a G_2 jsou ekvivalentní a $T_1 = T_2$, $D_1 = D_2$, tak gramatiky $Z(PG_1)$ a $Z(PG_2)$ jsou také ekvivalentní některé transformace:

∘ $A \to \alpha a \otimes \beta \;\Rightarrow\; A \to \alpha \otimes a \beta$

∘ $A \to \alpha \beta \gamma \;\Rightarrow\; A \to \alpha A' \gamma$, $A' \to \beta$

pohlcení zleva

∘ $A \to \alpha \beta \otimes \gamma \;\Rightarrow\; A \to \alpha A' \gamma$, $A' \to \beta \otimes$

pohlcení zprava

∘ $A \to \alpha \otimes \beta \gamma \;\Rightarrow\; A \to \alpha A' \gamma$, $A' \to \otimes \beta$

vložení ε-pravidla

∘ $A \to \alpha \otimes \gamma \;\Rightarrow\; A \to \alpha A' \gamma$, $A' \to \otimes$

- jedná se pouze o pohlcení kde β je prázdná
- vstupní gramatika už nemusí být LR(k)
  (pouze tato transformace)

- pravidla s levou rekurzí jsou problematická (ale pouze pro ~~výstupní gramatiku~~) příklad

$$\begin{aligned} A &\to yAby \\ A &\to ax \end{aligned} \;\Rightarrow\; \begin{aligned} A &\to A'Aby \\ A' &\to y \\ A &\to ax \end{aligned}$$

– tato ale není LR(k) pro žádné k

∘ nekrásání

$$\begin{aligned} A &\to \alpha x B \beta \\ B &\to \delta_1 | \delta_2 | \dots | \delta_n \end{aligned} \;\Rightarrow\; \begin{aligned} A &\to \alpha A' \beta \\ A' &\to x\delta_1 | x\delta_2 | \dots | x\delta_n \end{aligned}$$

- nekrásání ale nebude na předchozí případ také fungovat
  → nekonečně se zacyklí

- nekrásání funguje pouze pokud výstupní gramatika nemá symbol před levě rekurzivním symbolem

- ne všechny gramatiky lze transformovat
  - např. postfixový zápis na prefix nebo infix není možný

LR - atributované gramatiky

- umožňují i zděděné atributy

atributovaná LR(k) položka pro $ATG = (TG, A, V, F)$

$$[A \to \alpha.\beta, x, d, w]$$

pravidlo příl. gramatiky

$x \in D^*$ - výstupní symboly

sémantická pravidla pro počítání zděděných atributů

$w \in T^{*k}$ výhled délky k

sémantický konflikt:

$$[A \to \alpha.\beta, x_1, d_1, w]$$
$$[B \to \gamma.\delta, x_2, d_2, v]$$

máme různá sem. pravidla

pokud $d_1 \neq d_2 \wedge \beta, \delta \in (N \cup T \cup D)^* \cup \{\varepsilon\}$

nebo $x_1 \neq x_2 \wedge$

$$FIRST_k(\beta u) \cap FIRST_k(\delta v) \neq \emptyset$$

nevíme co poslat na výstup (jako překladový konflikt)

→ parsing table polé obsahují navíc i sém. pravidla

S - atributované gramatiky

řešení na atributech položek

- vstupem je postfixová gramatika
- neterminální symboly mají pouze syntetizované atributy
- dědičné nejsou použity v sémantických pravidlech

řešení na předcích a sourozencích vlevo

modifikace LR-parseru:

- vstupní symbol je uložen na zásobník včetně svých syntetizovaných atributů
- při redukci se vypočtou dědičné atributy výstupních symbolů z pravé strany pravidla
  - → i se svými atributy jsou předány na výstup
  - → spočítají se synt. atributy N na levé straně a uloží na zásobník

?

(alternativně: atribut počátečního symb. S je význačný atribut
→ výsledkem je hodnota význačného atributu v kořeni stromu)

příklad:

$E \to \oplus EE+$

$E \to \circledast EE*$

$E \to a\,\textcircled{a}$

$\# = \{E' \to .E,\ E \to .EE+,\ E \to .EE*,\ E \to .a\}$

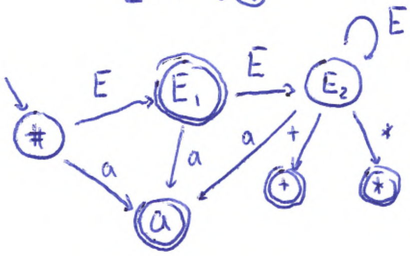
$a = \{E \to a.\}$

$E_1 = \{E' \to E.,\ E \to E.E+,\ E \to E.E*,\ E \to .EE+,\ E \to .EE*,\ E \to .a\}$

$E_2 = \{E \to EE.+,\ E \to EE.*,\ E \to E.E+,\ E \to E.E*,\ E \to .EE+,\ E \to .EE*,\ E \to .a\}$

$+ = \{E \to EE+.\}$

$* = \{E \to EE*.\}$



	a	+	*	ε
#	sh			
a	R(3)	R(3)	R(3)	R(3)
+	R(1)	R(1)	R(1)	R(1)
*	R(2)	R(2)	R(2)	R(2)
E_1	sh			Acc
E_2	sh	sh	sh	

R_1: $E^0 \to E^1E^2+$
$E^0.p = f_3(\oplus, E^1.p, E^2.p)$

R_2: $E^0 \to E^1E^2*$
$E^0.p = f_3(\circledast, E^1.p, E^2.p)$

R_3: $E^0 \to a$
$E^0.p = \textcircled{a}$

$(\#, aa+a*, \varepsilon) \vdash (\#a, a+a*, \varepsilon) \overset{R_3}{\vdash} (\#E_1(a), a+a*, \varepsilon)$

$\vdash (\#E_1(a)a, +a*, \varepsilon) \overset{R_3}{\vdash} (\#E_1(a)E_2(a), +a*, \varepsilon) \vdash (\#E_1(a)E_2(a)+, a*, \varepsilon) \overset{R_1}{\vdash} (\#E_1(+aa), a*, \varepsilon) \vdash$

$\vdash (\#E_1(+aa)a, *, \varepsilon) \overset{R_3}{\vdash} (\#E_1(+aa)E_2(a), *, \varepsilon) \vdash (\#E_1(+aa)E_2(a)*, \varepsilon, \varepsilon) \overset{R_2}{\vdash} (\#E_1(*+aaa), \varepsilon, \varepsilon) \vdash$

$\vdash$ accept

1, $E \to ⊕EE+$ $\quad E^{\circ}.p = f_3(⊕, E_1.p, E_2.p)$ $\qquad$ 0) $E' \to E$

2, $E \to ⊛EE*$ $\quad E^{\circ}.p = f_3(⊛, E_1.p, E_2.p)$

3, $E \to a@$ $\quad E^{\circ}.p = @$

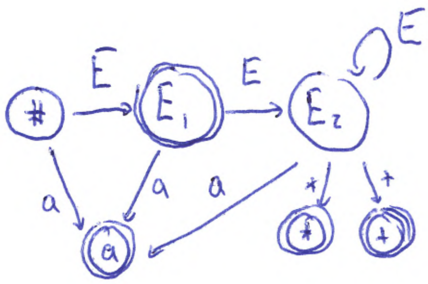
$+ = \{E \to EE+.\}$

$* = \{E \to EE*.\}$

$\# = \{E' \to .E,\ E \to .EE+,\ E \to .EE*,\ E \to .a\}$

$E_1 = \{E' \to E.,\ E \to E.E+,\ E \to E.E*,\ E \to .EE+,\ E \to .EE*,\ E \to .a\}$

$E_2 = \{E \to EE.+,\ E \to EE.*,\ E \to E.E+,\ E \to E.E*,\ E \to .EE+,\ E \to .EE*,\ E \to .a\}$



$a = \{E \to a.\}$

	a	+	*	ε
#	sh			
a	R(3)	R(3)	R(3)	R(3)
E_1	sh			acc
E_2	sh	sh	sh	
+	R(1)	R(1)	R(1)	R(1)
*	R(2)	R(2)	R(2)	R(2)

$(\#, aaa+*, \varepsilon) \vdash (\#a, aa+*, \varepsilon) \vdash$

$\vdash (\#E_1(a), aa+*, \varepsilon) \vdash (\#E_1(a)a, a+*, \varepsilon) \vdash (\#E_1(a)E_2(a), a+*, \varepsilon)$

$\vdash (\#E_1(a)E_2(a)a, +*, \varepsilon) \vdash (\#E_1(a)E_2(a)E_2(a), +*, \varepsilon) \vdash$

$\vdash (\#E_1(a)E_2(a)E_2(a)+, *, \varepsilon) \vdash (\#E_1(a)E_2(+aa), *, \varepsilon) \vdash$

$\vdash (\#E_1(a)E_2(+aa)*, \varepsilon, \varepsilon) \vdash (\#E_1(*a+aa), \varepsilon, \varepsilon) \vdash$ acc