

METODY TRANSLACJI 2004

niektóre przykłady z wykładu

notatka nr 1

Shift-Reduce-Parsing

Shift-Reduce-Parsing dla $id_1 + id_2 * id_3$

Gramatyka:

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

Wyprowadzenie:

$E \Rightarrow \underline{E + E}$
 $\Rightarrow E + \underline{E * E}$
 $\Rightarrow E + E * \underline{id_3}$
 $\Rightarrow E + \underline{id_2} * id_3$
 $\Rightarrow \underline{id_1} + id_2 * id_3$

Parsing

Stos	input	czynność
\$	$id_1 + id_2 * id_3 \$$	shift
$\$id_1$	$+id_2 * id_3 \$$	redukcja przez $E \rightarrow id$
$\$E$	$+id_2 * id_3 \$$	shift
$\$E +$	$id_2 * id_3 \$$	shift
$\$E + id_2$	$*id_3 \$$	redukcja przez $E \rightarrow id$
$\$E + E$	$*id_3 \$$	shift
$\$E + E *$	$id_3 \$$	shift
$\$E + E * id_3$	\$	redukcja przez $E \rightarrow id$
$\$E + E * E$	\$	redukcja przez $E \rightarrow E * E$
$\$E + E$	\$	redukcja przez $E \rightarrow E + E$
$\$E$	\$	<u>Accept</u>

Gramatyki operatorowe

gramatyka nie-operatorowa:

$E \rightarrow EAE \mid id$

$A \rightarrow + \mid * \mid -$

równoważna gramatyka operatorowa:

$E \rightarrow E + E \mid E * E \mid E - E \mid id$

Priorytety dla operacji arytmetycznych

	<i>id</i>	+	*	\$
<i>id</i>		>	>	>
+	<		<	>
*	<	>		>
\$	<	<	<	

Parsing dla gramatyki operatorowej

$\$id + id * id\$$
 $\$ \langle id \rangle + \langle id \rangle * \langle id \rangle \$$
 $\$E + id * id \quad \$$ (ignoruj znaki nieterminalne)
 $\$ + id * id \$$
 $\$ \langle + \langle id \rangle * \langle id \rangle \$$
 $\$ \langle + \langle * \langle id \rangle \$$
 $\$ \langle + \langle * \rangle \$$
 $\$ \langle + \rangle \$$
 $\$\$$

Operator-Precedence

zła gramatyka

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

gramatyka Operator-precedence dla tego samego języka:

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

Obliczenie \langle , \rangle

znaki	pierwszy symbol terminalny, który można wyprowadzić tzn. LEADING	, ostatni symbol terminalny, który można wyprowadzić tzn. TRAILING
<i>E</i>	+ * (<i>id</i>	+ *) <i>id</i>
<i>T</i>	* (<i>id</i>	*) <i>id</i>
<i>F</i>	(<i>id</i>) <i>id</i>

Zasada obliczania LEADING i TRAILING

- i) $a \in \text{LEADING}(A)$, gdy mamy produkcję $A \rightarrow \gamma a \delta$, gdzie $\gamma = \varepsilon$ lub γ jest nieterminalny.
- ii) \exists Produkcja $A \rightarrow B\alpha$ oraz $a \in \text{LEADING}(B) \Rightarrow a \in \text{LEADING}(A)$
- iii) powtarzaj ii) aż nic się nie zmienia

Algorytm

1. $L(A, a) = \text{False} \forall a$
2. zmiana wg i): dla każdego $L(A, a) = \text{True}$ zapisujemy (A, a) na stos.
3. while (stos nie jest pusty)
 - weź (B, a) ze stosu
 - for each produkcja $A \rightarrow B\alpha$ do
 - if $(L(A, a) = \text{False})$ then
 - $L(A, a) = \text{True}$
 - zapisz (A, a) na stosie
 - endif
 - endfor
- endwhile

Algorytm obliczający $\langle \cdot, \dot{=} \cdot \rangle$

- dla każdej produkcji $A \rightarrow x_1 \dots x_k$
for $i = 1$ to k begin
- if $(x_i, x_{i+1}$ są symbolami terminalnymi) then
 - $x_i \dot{=} x_{i+1}$
 - if $(x_i, x_{i+2}$ są symbolami terminalnymi) and $(x_{i+1}$ jest nieterminalny)
 - then

$$x_i \doteq x_{i+2}$$

if (x_i jest terminalny) and (x_{i+1} jest nieterminalny) then

$$\forall a \in \text{LEADING}(x_{i+1}): x_i < a$$

if (x_i jest nieterminalny) then

$$\forall a \in \text{TRAILING}(x_i): a > x_{i+1}$$

end

Uproszczenie

dla każdego terminala a definiuje $f(a), g(a) \in \mathbb{Z}$ tak aby

$$f(a) < g(b) \Leftrightarrow a < b$$

$$f(a) = g(b) \Leftrightarrow a \doteq b$$

$$f(a) > g(b) \Leftrightarrow a > b$$

Dla naszej gramatyki:

	+	*	()	id	\$
f	2	4	0	4	4
g	1	3	5	0	5

na przykład:

$$f(+) > g(+) \text{ , więc } + > +$$

$$f(*) > g(+) \text{ , więc } * > +$$

$$g(*) > f(+) \text{ , więc } + < *$$