



AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE

## Gramatyki z pierwszeństwem operatorów

### Teoria kompilacji

Dr inż. Janusz Majewski  
Katedra Informatyki



## Gramatyka operatorowa

Definicja:

$G = \langle V, \Sigma, P, S \rangle \in \mathcal{G}_{BK}$  jest gramatyką operatorową  $\Leftrightarrow$

- (i)  $G$  – jest gramatyką prawidłową (bez  $\varepsilon$ -produkcji)

$$\forall (A \rightarrow x_1 \dots x_n) \in P:$$

- (ii)  $x_i \in V \Rightarrow x_{i+1} \in \Sigma$   
 $x_{i+1} \in V \Rightarrow x_i \in \Sigma$

(czyli dwa symbole nieterminalne nigdy nie sąsiadują ze sobą w prawych stronach produkcji). W analizie gramatyk operatorowych przyjmuje się termin „operator” dla określenia terminala i termin „operand” dla określenia nieterminala.



## Relacje pierwszeństwa

Dla terminali określa się relacje pierwszeństwa

$\langle \cdot, \equiv, \cdot \rangle \subset (\Sigma \cup \{\$\}) \times (\Sigma \cup \{\$\})$  w następujący sposób:

(1)  $a \equiv b$  gdy  $(A \rightarrow \alpha a \gamma b \beta) \in P \wedge \gamma \in V \cup \{\varepsilon\}$ ,  $a, b \in \Sigma$

(2)  $a \langle \cdot b$  gdy  $(A \rightarrow \alpha a B \beta) \in P \wedge B \xrightarrow{+} \gamma b \delta \wedge \gamma \in V \cup \{\varepsilon\}$

(3)  $a \cdot \rangle b$  gdy  $(A \rightarrow \alpha B b \beta) \in P \wedge B \xrightarrow{+} \delta a \gamma \wedge \gamma \in V \cup \{\varepsilon\}$

(4)  $\$ \langle \cdot a$  gdy  $S \xrightarrow{+} \gamma a \alpha \wedge \gamma \in V \cup \{\varepsilon\}$

(5)  $a \cdot \rangle \$$  gdy  $S \xrightarrow{+} \alpha a \gamma \wedge \gamma \in V \cup \{\varepsilon\}$

We wszystkich powyższych określeniach  $a, b \in \Sigma$ .



## Gramatyka z pierwszeństwem operatorów

### Definicja:

Gramatyka  $G = \langle V, \Sigma, P, S \rangle \in \mathcal{G}_{BK}$  jest gramatyką z pierwszeństwem operatorów  $\Leftrightarrow$

- (i)  $G$  – jest gramatyką operatorową,
- (ii) dla każdej pary „operatorów” (terminali) zachodzi co najwyżej jedna z relacji pierwszeństwa:  $\langle \cdot, \equiv, \cdot \rangle$



## Przykład

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid \underline{id}$$

	+	*	(	)	<u>id</u>	\$
+	•>	<•	<•	•>	<•	•>
*	•>	•>	<•	•>	<•	•>
(	<•	<•	<•	<u>•</u>	<•	
)	•>	•>		•>		•>
<u>id</u>	•>	•>		•>		•>
\$	<•	<•	<•		<•	

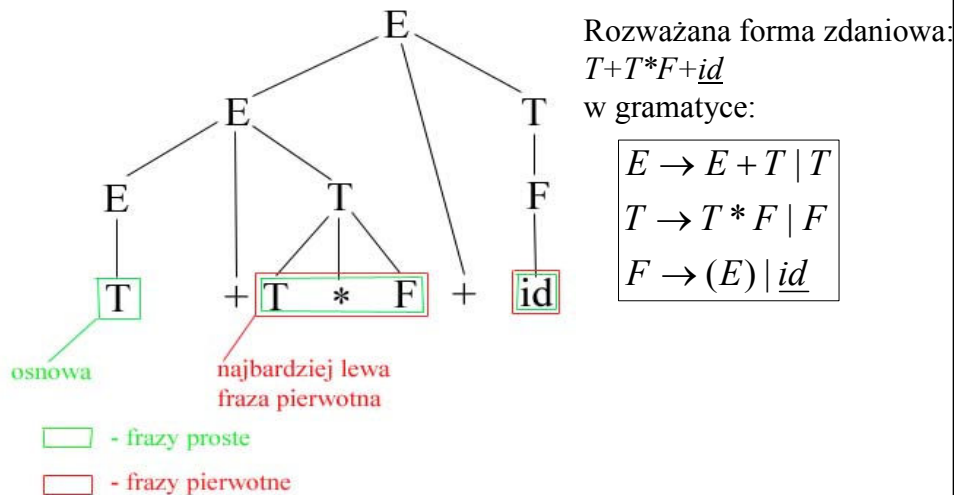


## Fraza pierwotna

Mając daną gramatykę pierwszeństw operatorów i znając relacje pierwszeństw dla terminali tej gramatyki możemy w każdej formie zdaniowej określić frazę pierwotną.

Frazą pierwotną formy zdaniowej nazywamy dowolną frazę zawierającą co najmniej jeden terminal i nie zawierającą żadnej innej frazy pierwotnej (oprócz siebie).

## Przykład



## Relacje pierwszeństw

### Twierdzenie:

Niech  $G = \langle V, \Sigma, P, S \rangle$  - gramatyka pierwszeństw operatorów i

$$\begin{matrix} * \\ \$S\$ \Rightarrow \alpha A \omega \Rightarrow \alpha \beta \omega \\ R \qquad R \end{matrix}$$

Wówczas:

- (1) Relacje  $\prec \bullet$  lub  $\equiv$  spełnione są między kolejnymi terminalami (i symbolem  $\$$ ) łańcucha  $\alpha$
- (2) Relacja  $\prec \bullet$  spełniona jest między skrajnym prawym terminalem łańcucha  $\alpha$  i skrajnym lewym terminalem łańcucha  $\beta$
- (3) Relacja  $\equiv$  spełniona jest między kolejnymi terminalami łańcucha  $\beta$
- (4) Relacja  $\bullet >$  spełniona jest między skrajnym prawym terminalem łańcucha  $\beta$  i pierwszym symbolem (oczywiście też terminalem, bo rozważamy wywód prawostronny) łańcucha  $\omega$



## Przykład

	+	*	(	)	id	\$
+	>	<	<•	>	<	>
*	>	>	<	>	<	>
(	<	<	<	≡	<	
)	>	>		>		>
id	>	>		>		>
\$	<	<	<		<	

Przykład:

$$\underbrace{\$E\$}_{\$S\$} \xRightarrow{*} \underbrace{\$E+T}_{\alpha} \underbrace{+id}_{\omega} \$ \xRightarrow{R} \underbrace{\$E+T}_{\alpha} \underbrace{*F}_{\beta} \underbrace{+id}_{\omega} \$$$

\$ E + T \* F + id \$

\$ <• + <• \* •> + <•\*\* id •>\*\* \$

przy czym:

\*\* - relacje wewnątrz łańcucha  $\omega$ , istotne później

$\beta$  - fraza pierwotna (tutaj także osnowa)



## Gramatyka szkieletowa

Definicja:

Niech  $G = \langle V, \Sigma, P, S \rangle$  - gramatyka operatorowa.

Gramatyką szkieletową  $G_s$  dla gramatyki  $G$  nazywamy gramatykę  $G_s = \langle \{S\}, \Sigma, P', S \rangle$  zawierającą produkcję

typu:  $(S \rightarrow X_1 \dots X_m) \in P'$  dla każdej produkcji:

$(A \rightarrow Y_1 \dots Y_m) \in P$ , przy czym dla  $1 \leq i \leq m$ :

(1)  $X_i = Y_i$  gdy  $Y_i \in \Sigma$

(2)  $X_i = S$  gdy  $Y_i \in V$

oraz  $P'$  nie zawiera produkcji  $S \rightarrow S$ .

$G_s$  - na ogół nie jest jednoznaczna.

$L(G) \subseteq L(G_s)$



## Gramatyka szkieletowa

$G_S$  - na ogół nie jest jednoznaczna.

$$L(G) \subseteq L(G_S)$$

Przy rozbiore w oparciu o pierwszeństwa „operatorów” nie troszczymy się o nieterminale, gdyż są dla nas nierozróżnialne, bo:

- nie znamy relacji pomiędzy nieterminalami
- nie potrafimy wykryć osnowy, gdy nie zawiera ona terminala
- nie umiemy w związku z tym zredukować wg produkcji łańcuchowych



## Gramatyka szkieletowa - przykład

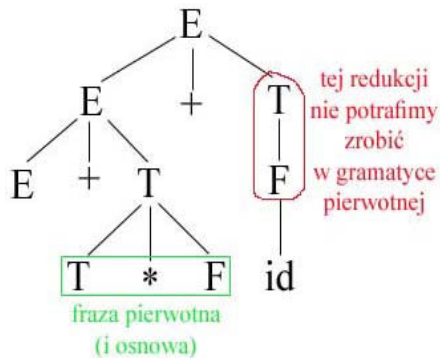
Tworzymy gramatykę szkieletową zastępując wszystkie nieterminale symbolem początkowym  $S$  i eliminując produkcje łańcuchowe.

$$\begin{array}{l} E \rightarrow E + T \mid T \\ T \rightarrow T * F \mid F \\ F \rightarrow (E) \mid \underline{id} \end{array} \quad \begin{array}{l} \text{pierwotna} \\ \Rightarrow \\ \text{szkieletowa} \end{array} \quad \begin{array}{l} E \rightarrow E + E \\ E \rightarrow E * E \\ E \rightarrow (E) \\ E \rightarrow \underline{id} \end{array}$$

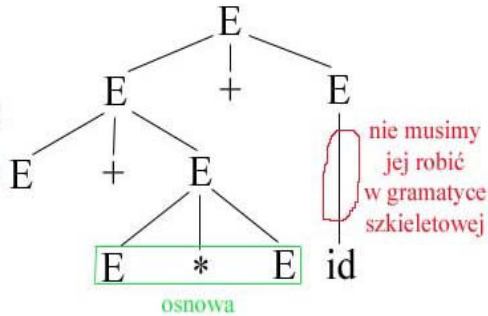


## Gramatyka szkieletowa

Drzewo syntaktyczne  
w gramatyce pierwotnej



Drzewo syntaktyczne  
w gramatyce szkieletowej

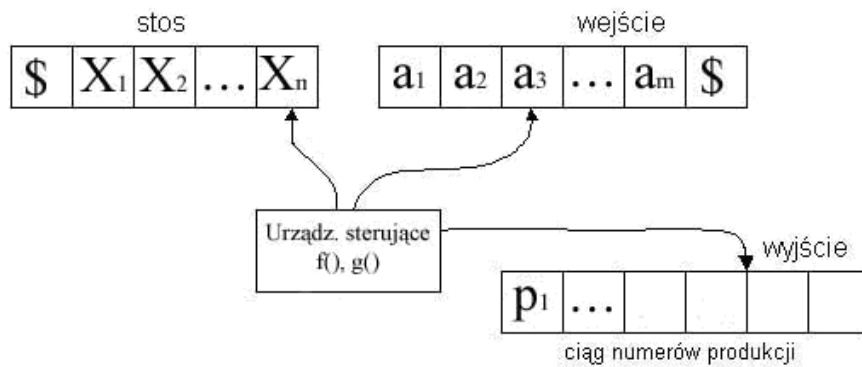


## Parsing szkieletowy

Otrzymana gramatyka szkieletowa jest na ogół niejednoznaczna i generuje słowa nie należące do języka gramatyki pierwotnej. JEDNOZNACZNY jest natomiast algorytm parsingu, który dodatkowo uwzględnia informacje z tablicy relacji pierwszeństw terminali i akceptuje słowa gramatyki pierwotnej. Dążymy do tego, aby algorytm parsingu akceptował słowa gramatyki pierwotnej i tylko te słowa, ale tak być nie musi. Rozbiór polega na odtworzeniu szkieletowego drzewa syntaktycznego poprzez redukcje osnów gramatyki szkieletowej (co sprowadza się do redukcji fraz pierwotnych w gramatyce pierwotnej).



## Algorytm shift-reduce dla pierwszeństw operatorów



## Algorytm shift-reduce dla pierwszeństw operatorów

We:  $G_S = \langle \{S\}, \Sigma, P', S \rangle$  oraz relacje pierwszeństw „operatorów” dla  $G = \langle V, \Sigma, P, S \rangle$

Wy: drzewo rozbioru syntaktycznego dla  $G_S$

$\beta$  – oznacza  $S$  lub  $\varepsilon$

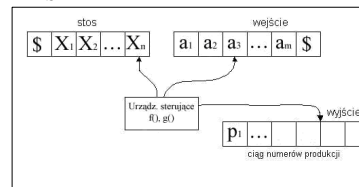
$a, b$  – oznacza terminal lub  $\$$

Działanie parsera:

Działanie parsera opisują funkcje  $f$  i  $g$

Funkcja  $f(\dots)$  jest funkcją określającą działanie parsera.

Funkcja  $g(\dots)$ , wywoływana, gdy  $f(\dots) = \underline{red}$ , jest odpowiedzialna za notowanie numerów produkcji.



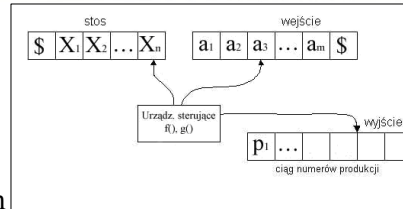




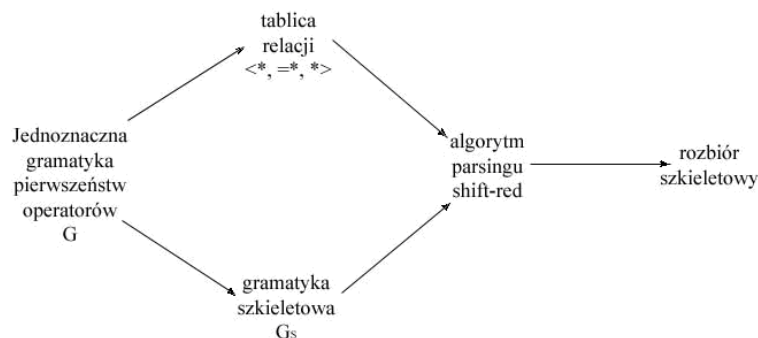
## Algorytm shift-reduce dla pierwszeństw operatorów

Działanie parsera:

- (1)  $f(a\beta, b) = \text{shift}$  gdy  $a < \bullet b$  lub  $a \doteq b$
- (2)  $f(a\beta, b) = \text{red}$  gdy  $a \bullet > b$
- (3)  $f(\$S, \$) = \text{acc}$
- (4)  $f(\alpha, \omega) = \text{err}$  w pozostałych przypadkach
- (5)  $g(a\beta b\gamma, \omega) = i$  gdy:
  - (a)  $\beta$  jest  $S$  lub  $\varepsilon$
  - (b)  $a < \bullet b$
  - (c) relacja  $\doteq$  jest spełniona dla kolejnych terminali łańcucha  $\gamma$
  - (d)  $S \rightarrow \beta b \gamma$  jest produkcją w  $G_S$  numerze  $i$
- (6)  $g(\alpha, \omega) = \text{err}$  w pozostałych przypadkach



## Parsing wykorzystujący pierwszeństwa operatorów



Powyzsza procedura moze byc uproszczona w nastepujacy sposob:

- (1) konstruuje sie tylko gramatyke szkieletowa
- (2) określa sie tablice relacji pierwszeństw „operatorów” na podstawie zakładanych własności języka
- (3) buduje sie na tej podstawie algorytm parsingu



## Przykład

- (1)  $E \rightarrow \neg E$  (minus unarny)
- (2)  $E \rightarrow E \uparrow E$  (potęgowanie)
- (3)  $E \rightarrow E * E$  (mnożenie)
- (4)  $E \rightarrow E / E$  (dzielenie)
- (5)  $E \rightarrow E + E$  (dodawanie)
- (6)  $E \rightarrow E - E$  (odejmowanie)
- (7)  $E \rightarrow (E)$
- (8)  $E \rightarrow id$

Formułujemy własności operatorów:

Operator	Priorytet	Liczba operandów	Usytuowanie	Łączność
$\neg$	1 (najwyższy)	unarny	prefix	prawostronna
$\uparrow$	2	binarny		prawostronna
$*, /$	3	binarny		lewostronna
$+, -$	4	binarny		lewostronna



## Przykład c.d.

Ustalamy relacje między terminalami:

- (1) Jeśli operator binarny  $\Theta_1$  ma wyższy priorytet niż operator  $\Theta_2$ , to:  $\Theta_1 \bullet > \Theta_2$  i  $\Theta_2 < \bullet \Theta_1$

np.:  $* \bullet > +$ ,  $+ < \bullet *$

- (2) Jeśli operator binarny  $\Theta_1$  ma taki sam priorytet jak operator  $\Theta_2$ , to:

(a)  $\Theta_1 \bullet > \Theta_2$  i  $\Theta_2 \bullet > \Theta_1$  dla lewostronnie łącznych

(b)  $\Theta_1 < \bullet \Theta_2$  i  $\Theta_2 < \bullet \Theta_1$  dla prawostronnie łącznych

np.: lewostronnie łączne:  $+ \bullet > -$ ,  $- \bullet > +$ ,  $+ \bullet > +$ ,  $- \bullet > -$

prawostronnie łączne:  $\uparrow < \bullet \uparrow$



## Przykład c.d.

- (3) Jeśli  $\mathcal{G}$  jest unarnym prefixem, to:
- (a)  $\Theta \langle \bullet \mathcal{G}$  dla każdego operatora  $\Theta$  - binarnego lub unarnego
  - (b)  $\mathcal{G} \bullet \rangle \Theta$  gdy  $\mathcal{G}$  ma wyższy priorytet od  $\Theta$
  - (c)  $\mathcal{G} \langle \bullet \Theta$  gdy  $\mathcal{G}$  ma niższy priorytet od  $\Theta$
- np.:  $\neg \langle \bullet \neg, + \langle \bullet \neg, \neg \bullet \rangle +$



## Przykład c.d.

- (4) Pozostałe relacje ( $\Theta$  oznacza tutaj dowolny operator):
- $\Theta \langle \bullet \underline{id}, \Theta \langle \bullet (, ) \bullet \rangle \Theta, \Theta \bullet \rangle \$, \underline{id} \bullet \rangle \Theta, (\langle \bullet \Theta$   
 $\Theta \bullet \rangle), \$ \langle \bullet \Theta, (\overset{\cdot}{=}), (\langle \bullet (, ) \bullet \rangle), \$ \langle \bullet ($   
 $) \bullet \rangle \$, \$ \langle \bullet \underline{id}, \underline{id} \bullet \rangle \$, (\langle \bullet \underline{id}, \underline{id} \bullet \rangle)$



## Przykład c.d.

Tworzymy tabelę relacji:

	+	-	*	/	↑	<u>id</u>	(	)	¬	\$
+	•>	•>	<•	<•	<•	<•	<•	•>	<•	•>
-	•>	•>	<•	<•	<•	<•	<•	•>	<•	•>
*	•>	•>	•>	•>	<•	<•	<•	•>	<•	•>
/	•>	•>	•>	•>	<•	<•	<•	•>	<•	•>
↑	•>	•>	•>	•>	<•	<•	<•	•>	<•	•>
<u>id</u>	•>	•>	•>	•>	•>	err 3	err 3	•>	•>	•>
(	<•	<•	<•	<•	<•	<•	<•	≡	<•	err 4
)	•>	•>	•>	•>	•>	err 3	err 3	•>	•>	•>
¬	•>	•>	•>	•>	•>	<•	<•	•>	<•	•>
\$	<•	<•	<•	<•	<•	<•	<•	err 2	<•	err 1



## Przykład c.d.

Precyzujemy algorytm parsera: ( $\beta$  oznacza  $E$  lub  $\varepsilon$ )

$$f(a\beta, b) = \underline{shift}, \text{ gdy } a < \bullet b \text{ lub } a = \overset{\bullet}{b}$$

$$f(a\beta, b) = \underline{red}, \text{ gdy } a \bullet > b$$

$$f(\$E, \$) = \underline{acc},$$

$$f(\dots, \dots) = \underline{err}, \text{ w pozostałych przypadkach.}$$

$$g(b\beta \neg E) = 1, \text{ gdy } b < \bullet \neg$$

$$g(bE \uparrow E) = 2, \text{ gdy } b < \bullet \uparrow$$

$$g(bE * E) = 3, \text{ gdy } b < \bullet *$$

.....

.....

$$g(b\beta(E)) = 7, \text{ gdy } b < \bullet ($$

$$g(b\beta \underline{id}) = 8, \text{ gdy } b < \bullet \underline{id}$$

$$g(\dots, \dots) = \underline{err}, \text{ w pozostałych przypadkach.}$$



## Przykład c.d.

Przykład rozbioru szkieletowego dokonanego przez parser:

\$	$\underline{id}^* \neg (id+id) \uparrow id\$$	$\varepsilon$	$\mapsto [ \$ < \bullet \underline{id}, \underline{shift} ]$
\$ <u>i</u> d	$* \neg (id+id) \uparrow id\$$	$\varepsilon$	$\mapsto [ \underline{id} \bullet > *, \underline{red-8} ]$
\$E	$* \neg (id+id) \uparrow id\$$	8	$\mapsto [ \$ < \bullet *, \underline{shift} ]$
\$E*	$\neg (id+id) \uparrow id\$$	8	$\mapsto [ * < \bullet \neg, \underline{shift} ]$
\$E* $\neg$	$(id+id) \uparrow id\$$	8	$\mapsto [ \neg < \bullet (, \underline{shift} ]$
\$E* $\neg$ (	$id+id) \uparrow id\$$	8	$\mapsto [ ( < \bullet id, \underline{shift} ]$
\$E* $\neg$ ( <u>i</u> d	$+id) \uparrow id\$$	8	$\mapsto [ \underline{id} \bullet > +, \underline{red-8} ]$
\$E* $\neg$ (E	$+id) \uparrow id\$$	88	$\mapsto [ ( < \bullet +, \underline{shift} ]$
\$E* $\neg$ (E+	$id) \uparrow id\$$	88	$\mapsto [ + < \bullet id, \underline{shift} ]$
\$E* $\neg$ (E+ <u>i</u> d	$) \uparrow id\$$	88	$\mapsto [ \underline{id} \bullet > ), \underline{red-8} ]$



## Przykład c.d.

\$E* $\neg$ (E+E	$) \uparrow id\$$	888	$\mapsto [ + \bullet > ), \underline{red-5} ]$
\$E* $\neg$ (E	$) \uparrow id\$$	8885	$\mapsto [ ( \bullet = ), \underline{shift} ]$
\$E* $\neg$ (E)	$\uparrow id\$$	8885	$\mapsto [ ) \bullet > \uparrow, \underline{red-7} ]$
\$E* $\neg$ E	$\uparrow id\$$	88857	$\mapsto [ \neg \bullet > \uparrow, \underline{red-1} ]$
\$E*E	$\uparrow id\$$	888571	$\mapsto [ * < \bullet \uparrow, \underline{shift} ]$
\$E*E $\uparrow$	$id\$$	888571	$\mapsto [ \uparrow < \bullet id, \underline{shift} ]$
\$E*E $\uparrow$ <u>i</u> d	$\$$	888571	$\mapsto [ \underline{id} \bullet > \$, \underline{red-8} ]$
\$E*E $\uparrow$ E	$\$$	8885718	$\mapsto [ \uparrow \bullet > \$, \underline{red-2} ]$
\$E*E	$\$$	88857182	$\mapsto [ * \bullet > \$, \underline{red-3} ]$
\$E	$\$$	888571823	$\mapsto \underline{acc}$



## Przykład c.d.

(1) Błędy syntaktyczne wykrywane są wówczas, gdy nie jest określone pierwszeństwo terminali (por. tablica relacji pierwszeństw i definicja funkcji  $f(\dots, \dots)$ ):

**err 1:** „missing operand”

Akcja: wprowadzenie „id” na wejście

**err 2:** „unbalanced right parenthesis”

Akcja: zlikwidowanie „)” z wejścia

**err 3:** „missing operator”

Akcja: wprowadzenie „+” na wejście

**err 4:** „missing right parenthesis”

Akcja: zdjęcie „(” ze stosu wraz ze wszystkimi symbolami, które są między „(” a wierzchołkiem stosu z wierzchołkiem stosu włącznie

	id	(	)	\$
id	err 3	err 3	•>	•>
(	<•	<•	≡	err 4
)	err 3	err 3	•>	•>
\$	<•	<•	err 2	err 1



## Przykład c.d.

(2) Błędy syntaktyczne wykrywane wówczas, gdy badanie pierwszeństwa wskazuje na redukcję, a redukcji nie da się dokonać, gdyż zawartość stosu nie odpowiada prawej stronie żadnej produkcji (por. definicja funkcji  $g(\dots, \dots)$ ):

**err 5:** „missing operand(s)”

Redukowany jest operator (+, −, /, \*, ↑, −), a brak jednego lub obu nieterminali  $E$ .

Akcja: redukcja mimo wszystko, operator na stosie wskazuje, wg której produkcji zredukować

**err 6:** „no expression between parenthesis”

Brak nieterminala  $E$  pomiędzy “(” a “)”.

Akcja: redukcja wg produkcji 7 mimo błędu



## Przykład c.d.

### err 7: „missing operator”

Redukowany jest id, po lewej stronie id na stosie nie ma terminala.

Akcja: zdjęcie ze stosu nieterminala  $E$ , poprzedzającego id, po czym redukcja wg produkcji 8

### err 8: „missing operator”

Redukowane są „( $E$ )”, po lewej stronie „(” na stosie jest  $E$ .

Akcja: zdjęcie ze stosu nieterminala  $E$  poprzedzającego „(”, po czym redukcja wg produkcji 7



## Przykład c.d.

\$	+ <u>id</u> )id() <u>id id</u> \$	$\mapsto$ [ $\$ < \bullet +$ , <u>shift</u> ]
\$+	- <u>id</u> )id() <u>id id</u> \$	$\mapsto$ [ $+ \bullet > -$ , <u>red-5</u> , <u>err-5</u> ]

### **ERROR 5:** „missing operands”

\$E	- <u>id</u> )id() <u>id id</u> \$	$\mapsto$ [ $\$ < \bullet -$ , <u>shift</u> ]
\$E-	<u>id</u> )id() <u>id id</u> \$	$\mapsto$ [ $- < \bullet id$ , <u>shift</u> ]
\$E- <u>id</u>	) <u>id</u> () <u>id id</u> \$	$\mapsto$ [ <u>id</u> $\bullet >$ ), <u>red-8</u> ]
\$E-E	) <u>id</u> () <u>id id</u> \$	$\mapsto$ [ $- \bullet >$ ), <u>red-6</u> ]
\$E	) <u>id</u> () <u>id id</u> \$	$\mapsto$ [ $\$ ?$ ), <u>err-2</u> ]

### **ERROR 2:** „unbalanced right parenthesis”

\$E	<u>id</u> ) <u>id id</u> \$	$\mapsto$ [ $\$ < \bullet id$ , <u>shift</u> ]
-----	-----------------------------	--



## Przykład c.d.

\$Eid            ())id id\$             $\mapsto$  [id?(, err-3]

**ERROR 3: „missing operator”**

\$Eid            +())id id\$             $\mapsto$  [id•>+, red-8 err-7]

**ERROR 7: „missing operator”**

\$E                +())id id\$             $\mapsto$  [\$<•+, shift]

\$E+              ())id id\$             $\mapsto$  [+<•(, shift]

\$E+(             ))id id\$             $\mapsto$  [(=), shift]

\$E+()            )id id\$             $\mapsto$  []•>), red-7, err-6]

**ERROR 6: „no expression between parenthesis”**

\$E+E             )id id\$             $\mapsto$  [+•>), red-5]

\$E                )id id\$             $\mapsto$  [\$?), err-2]

**ERROR 2: “unbalanced right parenthesis”**

\$E                id id\$             $\mapsto$  [\$<•id, shift]



## Przykład c.d.

\$Eid            id\$                     $\mapsto$  [id?id, err-3]

**ERROR 3: “missing operator”**

\$Eid            +id\$                    ?

\$Eid            +id\$                     $\mapsto$  [id•>+, red-8, err-7]

**ERROR 7: “missing operator”**

\$E                +id\$                     $\mapsto$  [\$<•+, shift]

\$E+              id\$                       $\mapsto$  [+<•id, shift]

\$E+id            \$                         $\mapsto$  [id•> \$, red-8]

\$E+E            \$                         $\mapsto$  [+•> \$, red-5]

\$E                \$                         $\mapsto$  error, bo były błędy