

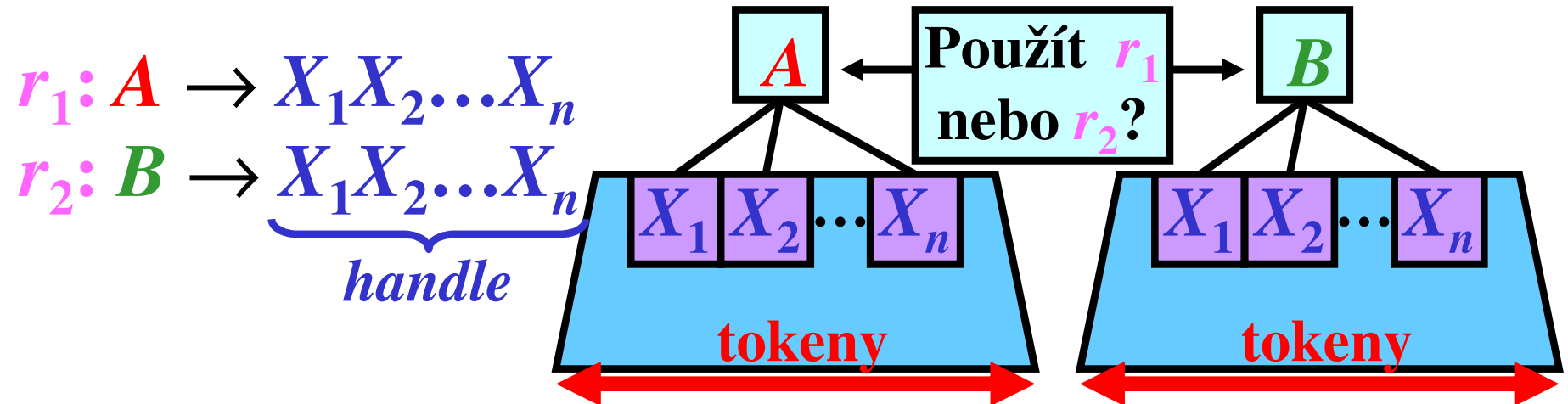
Kapitola VIII.

Syntaktická analýza

zdola nahoru

SA zdola nahoru: Problémy

1) Dvě nebo více pravidel mají stejnou pravou stranu



Pozn.: Pravá strana pravidla je označována slovem „*handle*“

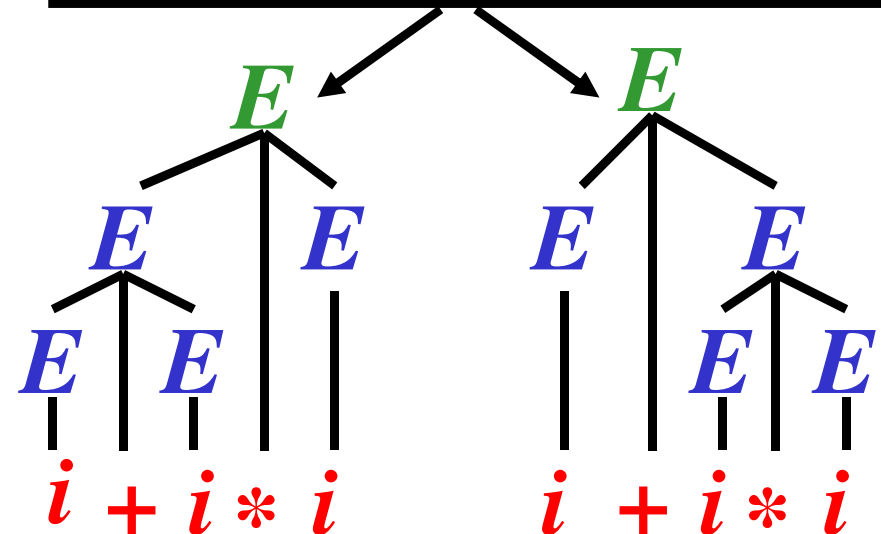
2) Nejednoznačné gramatiky

Který ze stromů vytvořit?

$G_{\text{expr2}} = (N, T, P, E)$, kde

$N = \{E\}$, $T = \{i, +, *, (,)\}$,

$P = \{$
 $\quad 1: E \rightarrow E + E, \quad 2: E \rightarrow E * E,$
 $\quad 3: E \rightarrow (E), \quad 4: E \rightarrow i \quad \}$



Syntaktické analyzátory pracující zdola nahoru

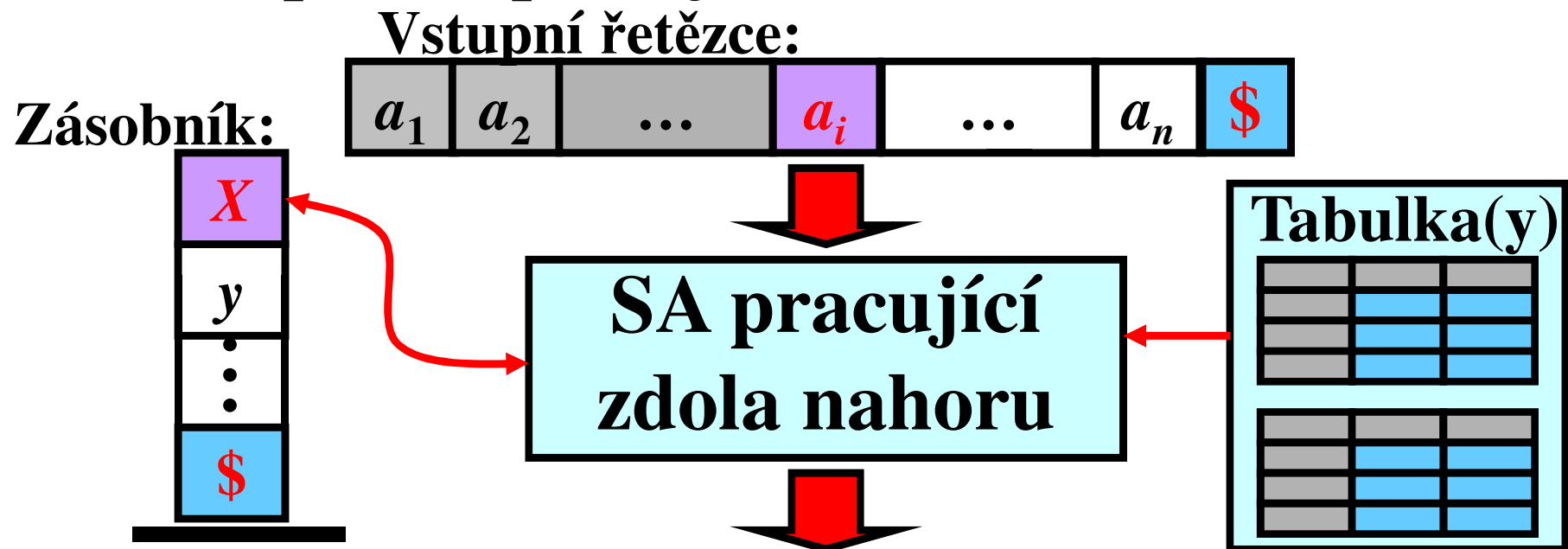
1) Precedenční syntaktický analyzátor

- nejslabší, ale jednoduše se implementuje

2) LR syntaktický analyzátor

- nejsilnější, ale složitý

• Model pro SA pracující zdola nahoru:



Pravý rozbor = **reverzovaná** posloupnost pravidel, která je použita v **nejpravější derivaci** pro vstupní řetězec.

Precedenční SA

- Nesmí existovat více pravidel se stejnou pravou stranou
 - Gramatika nesmí obsahovat ε -pravidla.
-
- Necht' $G = (N, T, P, S)$ je BKG, kde $T = \{a_1, a_2, \dots, a_n\}$

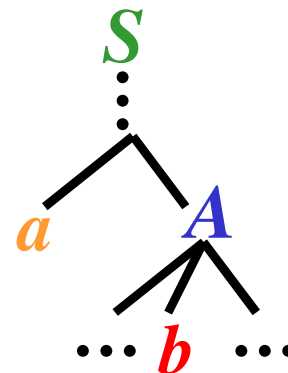
Precedenční tabulka:

	a_1	...	a_j	...	a_n	\$
a_1						
...						
a_i						
...						
a_n						
\$						

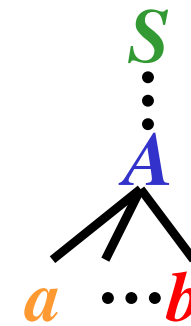
Tabulka[a_i, a_j] $\in \{<, =, >, nic\}$

Ilustrace významu $<, =, >$:

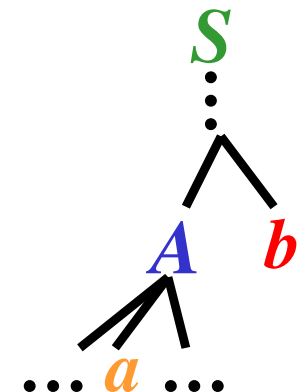
$a < b$



$a = b$



$a > b$



Precedenční SA: Algoritmus

- **Vstup:** Precedenční tabulka pro $G = (N, T, P, S)$; $x \in T^*$
- **Výstup:** Pravý rozbor x , pokud $x \in L(G)$, jinak **chyba**
- **Metoda:**
 - necht' funkce **top** vrací terminál na zásobníku nejbližze vrcholu
 - vlož **\$** na zásobník;
 - repeat
 - necht' $a = \text{top}$
 $b =$ aktuální znak na vstupu,
 - case Tabulka[a, b] of:
 - $=$: push(b) & přečti další symbol b ze vstupu
 - $<$: zaměň a za $a<$ na zásobníku & push(b) & přečti další symbol b ze vstupu
 - $>$: if $<y$ je na vrcholu zásobníku and $r: A \rightarrow y \in P$ then zaměň $<y$ za A & vypiš r na výstup else **chyba**
 - **prázdné políčko** : **chyba**
 - until $b = \$$ and zásobník obsahuje pouze $\$S$
 - **úspěch syntaktické analýzy**

Precedenční SA: Příklad

$G_{expr2} = (N, T, P, E)$, kde $N = \{E\}$, $T = \{i, +, *, (,)\}$,
 $P = \{1: E \rightarrow E+E, 2: E \rightarrow E * E, 3: E \rightarrow (E), 4: E \rightarrow i\}$

Precedenční tabulka pro G_{expr2} :

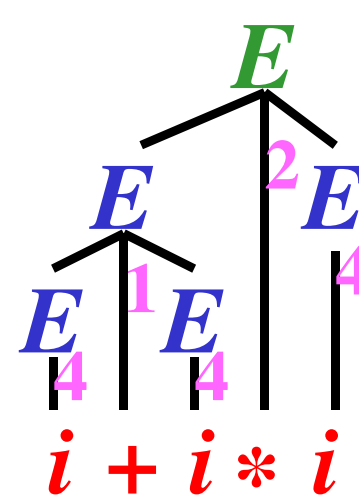
Vstupní token

	+	*	()	i	\$
+	>	<	<	>	<	>
*	>	>	<	>	<	>
(<	<	<	=	<	
)	>	>		>		>
i	>	>		>		>
\$	<	<	<		<	

Terminál na vrcholu zásobníku

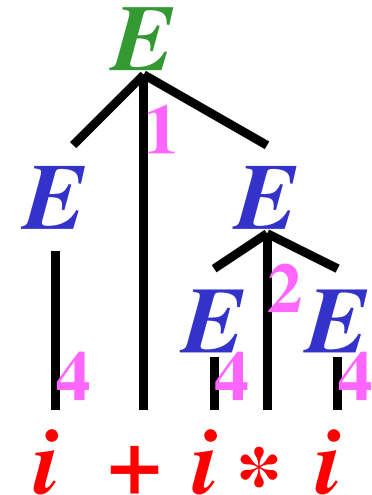
Pozn.: Asociativita a precedence operátorů tvoří základ precedenční tabulky:

☹ Špatný strom: ☺ Správný strom:



Pravý rozbor:

44142



Pravý rozbor:

44421

Precedenční SA: Příklad

	+	*	()	<i>i</i>	\$
+	>	<	<	>	<	>
*	>	>	<	>	<	>
(<	<	<	=	<	
)	>	>		>		>
<i>i</i>	>	>		>		>
\$	<	<	<		<	

Pravidla:

1: $E \rightarrow E + E$

2: $E \rightarrow E * E$

3: $E \rightarrow (E)$

4: $E \rightarrow i$

Vstupní řetězec: $i + i * i \$$

Pushdown	Op	Vstup	Rule
\$	<	$i + i * i \$$	
$\$ < i$	>	$+ i * i \$$	4: $E \rightarrow i$
$\$ E$	<	$+ i * i \$$	
$\$ < E +$	<	$i * i \$$	
$\$ < E + < i$	>	$* i \$$	4: $E \rightarrow i$
$\$ < E + E$	<	$* i \$$	
$\$ < E + < E *$	<	$i \$$	
$\$ < E + < E * < i$	>	$\$$	4: $E \rightarrow i$
$\$ < E + < E * E$	>	$\$$	2: $E \rightarrow E * E$
$\$ < E + E$	>	$\$$	1: $E \rightarrow E + E$
$\$ E$		$\$$	

Úspěch

Pravý rozbor: 44421

Konstrukce precedenční tabulky 1/5

- Necht' $G_{expr} = (N, T, P, E)$, kde $N = \{E\}$,
 $T = \{ (,), id_1, id_2, \dots, id_m, op_1, op_2, \dots, op_n \}$,
 $P = \{ E \rightarrow (E), E \rightarrow id_1, E \rightarrow id_2, \dots, E \rightarrow id_m,$
 $E \rightarrow E op_1 E, E \rightarrow E op_2 E, \dots, E \rightarrow E op_n E \}$

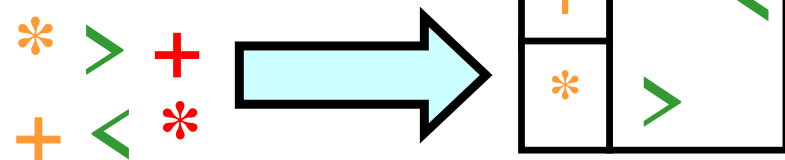
Pozn.: id_1, id_2, \dots, id_m jsou identifikátory,
 op_1, op_2, \dots, op_n jsou rozdílné operátory

1) Precedence operátorů:

- Pokud op_i má vyšší prioritu než op_j , potom:

$$op_i > op_j \text{ a } op_j < op_i$$

Příklad: Precedenční tabulka odvozená z priority operátorů gramatiky G_{expr2} :



Konstrukce precedenční tabulky 2/5

2) Asociativita:

Pozn.:

- op_i je levě asociativní $\Leftrightarrow a \text{ } op_i \text{ } b \text{ } op_i \text{ } c = (a \text{ } op_i \text{ } b) \text{ } op_i \text{ } c$
- op_i je pravě asociativní $\Leftrightarrow a \text{ } op_i \text{ } b \text{ } op_i \text{ } c = a \text{ } op_i (b \text{ } op_i \text{ } c)$

- Necht' op_i a op_j mají stejnou prioritu

- Pokud op_i a op_j jsou levě asociativní potom:

$$op_i > op_j \text{ a } op_j > op_i$$

- Pokud op_i a op_j jsou pravě asociativní potom:

$$op_i < op_j \text{ a } op_j < op_i$$

Příklad: Precedenční tabulka odvozená z asociativity operátorů gramatiky G_{expr2} :

$+$ je levě asociativní
 $*$ je levě asociativní



	+	*
+	>	
*		>

Konstrukce precedenční tabulky 3/5

3) Identifikátory:

- Pokud $a \in T$ může být hned před id_i , pak:
- Pokud $a \in T$ může být hned za id_i , pak:

$$a < id_i$$

$$id_i > a$$

Příklad: Část precedenční tabulka pro identifikátory:

$\$i * (i + i) * i$
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 $\$, (, +, *$ může být před i

$i * (i + i) * i \$$
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 $*, +,), \$$ může být za i

	+	*	()	i	\$
+					<	
*					<	
(<	
)						
i	>	>		>		>
\$					<	

Konstrukce precedenční tabulky 4/5

4) Závorky:

- Pro jeden pár závorek platí:
- Necht' $a \in T - \{), \$\}$. Pak:
- Necht' $a \in T - \{ (, \$\}$. Pak:
- Necht' $a \in T$ a a může být hned před (. Pak:
- Necht' $a \in T$ a a může být hned za). Pak:

(=)

(< a

a >)

a < (

) > a

Příklad: Část precedenční tabulky pro závorky

$\$(i + ((i * (i + (i + i))))))$

$\$, (, *, +$ může být před (

$(((((i + i) + i) * i) + i)\$$

$+, *,), \$$ může být za)

	+	*	()	i	\$
+			<	>		
*			<	>		
(<	<	<	=	<	
)	>	>		>		>
i				>		
\$			<			

Konstrukce precedenční tabulky 5/5

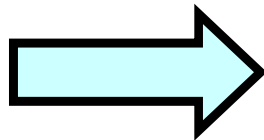
5) Ukončovač řetězce \$

- Nechť op_i je libovolný operátor:

$$\$ < \text{op}_i \text{ and } \text{op}_i > \$$$

Příklad: Část precedenční tabulky pro ukončovače:

$\$ < +$
 $\$ < *$
 $+ > \$$
 $* > \$$



	+	*	\$
+			>
*			>
\$	<	<	

Celkově:

	+	*	()	i	\$
+	>	<	<	>	<	>
*	>	>	<	>	<	>
(<	<	<	=	<	
)	>	>		>		>
i	>	>		>		>
\$	<	<	<		<	