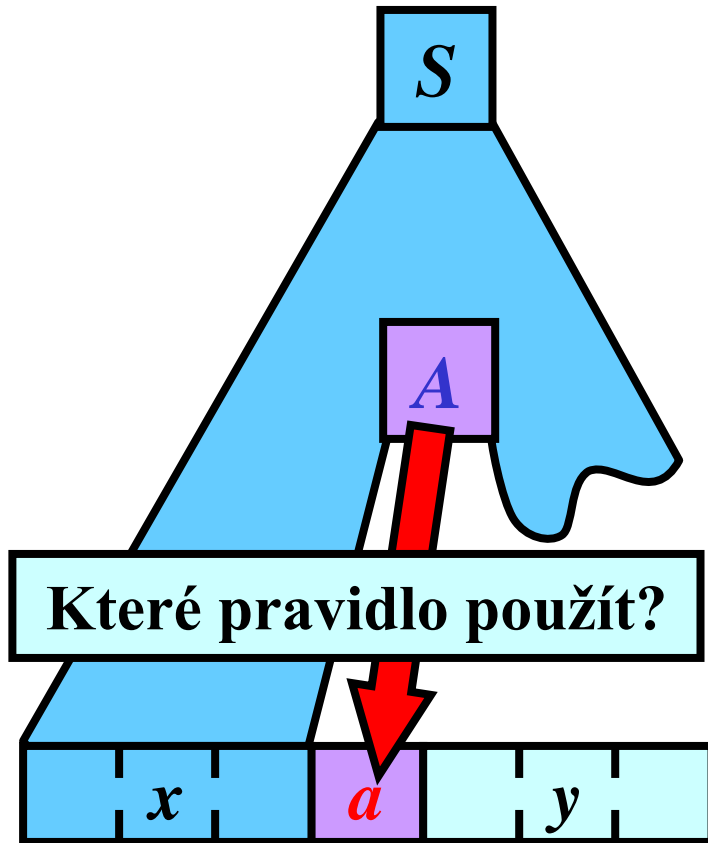


Kapitola VIII.
Syntaktická analýza
shora dolů

SA shora dolů: Úvod

Problém:



Myšlenka:

Tabulka:

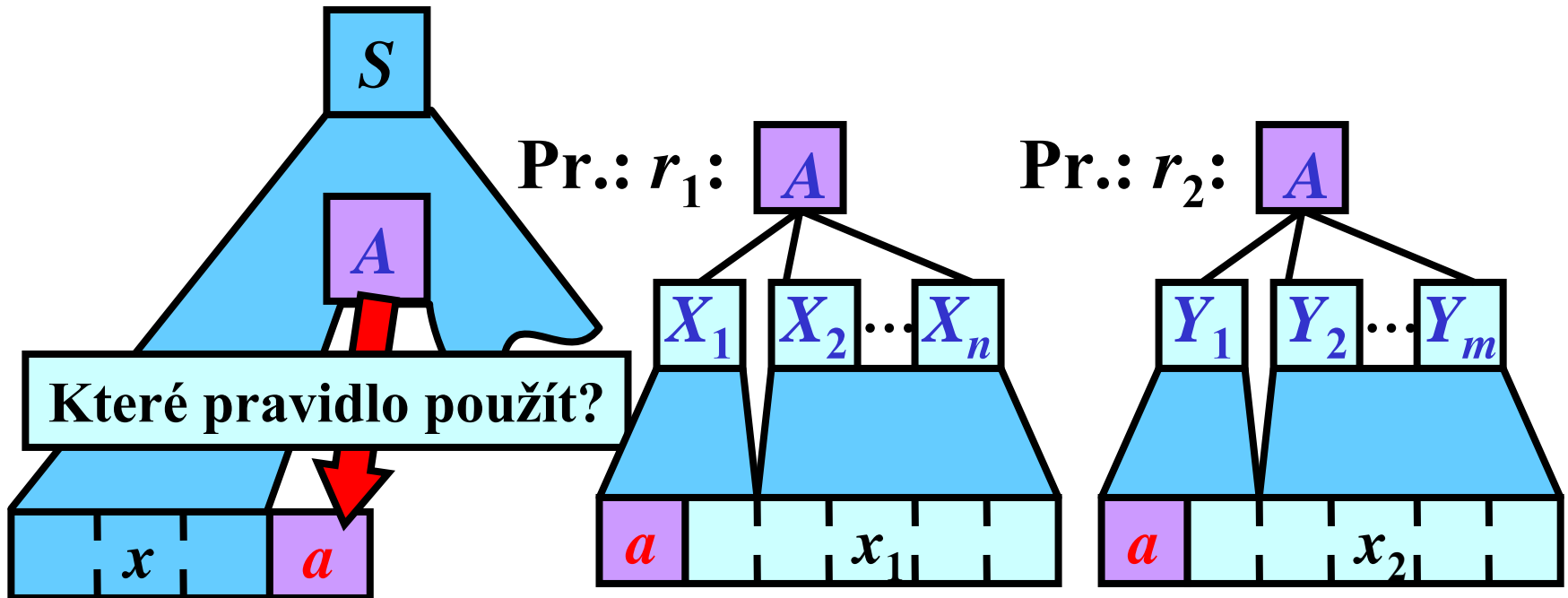
α	...	a	...
...			
A		$\alpha(A, a)$	
...			

Použij: $r: A \rightarrow x$

Otázka: Je možné sestrojít tuto tabulku pro **libovolnou** BKG?

Odpověď: **NE**

Výběr pravidla pomocí tabulky

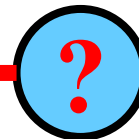


Tabulka:

α	...	a	...
...			
A		$\alpha(A, a)$	
...			

Použij r_1 : $A \rightarrow X_1 X_2 \dots X_n$

Použij r_2 : $A \rightarrow Y_1 Y_2 \dots Y_m$

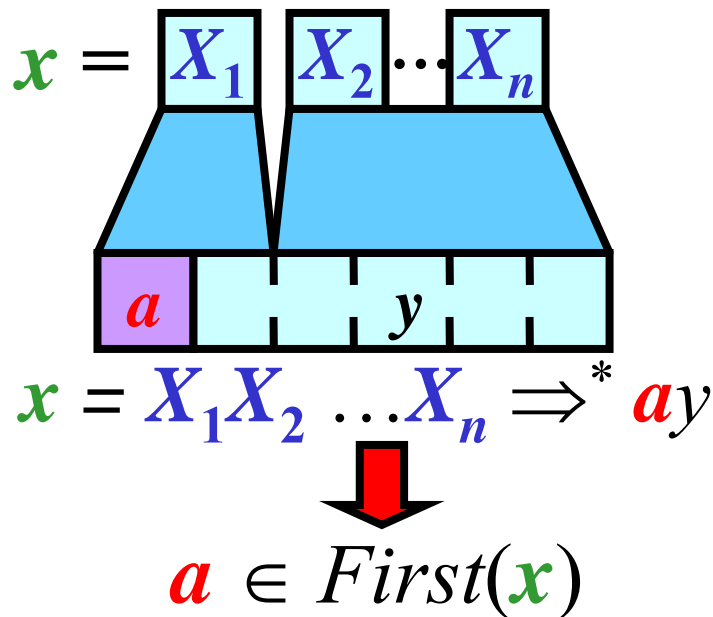


Množina *First*

Myšlenka: $First(x)$ je množina všech terminálů, kterými může začínat větná forma derivovatelná z x

Definice: Necht' $G = (N, T, P, S)$ je BKG. Pro každé $x \in (N \cup T)^*$ je definováno $First(x)$ jako:
 $First(x) = \{a: a \in T, x \Rightarrow^* ay; y \in (N \cup T)^*\}$.

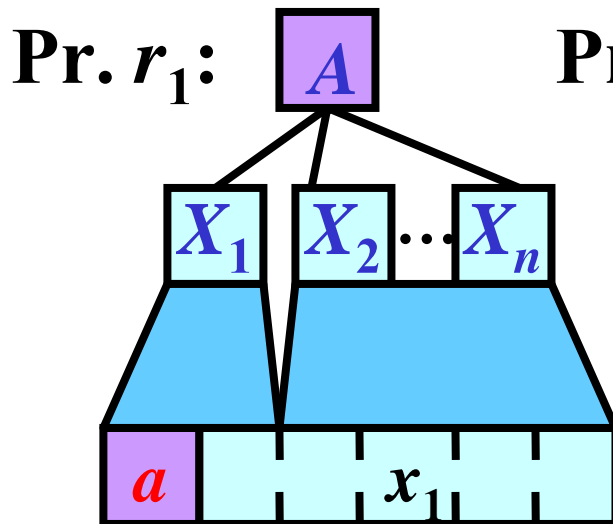
Ilustrace:



LL gramatika bez ε -pravidel

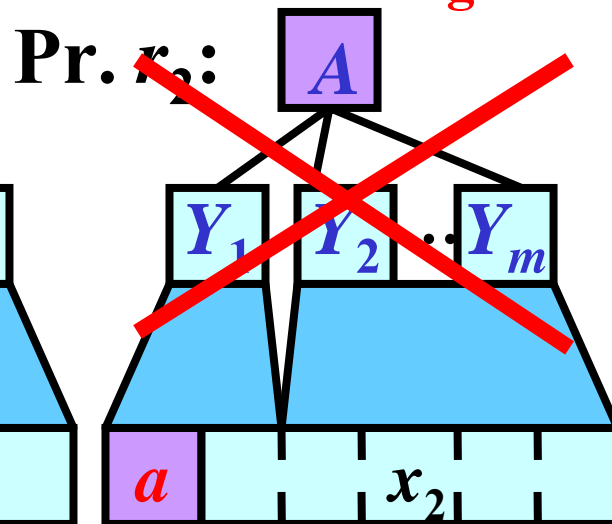
Definice: Necht' $G = (N, T, P, S)$ je BKG bez ε -pravidel. G je *LL gramatika*, pokud pro každé $a \in T$ a $A \in N$ existuje **maximálně jedno pravidlo** $A \rightarrow X_1X_2\dots X_n \in P$ takové, že: $a \in First(X_1X_2\dots X_n)$

Ilustrace:



$a \in First(X_1X_2\dots X_n)$

Nesmí nastat v LL-gramatice



$a \in First(Y_1Y_2\dots Y_m)$

Tabulka:

α	...	a	...
...			
A		$\alpha(A, a)$	
...			

↓

Pouze pravidlo:

$r_1: A \rightarrow X_1X_2\dots X_n$

Jednoduchý programovací jazyk (JPJ)

- 1: <prog> → begin <st-list>
 2: <st-list> → <stat> ; <st-list>
 3: <st-list> → end
 4: <stat> → read id
 5: <stat> → write <item>
 6: <stat> → id := add (<item> <it-list>
 7: <it-list> → , <item> <it-list>
 8: <it-list> →)
 9: <item> → int
 10: <item> → id

Pozn.: G_{JPJ} je LL gramatika

Příklad:

```
begin
  read i;
  j := add(i, 1);
  write j;
end
```

∈ JPJ

Algoritmus: $First(X)$

- **Vstup:** $G = (N, T, P, S)$ bez ε -pravidel
 - **Výstup:** $First(X)$ pro každé $X \in N \cup T$
-
- **Metoda:**
 - pro každé $a \in T$: $First(a) := \{a\}$
 - Použij následující pravidlo, dokud bude možné měnit nějakou množinu $First$:
 - if $A \rightarrow X_1X_2\dots X_n \in P$, then přidej $First(X_1)$ do $First(A)$
-

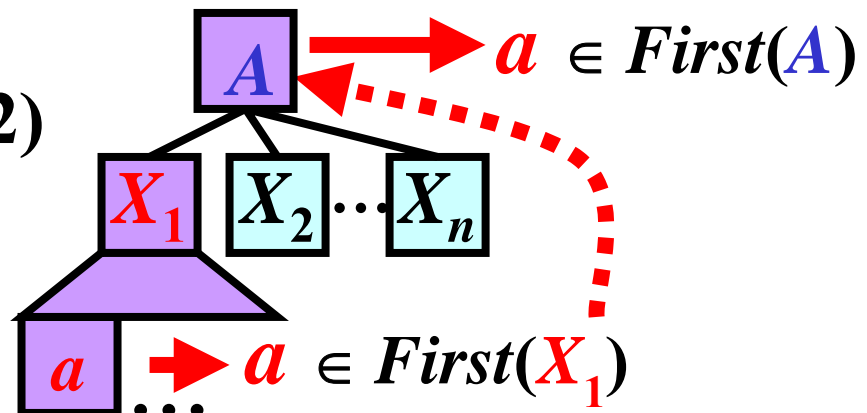
Ilustrace:

1) pro každé $a \in T$:

$$First(a) := \{a\},$$

protože $a \Rightarrow^0 a$

2)



$First(X)$ pro JPJ: Příklad

$First(\underline{\text{begin}}) := \{\underline{\text{begin}}\}$	$First(\underline{\text{id}}) := \{\underline{\text{id}}\}$	$First(\underline{,}) := \{\underline{,}\}$
$First(\underline{\text{end}}) := \{\underline{\text{end}}\}$	$First(\underline{\text{int}}) := \{\underline{\text{int}}\}$	$First(\underline{()}) := \{\underline{()}\}$
$First(\underline{\text{read}}) := \{\underline{\text{read}}\}$	$First(\underline{:=}) := \{\underline{:=}\}$	$First(\underline{)}) := \{\underline{)})\}$
$First(\underline{\text{write}}) := \{\underline{\text{write}}\}$	$First(\underline{\text{add}}) := \{\underline{\text{add}}\}$	$First(\underline{;}) := \{\underline{;}\}$

$\langle \text{item} \rangle \rightarrow \underline{\text{id}} \in P:$	přidej $First(\underline{\text{id}})$ do $First(\langle \text{item} \rangle)$
$\langle \text{item} \rangle \rightarrow \underline{\text{int}} \in P:$	přidej $First(\underline{\text{int}})$ do $First(\langle \text{item} \rangle)$
Celkově: $First(\langle \text{item} \rangle)$	$= \{\underline{\text{id}}, \underline{\text{int}}\}$

$\langle \text{it-list} \rangle \rightarrow \underline{)} \in P:$	přidej $First(\underline{)})$ do $First(\langle \text{it-list} \rangle)$
$\langle \text{it-list} \rangle \rightarrow \underline{,} \dots \in P:$	přidej $First(\underline{,})$ do $First(\langle \text{it-list} \rangle)$
Celkově: $First(\langle \text{it-list} \rangle)$	$= \{\underline{), \underline{,}}\}$

$\langle \text{stat} \rangle \rightarrow \underline{\text{id}} \dots \in P:$	přidej $First(\underline{\text{id}})$ do $First(\langle \text{stat} \rangle)$
$\langle \text{stat} \rangle \rightarrow \underline{\text{write}} \dots \in P:$	přidej $First(\underline{\text{write}})$ do $First(\langle \text{stat} \rangle)$
$\langle \text{stat} \rangle \rightarrow \underline{\text{read}} \dots \in P:$	přidej $First(\underline{\text{read}})$ do $First(\langle \text{stat} \rangle)$
Celkově: $First(\langle \text{stat} \rangle)$	$= \{\underline{\text{id}}, \underline{\text{write}}, \underline{\text{read}}\}$

$\langle \text{st-list} \rangle \rightarrow \underline{\text{end}} \in P:$	přidej $First(\underline{\text{end}})$ do $First(\langle \text{st-list} \rangle)$
$\langle \text{st-list} \rangle \rightarrow \langle \text{stat} \rangle \dots \in P:$	přidej $First(\langle \text{stat} \rangle)$ do $First(\langle \text{st-list} \rangle)$
Celkově: $First(\langle \text{st-list} \rangle)$	$= \{\underline{\text{id}}, \underline{\text{write}}, \underline{\text{read}}, \underline{\text{end}}\}$

$\langle \text{prog} \rangle \rightarrow \underline{\text{begin}} \dots \in P:$	přidej $First(\underline{\text{begin}})$ do $First(\langle \text{prog} \rangle)$
Celkově: $First(\langle \text{prog} \rangle)$	$= \{\underline{\text{begin}}\}$

Konstrukce LL-tabulky

α	...	a	...
...			
A		$\alpha(A, a)$	
...			

$\alpha(A, a) = A \rightarrow X_1X_2\dots X_n \in P$
 pokud $a \in First(X_1)$; jinak
 $\alpha(A, a)$ je prázdné \Rightarrow **CHYBA**

Vytvořme: LL tabulku

	id	int	:=	...
<prog>				
<st-list>	2	$id \in First(<stat>)$		
<stat>	6	$id \in First(id)$		
<it-list>				
<item>	10	$id \in First(id)$		

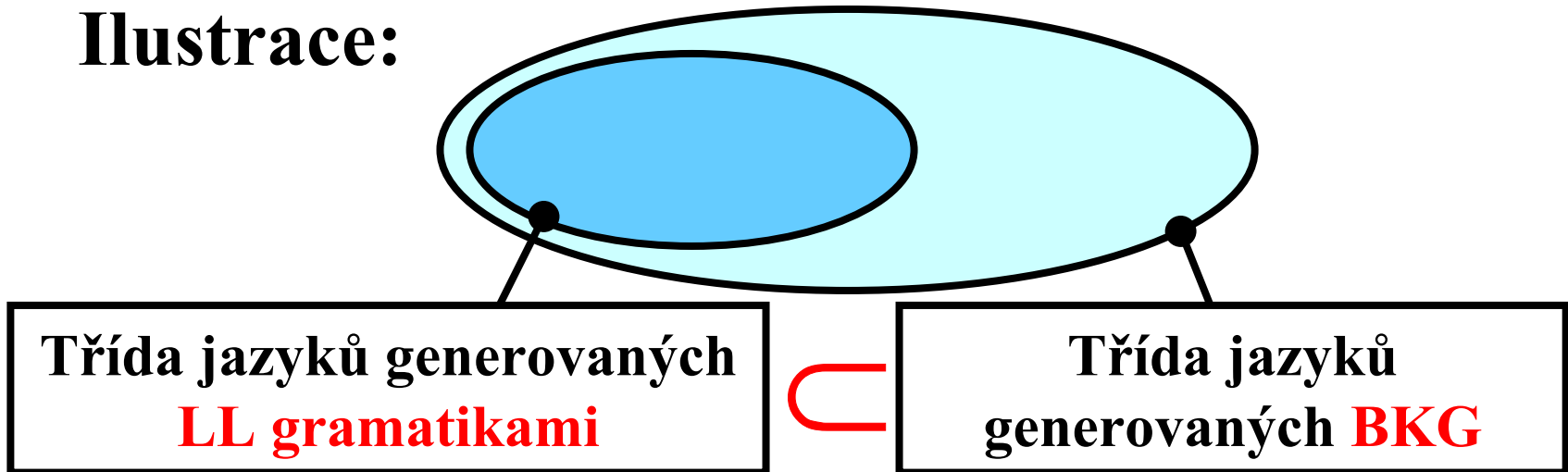
Prav. r: $A \rightarrow X_1X_2\dots X_n$	$First(X_1)$
1: <prog> \rightarrow begin ...	{ <u>begin</u> }
2: <st-list> \rightarrow <stat> ...	{ <u>id</u> , <u>write</u> , <u>read</u> }
3: <st-list> \rightarrow end	{ <u>end</u> }
4: <stat> \rightarrow read ...	{ <u>read</u> }
5: <stat> \rightarrow write ...	{ <u>write</u> }
6: <stat> \rightarrow id ...	{ <u>id</u> }
7: <it-list> \rightarrow , ...	{ <u>,</u> }
8: <it-list> \rightarrow)	{ <u>)</u> }
9: <item> \rightarrow int	{ <u>int</u> }
10: <item> \rightarrow id	{ <u>id</u> }

Zbytek vytvořme
analogicky.

LL gramatiky: Úspěšné transformace

Obecně: BKG jsou silnější než LL-gramatiky

Ilustrace:



• **Některé** BKG mohou být převedeny na ekvivalentní LL gramatiky pomocí následujících transformací:

- 1) Faktorizace (vytýkání)
- 2) Odstranění levé rekurze

Pozn.: Pravidlo tvaru $A \rightarrow Ax$, kde $A \in N$, $x \in (N \cup T)^*$ se nazývá *levě rekurzivní pravidlo*.

Faktorizace (vytýkání)

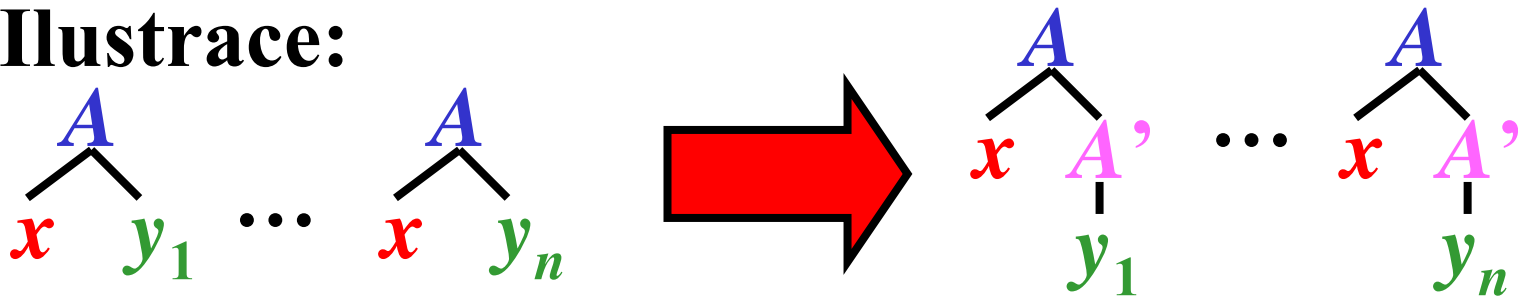
Myšlenka: Zaměnit pravidla tvaru:

$A \rightarrow xy_1, A \rightarrow xy_2, \dots, A \rightarrow xy_n$ na:

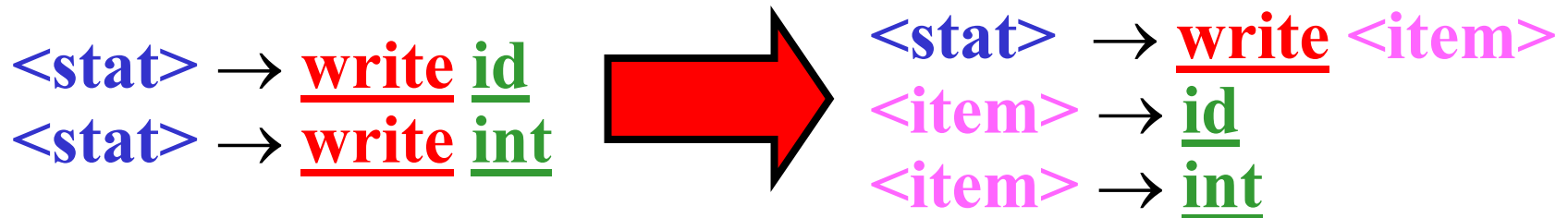
$A \rightarrow xA', A' \rightarrow y_1, A' \rightarrow y_2, \dots, A' \rightarrow y_n,$

kde A' je nový neterminál

Ilustrace:



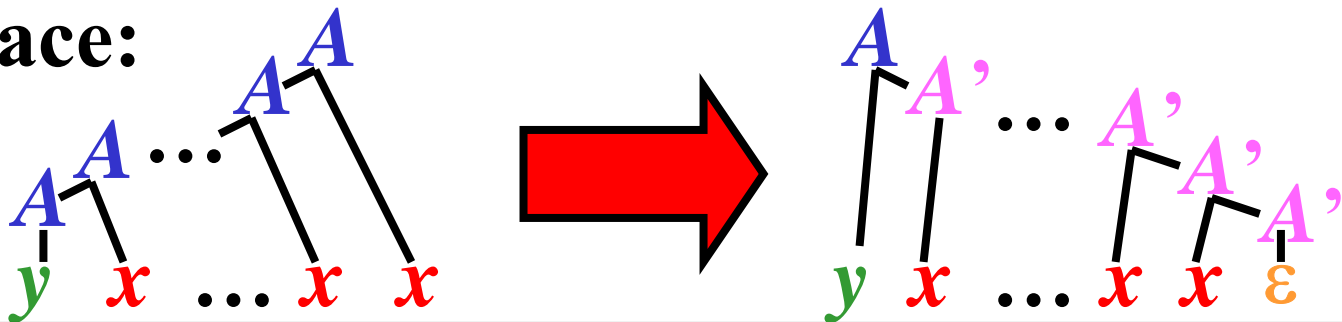
Příklad:



Odstranění levé rekurze

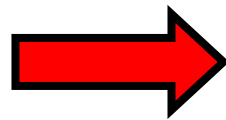
Myšlenka: Zaměnit pravidla tvaru: $A \rightarrow Ax$, $A \rightarrow y$ za: $A \rightarrow yA'$, $A' \rightarrow xA'$, $A' \rightarrow \varepsilon$, kde A' je nový neterminál.

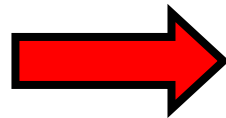
Ilustrace:



Příklad:

$$\left. \begin{array}{l} E \rightarrow E+T \\ E \rightarrow T \end{array} \right\}$$

$$\left. \begin{array}{l} T \rightarrow T*F \\ T \rightarrow F \end{array} \right\}$$


$$E \rightarrow TE', E' \rightarrow +TE', E' \rightarrow \varepsilon$$


$$T \rightarrow FT', T' \rightarrow *FT', T' \rightarrow \varepsilon$$

$$F \rightarrow (E)$$

$$F \rightarrow (E)$$

$$F \rightarrow i$$

$$F \rightarrow i$$

LL-gramatiky s ε -pravidly: Úvod

Proč ε -pravidla?

- Odstranění levé rekurze vytvoří ε -pravidla
- ε -pravidla často udělají gramatiku „čistější“

Zjednodušení této části:

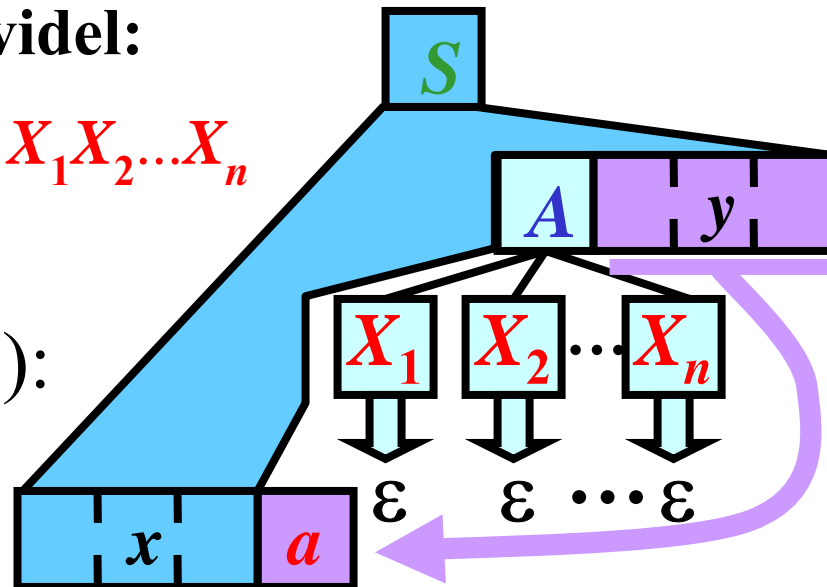
Budeme předpokládat, že každý vstupní řetězec je zakončen $\$$.

Pozn.: $\$$ značí „zakončovač“

Hlavní problém ε -pravidel:

Pravidlo $r: A \rightarrow X_1X_2\dots X_n$

Možná: $a \notin First(A)$:



Pozn.: Musíme definovat další množiny: *Empty*, *Follow* a *Predict*.

Gramatika pro aritmetické výrazy

- $G_{expr3} = (N, T, P, \mathbf{E})$, kde
- $N = \{\mathbf{E}, \mathbf{E}', \mathbf{F}, \mathbf{F}', \mathbf{T}\}$,
- $T = \{\mathbf{i}, \mathbf{+}, \mathbf{*}, \mathbf{(}, \mathbf{)}\}$,
- $P = \{$

$\mathbf{1: E} \rightarrow \mathbf{TE}'$,	$\mathbf{2: E}' \rightarrow \mathbf{+TE}'$,
$\mathbf{3: E}' \rightarrow \varepsilon$,	$\mathbf{4: T} \rightarrow \mathbf{FT}'$,
$\mathbf{5: T}' \rightarrow \mathbf{*FT}'$,	$\mathbf{6: T}' \rightarrow \varepsilon$,
$\mathbf{7: F} \rightarrow \mathbf{(E)}$,	$\mathbf{8: F} \rightarrow \mathbf{i}$

 $\}$

Příklad:

$$(i + i) * (i + i) \in L(G_{expr3})$$

Množina *Empty*

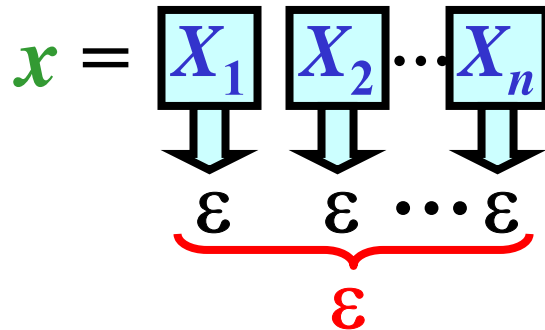
Myšlenka: *Empty*(x) je množina, která obsahuje jediný prvek ε pokud x derivuje ε , jinak je prázdná

Definice: Necht' $G = (N, T, P, S)$ je BKG.

$Empty(\mathbf{x}) = \{\varepsilon\}$ if $\mathbf{x} \Rightarrow^* \varepsilon$; jinak

$Empty(\mathbf{x}) = \emptyset$, kde $x \in (N \cup T)^*$.

Ilustrace:



$$\mathbf{x} = X_1 X_2 \cdots X_n \Rightarrow^* \varepsilon$$



$$Empty(\mathbf{x}) = \{\varepsilon\}$$

Algoritmus: *Empty*(X)

- **Vstup:** $G = (N, T, P, S)$
 - **Výstup:** $Empty(X)$ pro každý symbol $X \in N \cup T$
-
- **Metoda:**
 - pro každé $a \in T$: $Empty(a) := \emptyset$
 - pro každé $A \in N$:
 - if $A \rightarrow \varepsilon \in P$ then $Empty(A) := \{\varepsilon\}$
 - else $Empty(A) := \emptyset$
 - Použij následující pravidlo, dokud bude možné měnit nějakou množinu *Empty*:
 - if $A \rightarrow X_1 X_2 \dots X_n \in P$ and $Empty(X_i) = \{\varepsilon\}$ pro všechna $i = 1, \dots, n$ then $Empty(A) = \{\varepsilon\}$

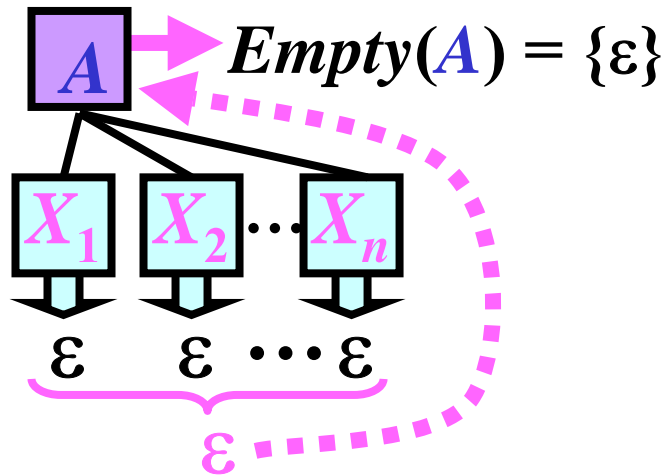
Předchozí algoritmus: Ilustrace

1) Pro každé $a \in T$: $Empty(a) := \emptyset$, protože $a \Rightarrow^* \varepsilon$

2) Pro každé $r: A \rightarrow \varepsilon \in P$: $Empty(A) := \{\varepsilon\}$, protože $A \Rightarrow^1 \varepsilon [r]$

3) Použij následující pravidlo, dokud bude možné měnit nějakou množinu $Empty$:

- if $A \rightarrow X_1 X_2 \dots X_n \in P$ and $Empty(X_i) = \{\varepsilon\}$
pro všechna $i = 1, \dots, n$ then $Empty(A) = \{\varepsilon\}$



$Empty(X)$ pro G_{expr3} : Příklad

$G_{expr3} = (N, T, P, \mathbf{E})$, kde: $N = \{\mathbf{E}, \mathbf{F}, \mathbf{T}\}$, $T = \{\mathbf{i}, +, *, (,)\}$,
 $P = \{$ **1**: $\mathbf{E} \rightarrow \mathbf{TE}'$, **2**: $\mathbf{E}' \rightarrow +\mathbf{TE}'$, **3**: $\mathbf{E}' \rightarrow \varepsilon$, **4**: $\mathbf{T} \rightarrow \mathbf{FT}'$
5: $\mathbf{T}' \rightarrow *\mathbf{FT}'$, **6**: $\mathbf{T}' \rightarrow \varepsilon$, **7**: $\mathbf{F} \rightarrow (\mathbf{E})$, **8**: $\mathbf{F} \rightarrow \mathbf{i} \}$

Inicializace:

$Empty(\mathbf{i})$	$:= \emptyset$	$Empty(\mathbf{E})$	$:= \emptyset$
$Empty(+)$	$:= \emptyset$	$Empty(\mathbf{E}')$	$:= \{\varepsilon\}$
$Empty(*)$	$:= \emptyset$	$Empty(\mathbf{T})$	$:= \emptyset$
$Empty(($	$:= \emptyset$	$Empty(\mathbf{T}')$	$:= \{\varepsilon\}$
$Empty($	$:= \emptyset$	$Empty(\mathbf{F})$	$:= \emptyset$

- **Žádná $Empty$ množina již nemůže být změněna**

Algoritmus: $First(X)$

- **Vstup:** $G = (N, T, P, S)$
 - **Výstup:** $First(X)$ pro každé $X \in N \cup T$
-
- **Metoda:**
 - pro každé $a \in T$: $First(a) := \{a\}$
 - pro každé $A \in N$: $First(A) := \emptyset$
 - **Používej následující pravidlo, dokud bude možné měnit nějakou množinu $First$:**
 - **if** $A \rightarrow X_1 X_2 \dots X_{k-1} X_k \dots X_n \in P$ **then**
 - přidej všechny symboly z $First(X_1)$ do $First(A)$
 - **if** $Empty(X_i) = \{\varepsilon\}$ pro $i = 1, \dots, k-1$, kde $k \leq n$ **then** přidej všechny symboly z $First(X_k)$ do $First(A)$

Předchozí algoritmus: Ilustrace

1) pro každé $a \in T$: $First(a) := \{a\}$, protože $a \Rightarrow^0 a$

2) pro každé $A \in N$: $First(A) := \emptyset$ (Inicializace)

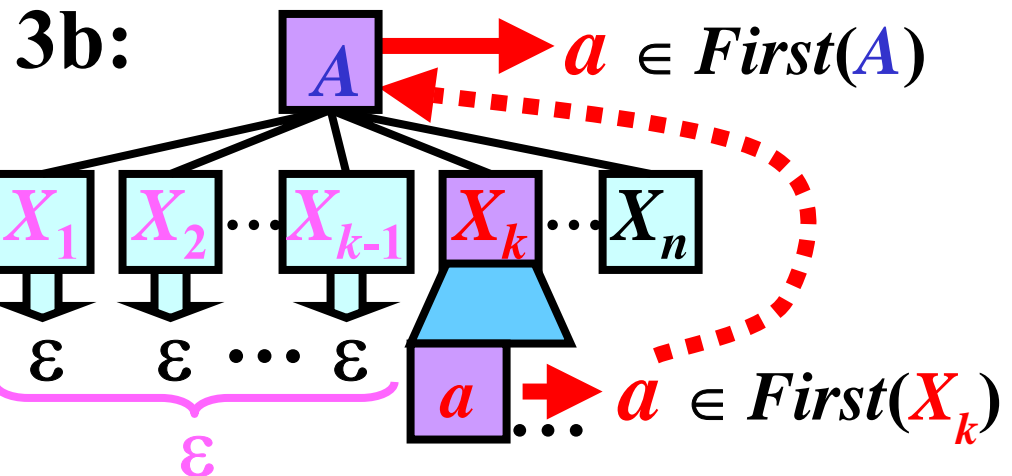
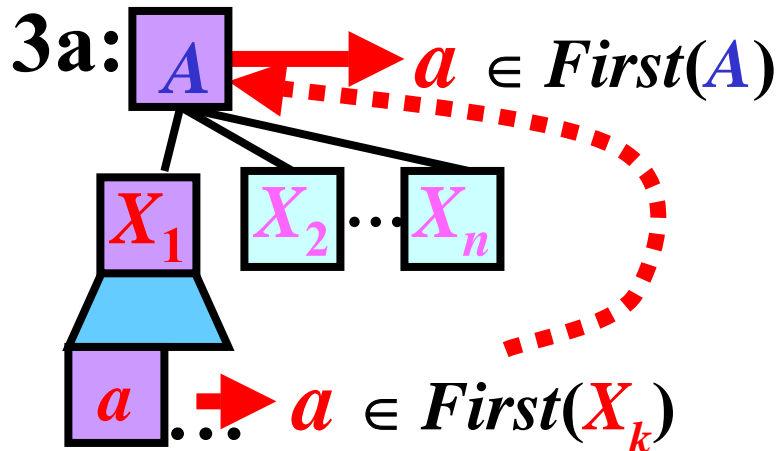
3) Použij následující pravidlo, dokud bude možné měnit nějakou množinu $First$:

• if $A \rightarrow X_1 X_2 \dots X_{k-1} X_k \dots X_n \in P$ then

3a) přidej všechny symboly z $First(X_1)$ do $First(A)$

3b) if $Empty(X_i) = \{\epsilon\}$ pro $i = 1, \dots, k-1$, kde $k < n$

then přidej všechny symboly z $First(X_k)$ do $First(A)$:



$First(X)$ for G_{expr3} : Příklad

Inicializace:

$First(i)$	$:= \{i\}$	$First(E)$	$:= \emptyset$
$First(+)$	$:= \{+\}$	$First(E')$	$:= \emptyset$
$First(*)$	$:= \{*\}$	$First(T)$	$:= \emptyset$
$First(()$	$:= \{(,)\}$	$First(T')$	$:= \emptyset$
$First()$	$:= \{) \}$	$First(F)$	$:= \emptyset$

$F \rightarrow i \in P$: přidej $First(i) = \{i\}$ do $First(F)$

$F \rightarrow (E) \in P$: přidej $First(() = \{(,)\}$ do $First(F)$

Celkově: $First(F) = \{i, (,)\}$

$T' \rightarrow *FT' \in P$: přidej $First(*) = \{*\}$ do $First(T')$

Celkově: $First(T') = \{*\}$

$T \rightarrow FT' \in P$: přidej $First(F) = \{i, (,)\}$ do $First(T)$

Celkově: $First(T) = \{i, (,)\}$

$E' \rightarrow +TE' \in P$: přidej $First(+) = \{+\}$ do $First(E')$

Celkově: $First(E') = \{+\}$

$E \rightarrow TE' \in P$: přidej $First(T) = \{i, (,)\}$ do $First(E)$

Celkově: $First(E) = \{i, (,)\}$

- **Žádná $First$ množina již nemůže být změněna.**

$First(X)$ & $Empty(X)$ pro G_{expr3} : Celkově

$G_{expr3} = (N, T, P, E)$, kde: $N = \{E, F, T\}$, $T = \{i, +, *, (,)\}$,
 $P = \{$ 1: $E \rightarrow TE'$, 2: $E' \rightarrow +TE'$, 3: $E' \rightarrow \varepsilon$, 4: $T \rightarrow FT'$
 5: $T' \rightarrow *FT'$, 6: $T' \rightarrow \varepsilon$, 7: $F \rightarrow (E)$, 8: $F \rightarrow i \}$

Množina $Empty$ pro všechna $X \in N \cup T$:	$Empty(i)$	$:= \emptyset$	$Empty(E)$	$:= \emptyset$
	$Empty(+)$	$:= \emptyset$	$Empty(E')$	$:= \{\varepsilon\}$
	$Empty(*)$	$:= \emptyset$	$Empty(T)$	$:= \emptyset$
	$Empty(($	$:= \emptyset$	$Empty(T')$	$:= \{\varepsilon\}$
	$Empty()$	$:= \emptyset$	$Empty(F)$	$:= \emptyset$

Množina $First$ pro všechna $X \in N \cup T$:	$First(i)$	$:= \{i\}$	$First(E)$	$:= \{i, ($
	$First(+)$	$:= \{+\}$	$First(E')$	$:= \{+\}$
	$First(*)$	$:= \{*\}$	$First(T)$	$:= \{i, ($
	$First(($	$:= \{($	$First(T')$	$:= \{*\}$
	$First()$	$:= \{)$	$First(F)$	$:= \{i, ($

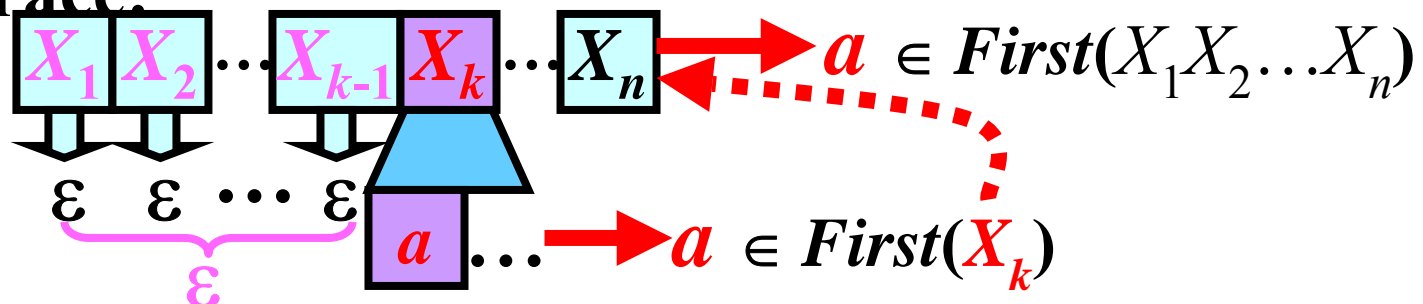
Pozn.: pro každé $a \in T$: $Empty(a) = \emptyset$, $First(a) = \{a\}$

Algoritmus: $First(X_1X_2\dots X_n)$

- **Vstup:** $G = (N, T, P, S)$; $First(X)$ & $Empty(X)$ pro každé $X \in N \cup T$; $x = X_1X_2\dots X_n$, kde $x \in (N \cup T)^+$
 - **Výstup:** $First(X_1X_2\dots X_n)$
-
- **Metoda:**
 - $First(X_1X_2\dots X_n) := First(X_1)$
 - Použij následující pravidlo, dokud bude možné měnit množinu $First(X_1X_2\dots X_{k-1}X_k\dots X_n)$:
 - if $Empty(X_i) = \{\varepsilon\}$ pro $i = 1, \dots, k-1$, kde $k \leq n$
 - then přidej všechny symboly z $First(X_k)$ do $First(X_1X_2\dots X_n)$
-

! Pozn.: $First(\varepsilon) = \emptyset$

Ilustrace:



$First(X_1X_2\dots X_n)$: Příklad

$G_{expr3} = (N, T, P, \mathbf{E})$, kde: $N = \{\mathbf{E}, \mathbf{F}, \mathbf{T}\}$, $T = \{i, +, *, (,)\}$,

$P = \{$ **1**: $\mathbf{E} \rightarrow \mathbf{TE}'$, **2**: $\mathbf{E}' \rightarrow +\mathbf{TE}'$, **3**: $\mathbf{E}' \rightarrow \varepsilon$, **4**: $\mathbf{T} \rightarrow \mathbf{FT}'$
5: $\mathbf{T}' \rightarrow *\mathbf{FT}'$, **6**: $\mathbf{T}' \rightarrow \varepsilon$, **7**: $\mathbf{F} \rightarrow (\mathbf{E})$, **8**: $\mathbf{F} \rightarrow i$ $\}$

Množiny Empty & First pro všechna $X \in N$:	$Empty(\mathbf{E})$	$:= \emptyset$	$First(\mathbf{E})$	$:= \{i, (\}$
	$Empty(\mathbf{E}')$	$:= \{\varepsilon\}$	$First(\mathbf{E}')$	$:= \{+\}$
	$Empty(\mathbf{T})$	$:= \emptyset$	$First(\mathbf{T})$	$:= \{i, (\}$
	$Empty(\mathbf{T}')$	$:= \{\varepsilon\}$	$First(\mathbf{T}')$	$:= \{*\}$
	$Empty(\mathbf{F})$	$:= \emptyset$	$First(\mathbf{F})$	$:= \{i, (\}$

Určeme: $First(\mathbf{E}'\mathbf{T}'\mathbf{F}\mathbf{E}\mathbf{T})$

1) $First(\underline{\mathbf{E}}'\mathbf{T}'\mathbf{F}\mathbf{E}\mathbf{T}) := First(\mathbf{E}') = \{+\}$

2) $First(\underline{\mathbf{E}}'\underline{\mathbf{T}}'\mathbf{F}\mathbf{E}\mathbf{T})$: přidej $First(\mathbf{T}') = \{*\}$ do $First(\mathbf{E}'\mathbf{T}'\mathbf{F}\mathbf{E}\mathbf{T})$

$Empty(\mathbf{E}') = \{\varepsilon\}$

3) $First(\underline{\mathbf{E}}'\mathbf{T}'\underline{\mathbf{F}}\mathbf{E}\mathbf{T})$: přidej $First(\mathbf{F}) = \{i, (\}$ do $First(\mathbf{E}'\mathbf{T}'\mathbf{F}\mathbf{E}\mathbf{T})$

$Empty(\mathbf{E}') = Empty(\mathbf{T}') = \{\varepsilon\}$

Celkově: $First(\mathbf{E}'\mathbf{T}'\mathbf{F}\mathbf{E}\mathbf{T}) = \{+, *, i, (\}$

Algoritmus: $Empty(X_1X_2\dots X_n)$

- **Vstup:** $G = (N, T, P, S)$; $Empty(X)$ pro všechna $X \in N \cup T$;
 $x = X_1X_2\dots X_n$, kde $x \in (N \cup T)^+$
- **Výstup:** $Empty(X_1X_2\dots X_n)$

• **Metoda:**

- **if** $Empty(X_i) = \{\varepsilon\}$ pro all $i = 1, \dots, n$ **then**

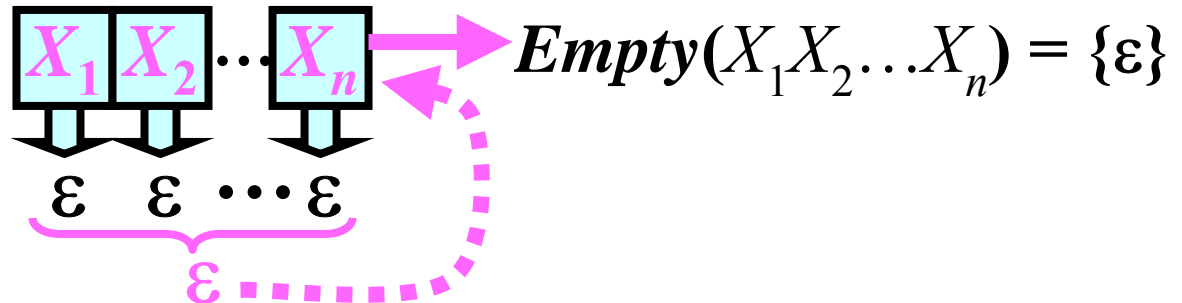
$$Empty(X_1X_2\dots X_n) := \{\varepsilon\}$$

else

$$Empty(X_1X_2\dots X_n) := \emptyset$$

! Pozn.: $Empty(\varepsilon) = \{\varepsilon\}$

Ilustrace:



$Empty(X_1 X_2 \dots X_n)$: Příklad

$G_{expr3} = (N, T, P, \mathbf{E})$, kde: $N = \{\mathbf{E}, \mathbf{F}, \mathbf{T}\}$, $T = \{\mathbf{i}, +, *, (,)\}$,
 $P = \{$ **1**: $\mathbf{E} \rightarrow \mathbf{TE}'$, **2**: $\mathbf{E}' \rightarrow +\mathbf{TE}'$, **3**: $\mathbf{E}' \rightarrow \varepsilon$, **4**: $\mathbf{T} \rightarrow \mathbf{FT}'$
5: $\mathbf{T}' \rightarrow *\mathbf{FT}'$, **6**: $\mathbf{T}' \rightarrow \varepsilon$, **7**: $\mathbf{F} \rightarrow (\mathbf{E})$, **8**: $\mathbf{F} \rightarrow \mathbf{i} \}$

Množina $Empty$ $Empty(\mathbf{E})$ $:= \emptyset$
pro všechna $X \in N$: $Empty(\mathbf{E}')$ $:= \{\varepsilon\}$
 $Empty(\mathbf{T})$ $:= \emptyset$
 $Empty(\mathbf{T}')$ $:= \{\varepsilon\}$
 $Empty(\mathbf{F})$ $:= \emptyset$

Určeme: $Empty(\mathbf{E}'\mathbf{T}')$

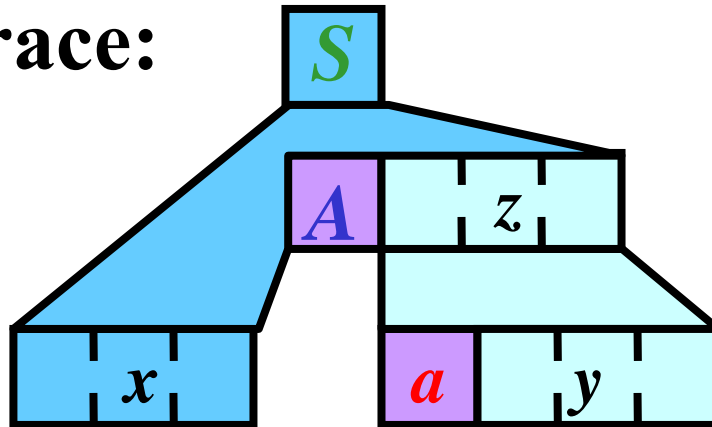
$Empty(\mathbf{E}') = Empty(\mathbf{T}') = \{\varepsilon\}$, tedy $Empty(\mathbf{E}'\mathbf{T}') = \{\varepsilon\}$

Množina *Follow*

Myšlenka: $Follow(A)$ je množina všech terminálů, které se mohou vyskytovat vpravo od A ve větné formě.

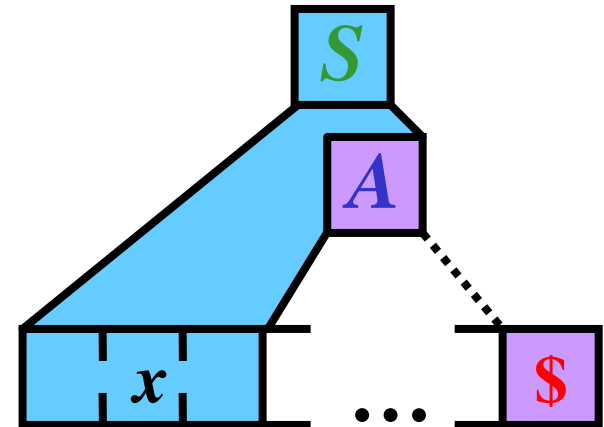
Definice: Necht' $G = (N, T, P, S)$ je BKG. Pro všechna $A \in N$ definujeme množinu $Follow(A)$:
 $Follow(A) = \{a: a \in T, S \Rightarrow^* xAay, x, y \in (N \cup T)^*\}$
 $\cup \{\$, S \Rightarrow^* xA, x \in (N \cup T)^*\}$

Ilustrace:



$$S \Rightarrow^* xAz \Rightarrow^* xAay$$

$$a \in Follow(A)$$



$$S \Rightarrow^* xA$$

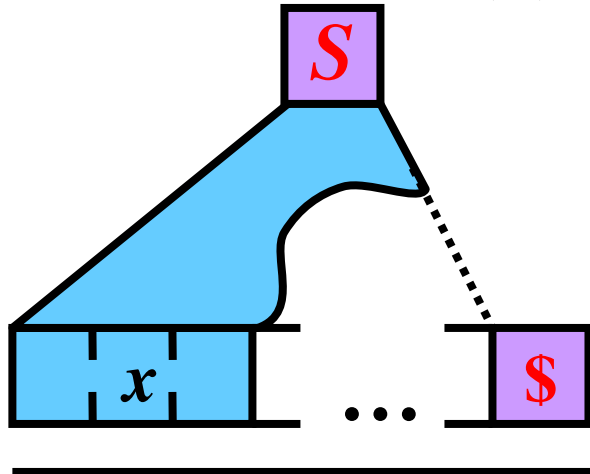
$$\$ \in Follow(A)$$

Algoritmus: *Follow*(*A*)

- **Vstup:** $G = (N, T, P, \mathbf{S})$;
 - **Výstup:** $Follow(A)$ pro každé $A \in N$
-
- **Metoda:**
 - $Follow(\mathbf{S}) := \{\mathbf{\$}\}$;
 - **Použijte následující pravidlo, dokud bude možné měnit nějakou množinu *Follow*:**
 - **if $A \rightarrow x\mathbf{B}y \in P$ then**
 - **if $y \neq \varepsilon$ then**
přidej všechny symboly z $First(y)$ do $Follow(\mathbf{B})$;
 - **if $Empty(y) = \{\varepsilon\}$ then**
přidej všechny symboly z $Follow(\mathbf{A})$ do $Follow(\mathbf{B})$;

Předchozí algoritmus: Ilustrace

1) $Follow(S) := \{\$ \}$



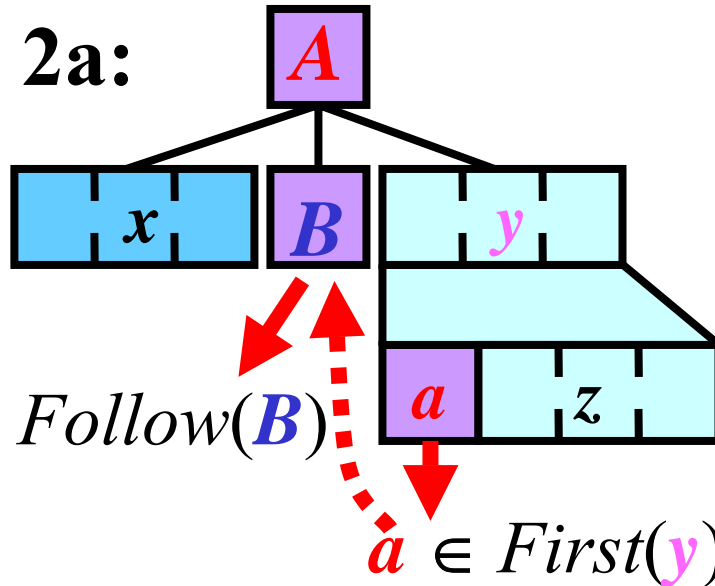
2) Použij následující pravidlo, dokud bude možné měnit $Follow$:

• if $A \rightarrow xBy \in P$ then

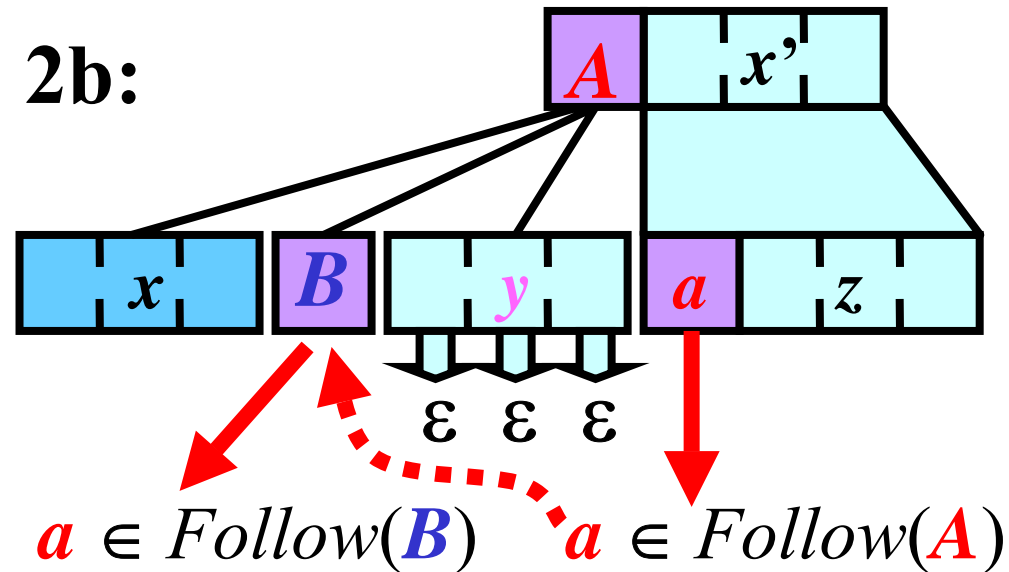
2a) if $y \neq \epsilon$ then přidej všechny symboly z $First(y)$ do $Follow(B)$

2b) if $Empty(y) = \{\epsilon\}$ then přidej všechny symboly z $Follow(A)$ do $Follow(B)$

2a:



2b:



$Follow(X)$ pro G_{expr3} : Příklad 1/3

$First(\mathbf{E})$	$:= \{i, ($	$Empty(\mathbf{E})$	$:= \emptyset$	$Follow(\mathbf{E})$	$:= \emptyset$
$First(\mathbf{E}')$	$:= \{+\}$	$Empty(\mathbf{E}')$	$:= \{\varepsilon\}$	$Follow(\mathbf{E}')$	$:= \emptyset$
$First(\mathbf{T})$	$:= \{i, ($	$Empty(\mathbf{T})$	$:= \emptyset$	$Follow(\mathbf{T})$	$:= \emptyset$
$First(\mathbf{T}')$	$:= \{*\}$	$Empty(\mathbf{T}')$	$:= \{\varepsilon\}$	$Follow(\mathbf{T}')$	$:= \emptyset$
$First(\mathbf{F})$	$:= \{i, ($	$Empty(\mathbf{F})$	$:= \emptyset$	$Follow(\mathbf{F})$	$:= \emptyset$

0) $Follow(\mathbf{E}) := \{\$,)\}$

1) $\mathbf{F} \rightarrow (\mathbf{E}) \in P$: přidej $First() = \{)\}$ do $Follow(\mathbf{E})$
 $\neq \varepsilon$

Celkově: $Follow(\mathbf{E}) = \{\$,)\}$

2) $\mathbf{E} \rightarrow \mathbf{T}\mathbf{E}' \in P$: přidej $Follow(\mathbf{E}) = \{\$,)\}$ do $Follow(\mathbf{E}')$
 $\varepsilon: Empty(\varepsilon) = \{\varepsilon\}$

$\mathbf{E} \rightarrow \mathbf{T}\mathbf{E}' \in P$: přidej $First(\mathbf{E}') = \{+\}$ do $Follow(\mathbf{T})$
 $\neq \varepsilon$

$\mathbf{E} \rightarrow \mathbf{T}\mathbf{E}' \in P$: přidej $Follow(\mathbf{E}) = \{\$,)\}$ do $Follow(\mathbf{T})$
 $Empty(\mathbf{E}') = \{\varepsilon\}$

Celkově: $Follow(\mathbf{E}') = \{\$,)\}$, $Follow(\mathbf{T}) = \{+, \$,)\}$

Follow(X) pro G_{expr3} : Příklad 2/3

$First(\mathbf{E}) := \{i, ($	$Empty(\mathbf{E}) := \emptyset$	$Follow(\mathbf{E}) := \{\$,)\}$
$First(\mathbf{E}') := \{+\}$	$Empty(\mathbf{E}') := \{\varepsilon\}$	$Follow(\mathbf{E}') := \{\$,)\}$
$First(\mathbf{T}) := \{i, ($	$Empty(\mathbf{T}) := \emptyset$	$Follow(\mathbf{T}) := \{+, \$,)\}$
$First(\mathbf{T}') := \{*\}$	$Empty(\mathbf{T}') := \{\varepsilon\}$	$Follow(\mathbf{T}') := \emptyset$
$First(\mathbf{F}) := \{i, ($	$Empty(\mathbf{F}) := \emptyset$	$Follow(\mathbf{F}) := \emptyset$

3) $\mathbf{E}' \rightarrow +\mathbf{T}\mathbf{E}' \underset{\varepsilon}{\downarrow} \in P$: přidej $Follow(\mathbf{E}') = \{\$,)\}$ do $Follow(\mathbf{E}')$
 $\varepsilon: Empty(\varepsilon) = \{\varepsilon\}$

$\mathbf{E}' \rightarrow +\mathbf{T}\mathbf{E}' \underset{\neq \varepsilon}{\downarrow} \in P$: přidej $First(\mathbf{E}') = \{+\}$ do $Follow(\mathbf{T})$

$\mathbf{E}' \rightarrow +\mathbf{T}\mathbf{E}' \underset{\neq \varepsilon}{\downarrow} \in P$: přidej $Follow(\mathbf{E}') = \{\$,)\}$ do $Follow(\mathbf{T})$
 $Empty(\mathbf{E}') = \{\varepsilon\}$

Celkově: Nic nezměněno

4) $\mathbf{T} \rightarrow \mathbf{F}\mathbf{T}' \underset{\varepsilon}{\downarrow} \in P$: přidej $Follow(\mathbf{T}) = \{+, \$,)\}$ do $Follow(\mathbf{T}')$
 $\varepsilon: Empty(\varepsilon) = \{\varepsilon\}$

$\mathbf{T} \rightarrow \mathbf{F}\mathbf{T}' \underset{\neq \varepsilon}{\downarrow} \in P$: přidej $First(\mathbf{T}') = \{*\}$ do $Follow(\mathbf{F})$

$\mathbf{T} \rightarrow \mathbf{F}\mathbf{T}' \underset{\neq \varepsilon}{\downarrow} \in P$: přidej $Follow(\mathbf{T}) = \{+, \$,)\}$ do $Follow(\mathbf{F})$
 $Empty(\mathbf{T}') = \{\varepsilon\}$

Celkově: $Follow(\mathbf{T}') = \{+, \$,)\}$, $Follow(\mathbf{F}) = \{*, +, \$,)\}$

Follow(X) pro G_{expr3} : Příklad 3/3

$First(\mathbf{E}) := \{i, ($	$Empty(\mathbf{E}) := \emptyset$	$Follow(\mathbf{E}) := \{\$,)\}$
$First(\mathbf{E}') := \{+\}$	$Empty(\mathbf{E}') := \{\varepsilon\}$	$Follow(\mathbf{E}') := \{\$,)\}$
$First(\mathbf{T}) := \{i, ($	$Empty(\mathbf{T}) := \emptyset$	$Follow(\mathbf{T}) := \{+, \$,)\}$
$First(\mathbf{T}') := \{*\}$	$Empty(\mathbf{T}') := \{\varepsilon\}$	$Follow(\mathbf{T}') := \{+, \$,)\}$
$First(\mathbf{F}) := \{i, ($	$Empty(\mathbf{F}) := \emptyset$	$Follow(\mathbf{F}) := \{*, +, \$,)\}$

5) $\mathbf{T}' \rightarrow *F\mathbf{T}' \in P$: přidej $Follow(\mathbf{T}') = \{+, \$,)\}$ do $Follow(\mathbf{T}')$
 ε : $Empty(\varepsilon) = \{\varepsilon\}$

$\mathbf{T}' \rightarrow *F\mathbf{T}' \in P$: přidej $First(\mathbf{T}') = \{*\}$ do $Follow(\mathbf{F})$

$\mathbf{T}' \rightarrow *F\mathbf{T}' \in P$: přidej $Follow(\mathbf{T}') = \{+, \$,)\}$ do $Follow(\mathbf{F})$
 $Empty(\mathbf{T}') = \{\varepsilon\}$

Konec: Žádná množina $Follow$ nemůže být změněna.

Celkově:

$Follow(\mathbf{E}) := \{\$,)\}$
$Follow(\mathbf{E}') := \{\$,)\}$
$Follow(\mathbf{T}) := \{+, \$,)\}$
$Follow(\mathbf{T}') := \{+, \$,)\}$
$Follow(\mathbf{F}) := \{*, +, \$,)\}$

Množina *Predict*

Myšlenka: $Predict(A \rightarrow x)$ je množina všech terminálů, které mohou být aktuálně nejlevěji vygenerovány, pokud pro libovolnou větnou formu použijeme pravidlo $A \rightarrow x$.

Definice: Necht' $G = (N, T, P, S)$ je BKGG. Pro každé $A \rightarrow x \in P$ definujeme množinu $Predict(A \rightarrow x)$ jako:

- pokud $Empty(x) = \{\varepsilon\}$ potom:

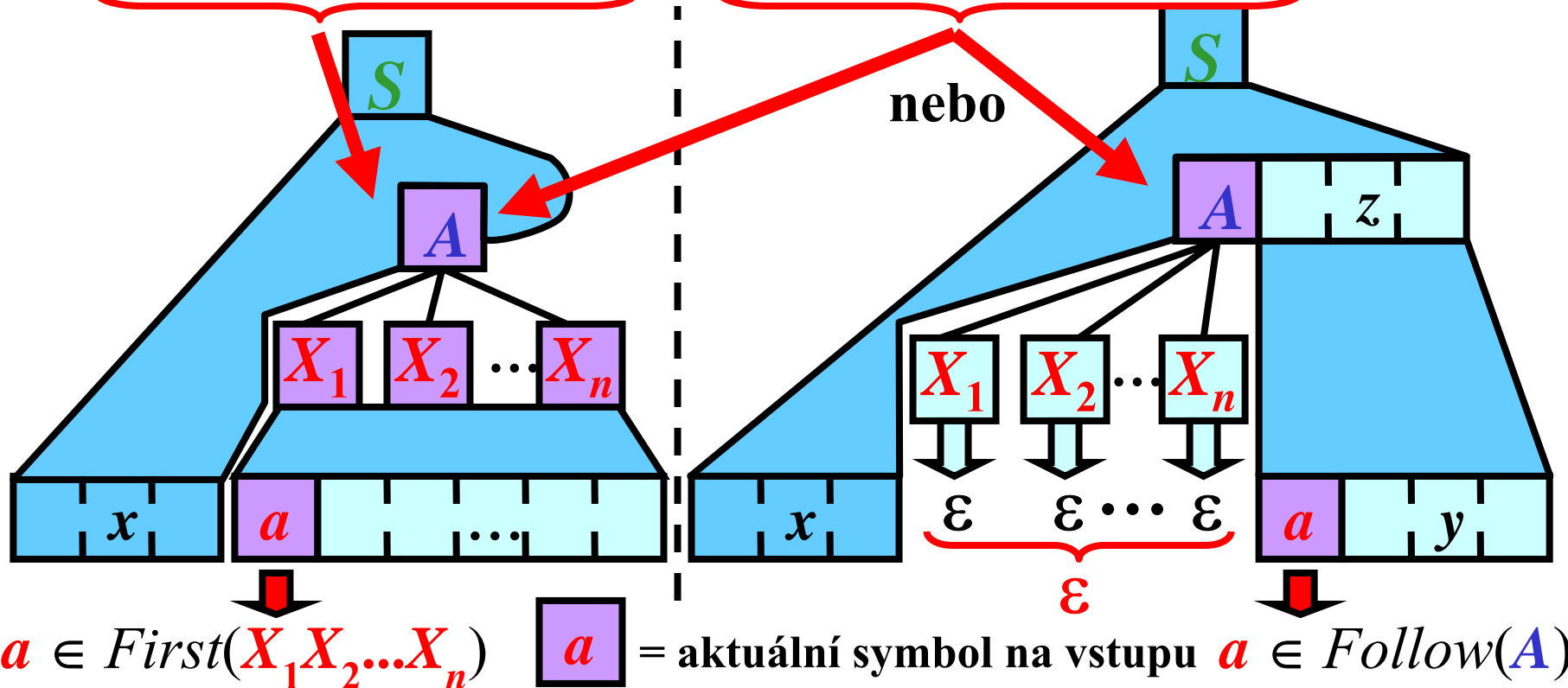
$$Predict(A \rightarrow x) = First(x) \cup Follow(A)$$

- jinak pokud $Empty(x) = \emptyset$ potom:

$$Predict(A \rightarrow x) = First(x)$$

Množina $Predict(A \rightarrow X_1X_2...X_n)$: Ilustrace

$Empty(X_1X_2...X_n) = \emptyset$ vs. $Empty(X_1X_2...X_n) = \{\epsilon\}$



Celkově: if $Empty(X_1X_2...X_n) = \{\epsilon\}$ then
 $Predict(A \rightarrow X_1X_2...X_n) = First(X_1X_2...X_n) \cup Follow(A)$;
else $Predict(A \rightarrow X_1X_2...X_n) = First(X_1X_2...X_n)$

Predict($A \rightarrow x$) pro G_{expr3} : Příklad 1/2

$First(\mathbf{E})$	$:= \{i, ()$	$Empty(\mathbf{E})$	$:= \emptyset$	$Follow(\mathbf{E})$	$:= \{\$,)\}$
$First(\mathbf{E}')$	$:= \{+\}$	$Empty(\mathbf{E}')$	$:= \{\varepsilon\}$	$Follow(\mathbf{E}')$	$:= \{\$,)\}$
$First(\mathbf{T})$	$:= \{i, ()$	$Empty(\mathbf{T})$	$:= \emptyset$	$Follow(\mathbf{T})$	$:= \{+, \$,)\}$
$First(\mathbf{T}')$	$:= \{*\}$	$Empty(\mathbf{T}')$	$:= \{\varepsilon\}$	$Follow(\mathbf{T}')$	$:= \{+, \$,)\}$
$First(\mathbf{F})$	$:= \{i, ()$	$Empty(\mathbf{F})$	$:= \emptyset$	$Follow(\mathbf{F})$	$:= \{*, +, \$,)\}$

1: $\mathbf{E} \rightarrow \mathbf{TE}'$

$Empty(\mathbf{TE}')$ = \emptyset , protože $Empty(\mathbf{T}) = \emptyset$

$Predict(\mathbf{1}) := First(\mathbf{TE}')$ = $First(\mathbf{T}) = \{i, ()$

2: $\mathbf{E}' \rightarrow +\mathbf{TE}'$

$Empty(+\mathbf{TE}')$ = \emptyset , protože $Empty(\mathbf{T}) = \emptyset$

$Predict(\mathbf{2}) := First(+\mathbf{TE}')$ = $First(+)$ = $\{+\}$

3: $\mathbf{E}' \rightarrow \varepsilon$

$Empty(\varepsilon) = \{\varepsilon\}$

$Predict(\mathbf{3}) := First(\varepsilon) \cup Follow(\mathbf{E}')$ = $\emptyset \cup \{\$,)\} = \{\$,)\}$

4: $\mathbf{T} \rightarrow \mathbf{FT}'$

$Empty(\mathbf{FT}')$ = \emptyset , protože $Empty(\mathbf{F}) = \emptyset$

$Predict(\mathbf{4}) := First(\mathbf{FT}')$ = $First(\mathbf{F}) = \{i, ()$

Predict($A \rightarrow x$) pro G_{expr3} : Příklad 2/2

$First(\mathbf{E}) := \{i, ($	$Empty(\mathbf{E}) := \emptyset$	$Follow(\mathbf{E}) := \{\$,)\}$
$First(\mathbf{E}') := \{+\}$	$Empty(\mathbf{E}') := \{\varepsilon\}$	$Follow(\mathbf{E}') := \{\$,)\}$
$First(\mathbf{T}) := \{i, ($	$Empty(\mathbf{T}) := \emptyset$	$Follow(\mathbf{T}) := \{+, \$,)\}$
$First(\mathbf{T}') := \{*\}$	$Empty(\mathbf{T}') := \{\varepsilon\}$	$Follow(\mathbf{T}') := \{+, \$,)\}$
$First(\mathbf{F}) := \{i, ($	$Empty(\mathbf{F}) := \emptyset$	$Follow(\mathbf{F}) := \{*, +, \$,)\}$

5: $\mathbf{T}' \rightarrow *FT'$

$Empty(*FT') = \emptyset$, protože $Empty(\mathbf{F}) = \emptyset$

$Predict(5) := First(*FT') = First(*) = \{*\}$

6: $\mathbf{T}' \rightarrow \varepsilon$

$Empty(\varepsilon) = \{\varepsilon\}$

$Predict(6) := First(\varepsilon) \cup Follow(\mathbf{T}') = \emptyset \cup \{+, \$,)\} = \{+, \$,)\}$

7: $\mathbf{F} \rightarrow (\mathbf{E})$

$Empty(\mathbf{E}) = \emptyset$, protože $Empty(\mathbf{E}) = \emptyset$

$Predict(7) := First(\mathbf{E}) = First(()) = \{($

8: $\mathbf{F} \rightarrow i$

$Empty(i) = \emptyset$

$Predict(8) := First(i) = \{i\}$

Konstrukce LL-tabulky

α	...	a	...
...			
A		$\alpha(A, a)$	
...			

$\alpha(A, a) = A \rightarrow X_1X_2\dots X_n \in P$ pokud $a \in \text{Predict}(A \rightarrow X_1X_2\dots X_n)$; jinak $\alpha(A, a)$ is je prázdné.

Určeme: LL tabulku pro G_{expr1}

	i	$+$	$*$	$($	$)$	$\$$
E	1					
E'						
T	4					
T'						
F	8					

$i \in \text{Predict}(1)$

$i \in \text{Predict}(4)$

$i \in \text{Predict}(8)$

Pravidlo r	$\text{Predict}(r)$
1: $E \rightarrow TE'$	$\{i, ($
2: $E' \rightarrow +TE'$	$\{+\}$
3: $E' \rightarrow \varepsilon$	$\{\$,)\}$
4: $T \rightarrow FT'$	$\{i, ($
5: $T' \rightarrow *FT'$	$\{*\}$
6: $T' \rightarrow \varepsilon$	$\{+, \$,)\}$
7: $F \rightarrow (E)$	$\{($
8: $F \rightarrow i$	$\{i\}$

Zbytek tabulky by se sestrojil analogicky.

SA založená na LL-tabulce: Příklad

	<i>i</i>	+	*	()	\$
<i>E</i>	1			1		
<i>E'</i>		2			3	3
<i>T</i>	4			4		
<i>T'</i>		6	5		6	6
<i>F</i>	8			7		

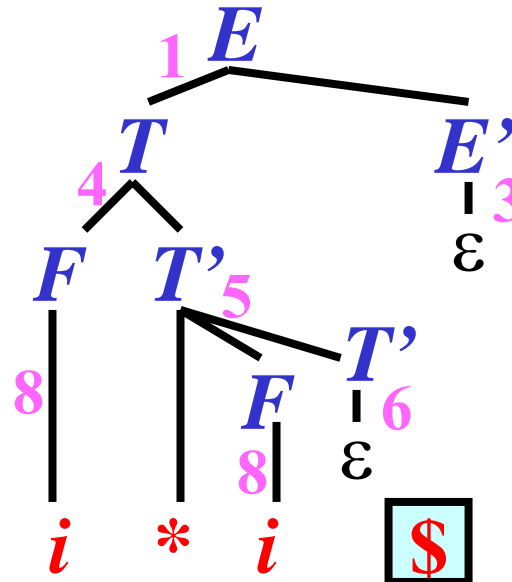
1: $E \rightarrow TE'$ 5: $T' \rightarrow *FT'$

2: $E' \rightarrow +TE'$ 6: $T' \rightarrow \varepsilon$

3: $E' \rightarrow \varepsilon$ 7: $F \rightarrow (E)$

4: $T \rightarrow FT'$ 8: $F \rightarrow i$

Otázka: $i * i \in L(G_{expr3})$?



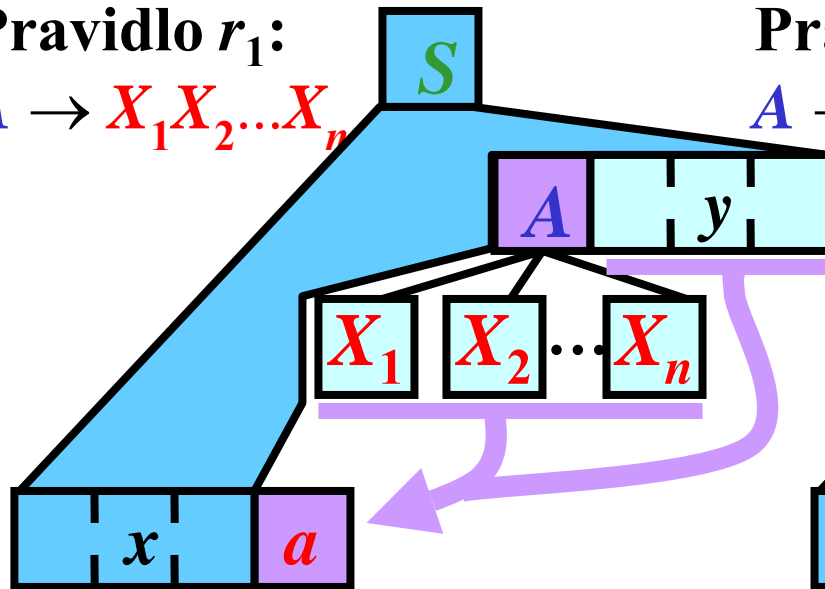
LL gramatiky s ε -pravidly: Definice

Definice: Necht' $G = (N, T, P, S)$ je BKG. G je *LL-gramatika*, pokud pro každé $a \in T$ a každé $A \in N$ existuje **maximálně jedno** A -pravidlo tvaru $A \rightarrow X_1X_2\dots X_n \in P$ a platí: $a \in \text{Predict}(A \rightarrow X_1X_2\dots X_n)$

Ilustrace:

Pravidlo r_1 :

$A \rightarrow X_1X_2\dots X_n$

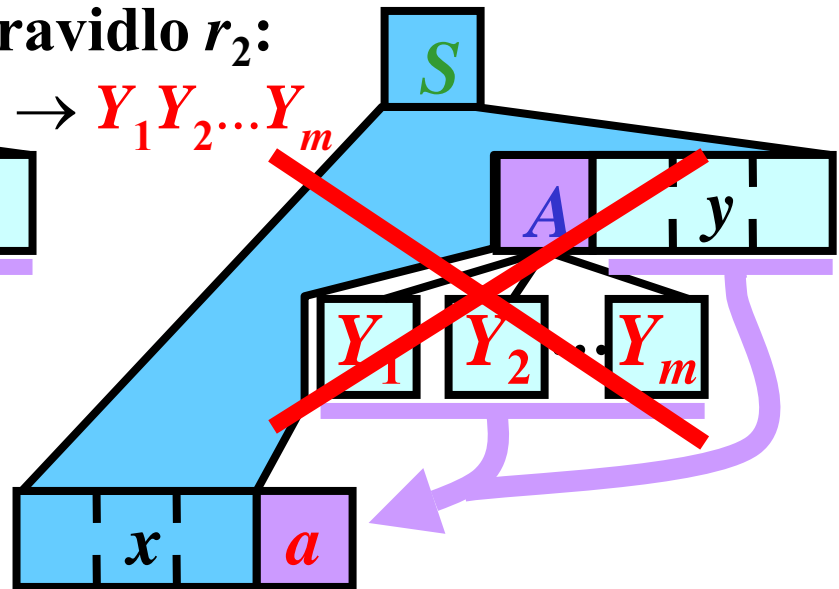


$a \in \text{Predict}(A \rightarrow X_1X_2\dots X_n)$

Nesmí nastat v LL-gramatice

Pravidlo r_2 :

$A \rightarrow Y_1Y_2\dots Y_m$

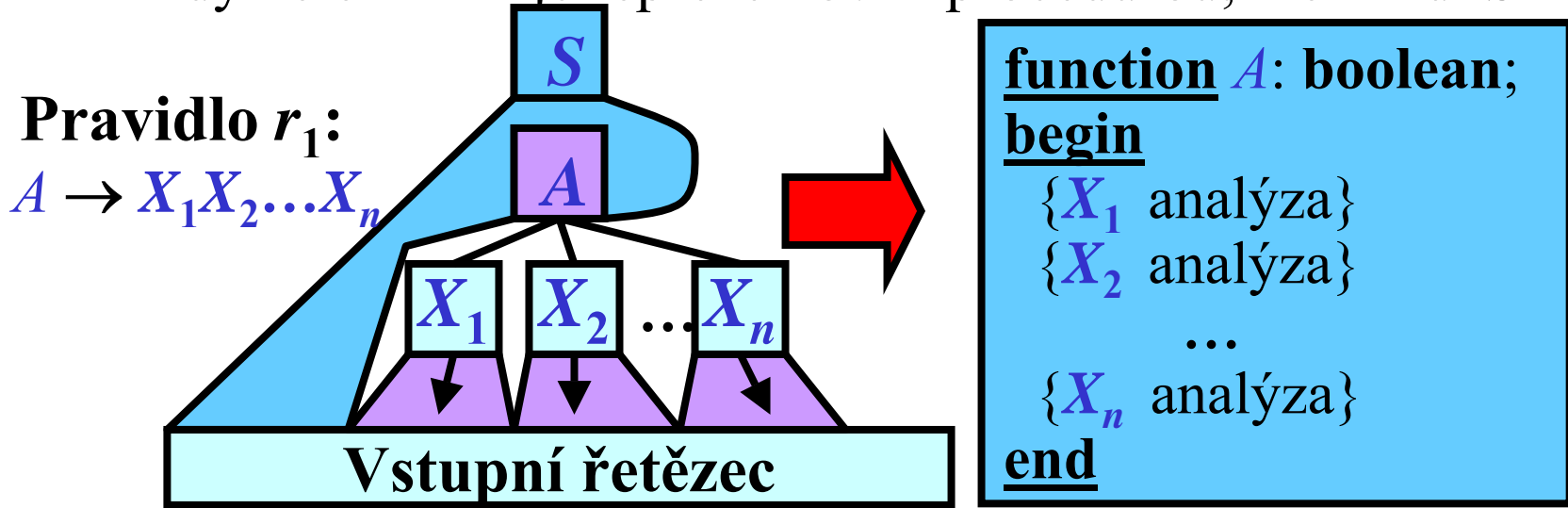


$a \in \text{Predict}(A \rightarrow Y_1Y_2\dots Y_m)$

Implementace LL Analyzátoru

1) Rekurzivní sestup

- Každý neterminál je reprezentován procedurou, která řídí SA:



2) Prediktivní syntaktická analýza

- Syntaktický analyzátor se zásobníkem řízený tabulkou



Právě tyto symboly v tomto pořadí jsou uloženy na zásobníku.

Rekurzivní sestup: Příklad 1/4

```

Procedure GetNextToken;
begin
  { tato procedura uloží následující token do proměnné "token" }
end

```

- Pro $E \in N$: Pravidlo 1: $E \rightarrow TE'$

```

function E: boolean;
begin
  E := false;
  if token in ['i', '('] then
    { simulace pravidla 1: E → TE' }
    E := T and E1;
end;

```

	<i>i</i>	+	*	()	\$
<i>E</i>	1			1		
<i>E'</i>		2			3	3
<i>T</i>	4			4		
<i>T'</i>		6	5		6	6
<i>F</i>	8			7		

- Pro $T \in N$: Pravidlo 4: $T \rightarrow FT'$

```

function T: boolean;
begin
  T := false;
  if token in ['i', '('] then
    { simulace pravidla 4: T → FT' }
    T := F and T1;
end;

```

	<i>i</i>	+	*	()	\$
<i>E</i>	1			1		
<i>E'</i>		2			3	3
<i>T</i>	4			4		
<i>T'</i>		6	5		6	6
<i>F</i>	8			7		

Rekurzivní sestup: Příklad 2/4

- Pro $E' \in N$: Pravidla 2: $E' \rightarrow +TE'$, 3: $E' \rightarrow \varepsilon$

```

function E1: boolean;
begin
  E1 := false;
  if token = '+' then begin
    { simulace pravidla 2:  $E' \rightarrow +TE'$  }
    GetNextToken;
    E1 := T and E1;
  end
  else
    if token in [')', '$'] then
      { simulace pravidla 3:  $E' \rightarrow \varepsilon$  }
      E1 := true;
    end;
end;

```

	i	+	*	()	\$
E	1			1		
E'		2			3	3
T	4			4		
T'		6	5		6	6
F	8			7		

2

3

Rekurzivní sestup: Příklad 3/4

- Pro $T' \in N$: Pravidla 5: $T' \rightarrow *FT'$, 6: $T' \rightarrow \varepsilon$

```

function T1: boolean;
begin
  T1 := false;
  if token = '*' then begin
    { simulace pravidla 5:  $T' \rightarrow *FT'$  }
    GetNextToken;
    T1 := F and T1;
  end
  else
    if token in ['+', ')', '$'] then
      { simulace pravidla 6:  $T' \rightarrow \varepsilon$  }
      T1 := true;
    end;
end;

```

	i	+	*	()	\$
E	1			1		
E'		2			3	3
T	4			4		
T'		6	5		6	6
F	8			7		

5

6

Rekurzivní sestup: Příklad 4/4

- Pro $F \in N$: Pravidla $7: F \rightarrow (E)$, $8: F \rightarrow i$
- ```

function F: boolean;
begin
 F := false;
 if token = '(' then begin
 { simulace pravidla 7: F → (E) }
 GetNextToken;
 if E then begin
 F := (token = ')');
 GetNextToken;
 end;
 end
 else
 if token = 'i' then begin
 { simulace pravidla 8: F → i }
 F := true;
 GetNextToken;
 end;
 end;
end;

```

|      | $i$ | $+$ | $*$ | $($ | $)$ | $\$$ |
|------|-----|-----|-----|-----|-----|------|
| $E$  | 1   |     |     | 1   |     |      |
| $E'$ |     | 2   |     |     | 3   | 3    |
| $T$  | 4   |     |     | 4   |     |      |
| $T'$ |     | 6   | 5   |     | 6   | 6    |
| $F$  | 8   |     |     | 7   |     |      |

```

Hlavní tělo programu:
begin
 GetNextToken;
 if E then
 write('OK')
 else
 write('ERROR')
end.

```

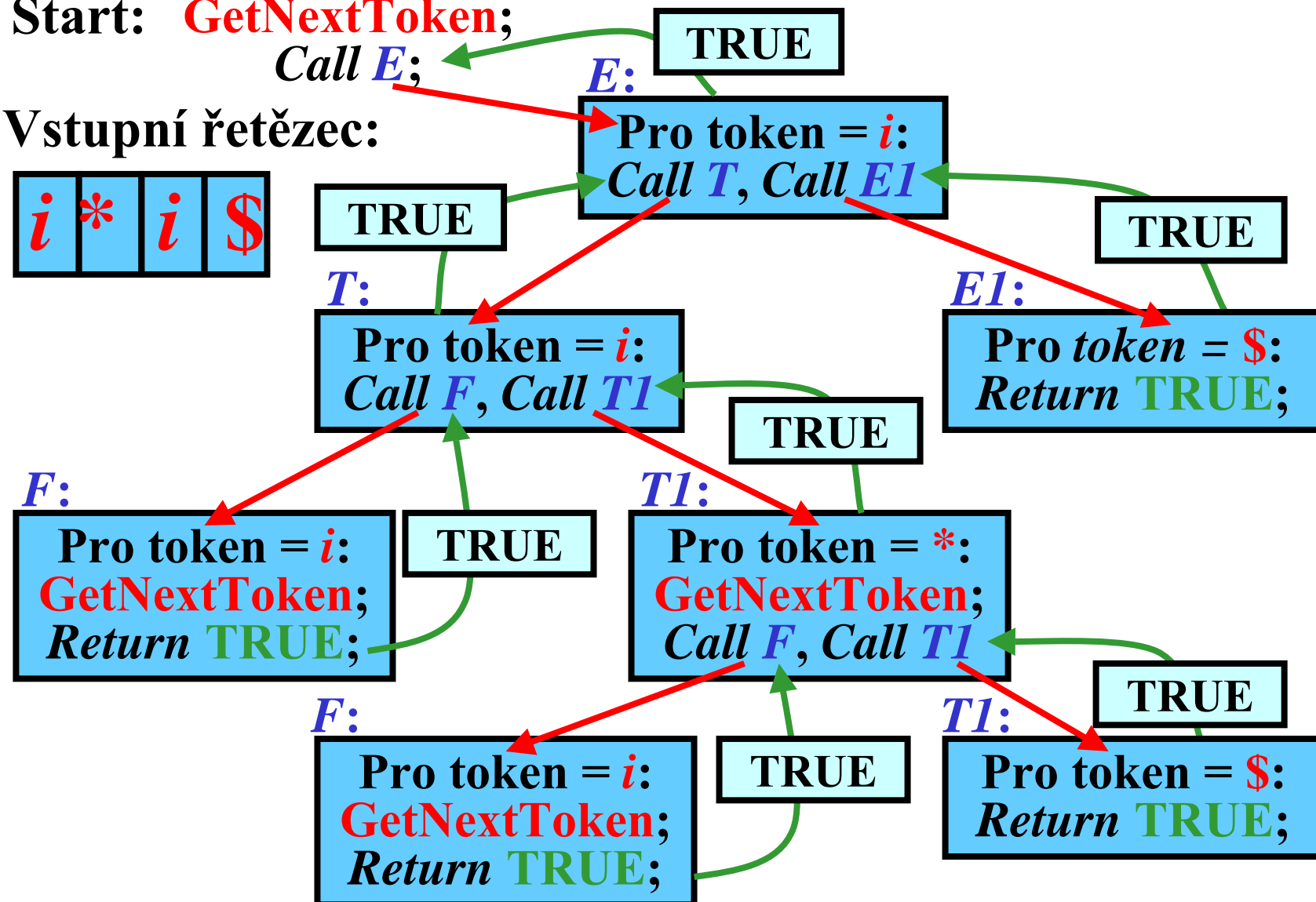
# Rekurzivní sestup: Ilustrace pro $i*i\$$

Start: **GetNextToken;**

*Call E;*

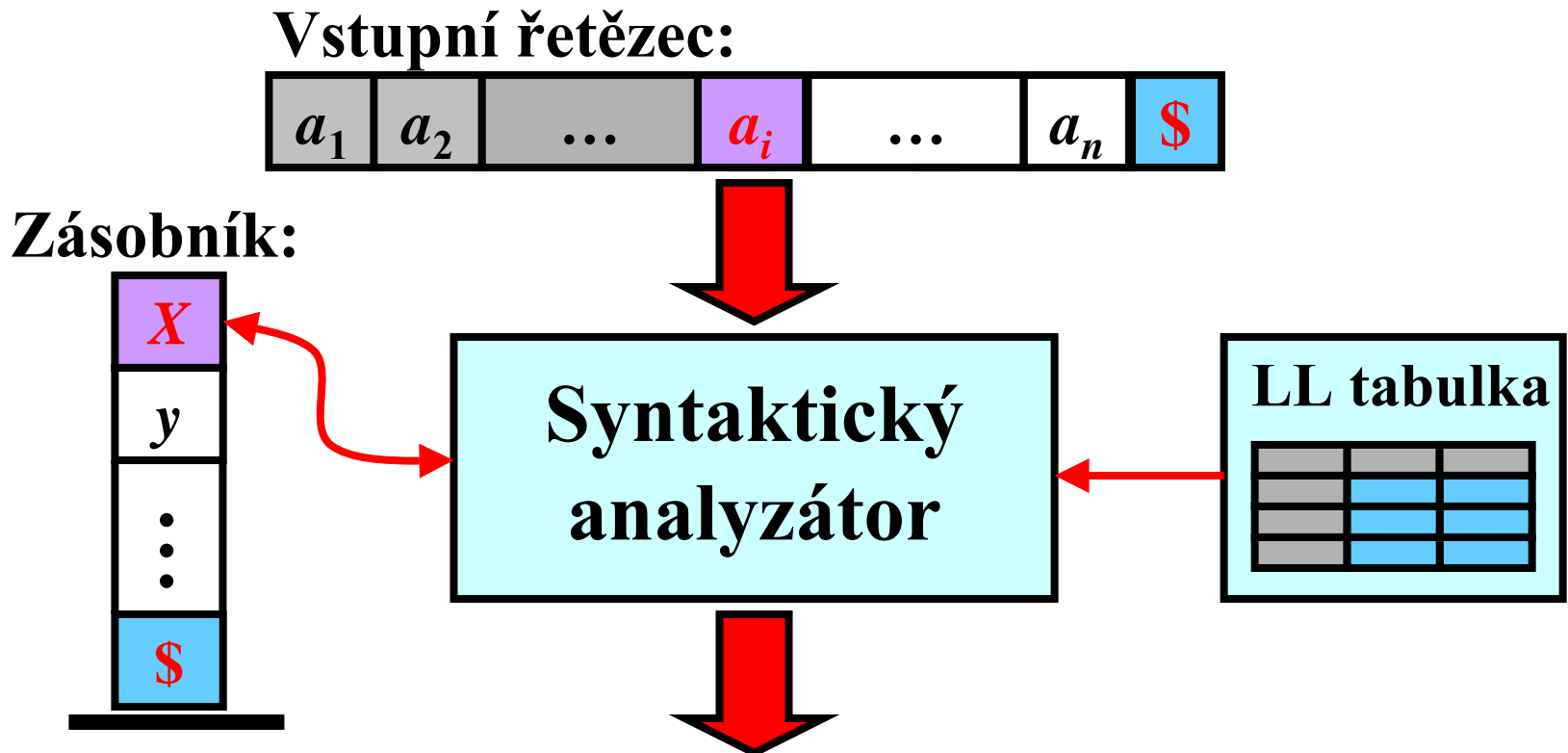
Vstupní řetězec:

|     |     |     |      |
|-----|-----|-----|------|
| $i$ | $*$ | $i$ | $\$$ |
|-----|-----|-----|------|



# Prediktivní syntaktická analýza

- Model pro prediktivní syntaktickou analýzu:



*Levý rozbor* = posloupnost pravidel, která je použita v nejlevější derivaci pro vstupní řetězec.

# Prediktivní SA: Algoritmus

- **Vstup:** LL-tabulka pro  $G=(N, T, P, S)$ ;  $x \in T^*$
- **Výstup:** Levý rozbor pro  $x$ , pokud  $x \in L(G)$  jinak chyba

## • Metoda:

- push(**\$**) & push(**S**) na zásobník

## • repeat

- necht'  $X$  je vrchol zásobníku a  $a$  aktuální token

## • case $X$ of:

- $X = \$$ : if  $a = \$$  then úspěch  
else chyba;

- $X \in T$ : if  $X = a$  then pop( $X$ ) & přečti další  $a$  ze vstupního řetězce  
else chyba;

- $X \in N$ : if  $r: X \rightarrow x \in \text{LL-tabulka}[X, a]$  then zaměň na vrcholu zásobníku  $X$  za reversal( $x$ ) & zapiš  $r$  na výstup  
else chyba;

until úspěch or chyba



# Prediktivní SA: Příklad

|           |          |   |   |   |   |    |
|-----------|----------|---|---|---|---|----|
|           | <i>i</i> | + | * | ( | ) | \$ |
| <i>E</i>  | 1        |   |   | 1 |   |    |
| <i>E'</i> |          | 2 |   |   | 3 | 3  |
| <i>T</i>  | 4        |   |   | 4 |   |    |
| <i>T'</i> |          | 6 | 5 |   | 6 | 6  |
| <i>F</i>  | 8        |   |   | 7 |   |    |

Vstupní řetězec: *i \* i \$*

Pravidla:

- 1:  $E \rightarrow TE'$
- 2:  $E' \rightarrow +TE'$
- 3:  $E' \rightarrow \varepsilon$
- 4:  $T \rightarrow FT'$
- 5:  $T' \rightarrow *FT'$
- 6:  $T' \rightarrow \varepsilon$
- 7:  $F \rightarrow (E)$
- 8:  $F \rightarrow i$

| Zásobník   | Vstup        | Pravidlo                        | Derivace                                    |
|------------|--------------|---------------------------------|---------------------------------------------|
| $\$E$      | <i>i*i\$</i> | 1: $E \rightarrow TE'$          | $\underline{E} \Rightarrow \underline{TE}'$ |
| $\$E'T$    | <i>i*i\$</i> | 4: $T \rightarrow FT'$          | $\Rightarrow \underline{FT}'E'$             |
| $\$E'T'F$  | <i>i*i\$</i> | 8: $F \rightarrow i$            | $\Rightarrow \underline{iT}'E'$             |
| $\$E'T'i$  | <i>i*i\$</i> |                                 |                                             |
| $\$E'T'$   | <i>*i\$</i>  | 5: $T' \rightarrow *FT'$        | $\Rightarrow i*\underline{FT}'E'$           |
| $\$E'T'F*$ | <i>*i\$</i>  |                                 |                                             |
| $\$E'T'F$  | <i>i\$</i>   | 8: $F \rightarrow i$            | $\Rightarrow i*\underline{iT}'E'$           |
| $\$E'T'i$  | <i>i\$</i>   |                                 |                                             |
| $\$E'T'$   | $\$$         | 6: $T' \rightarrow \varepsilon$ | $\Rightarrow i*i\underline{E}'$             |
| $\$E'$     | $\$$         | 3: $E' \rightarrow \varepsilon$ | $\Rightarrow i*i$                           |
| $\$$       | $\$$         |                                 |                                             |

Úspěch

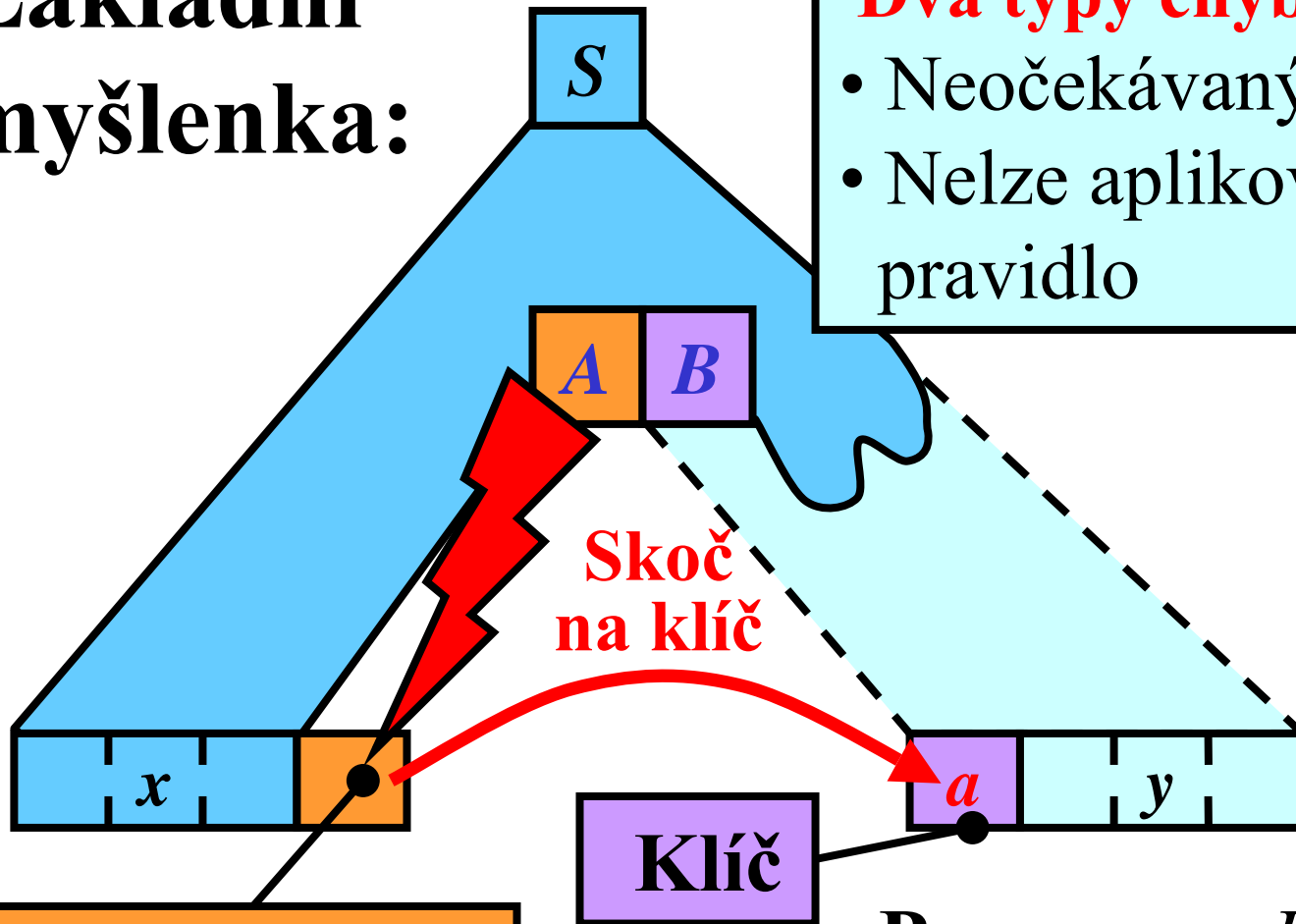
Levý rozbor: 1485863

# Zotavenení z chyb: Úvod

**Základní  
myšlenka:**

**Dva typy chyb:**

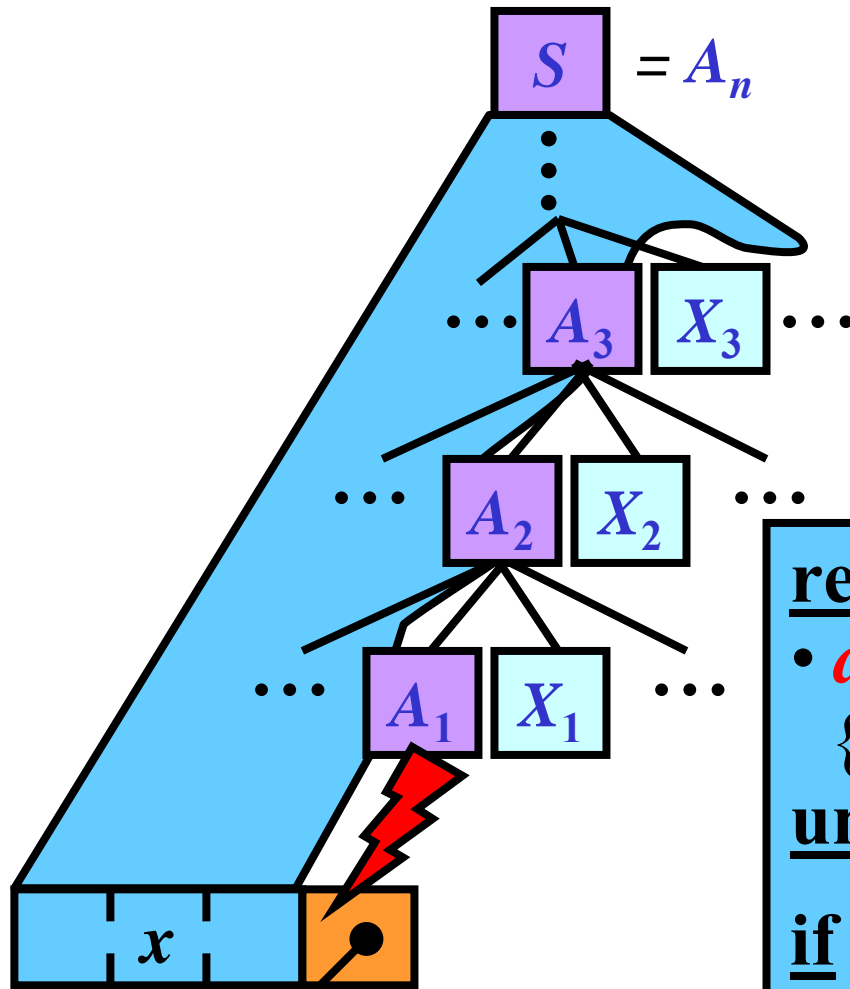
- Neočekávaný token
- Nelze aplikovat pravidlo



**Chybný token**

**Pozn.:  $a \in \text{Follow}(A)$**

# Hartmannova metoda: Zotavení z chyb



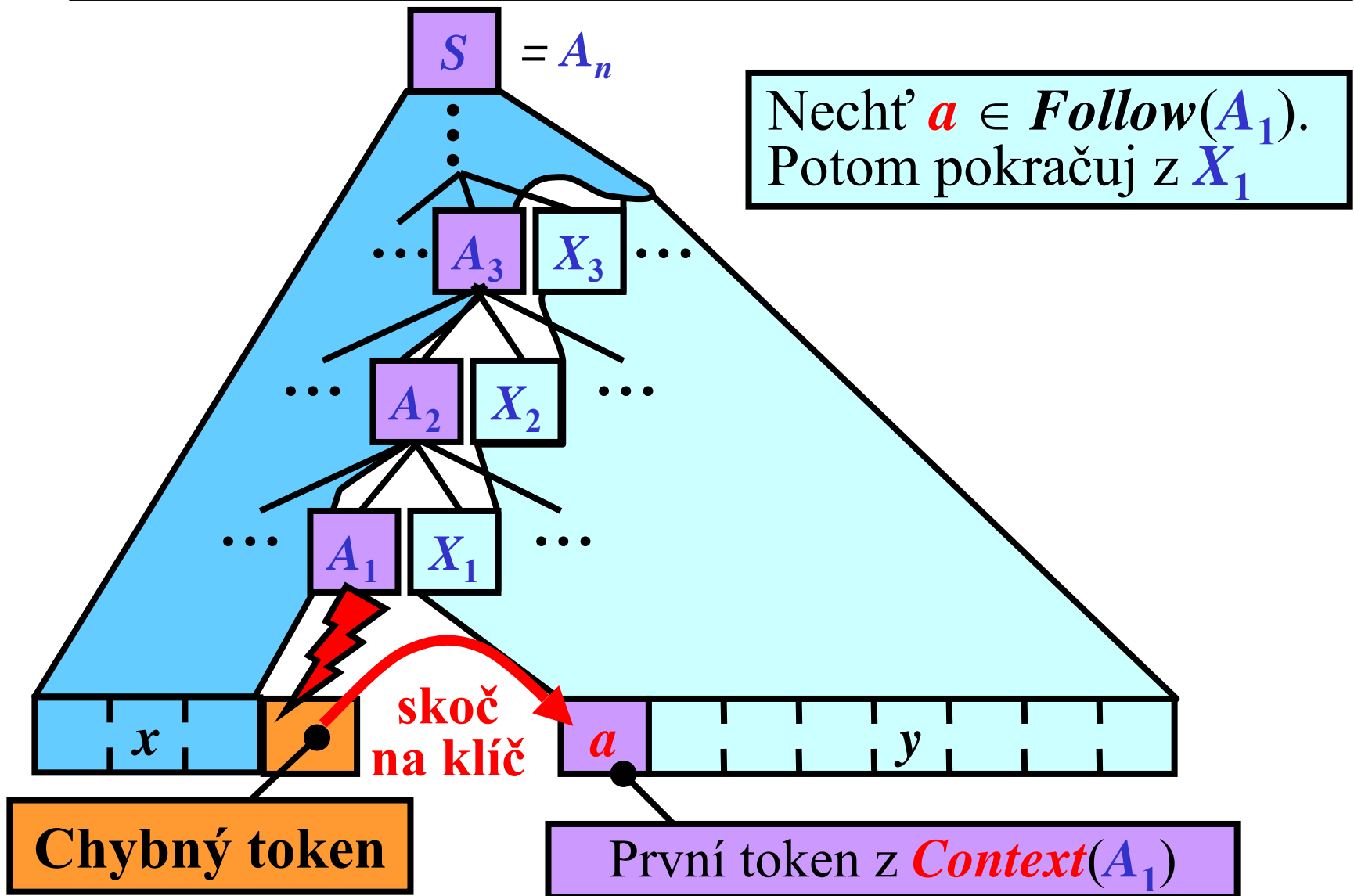
- Necht' **Context**( $A_1$ ) =  
 $Follow(A_1) \cup$   
 $Follow(A_2) \cup$   
 $\dots$   
 $Follow(A_n)$

## repeat

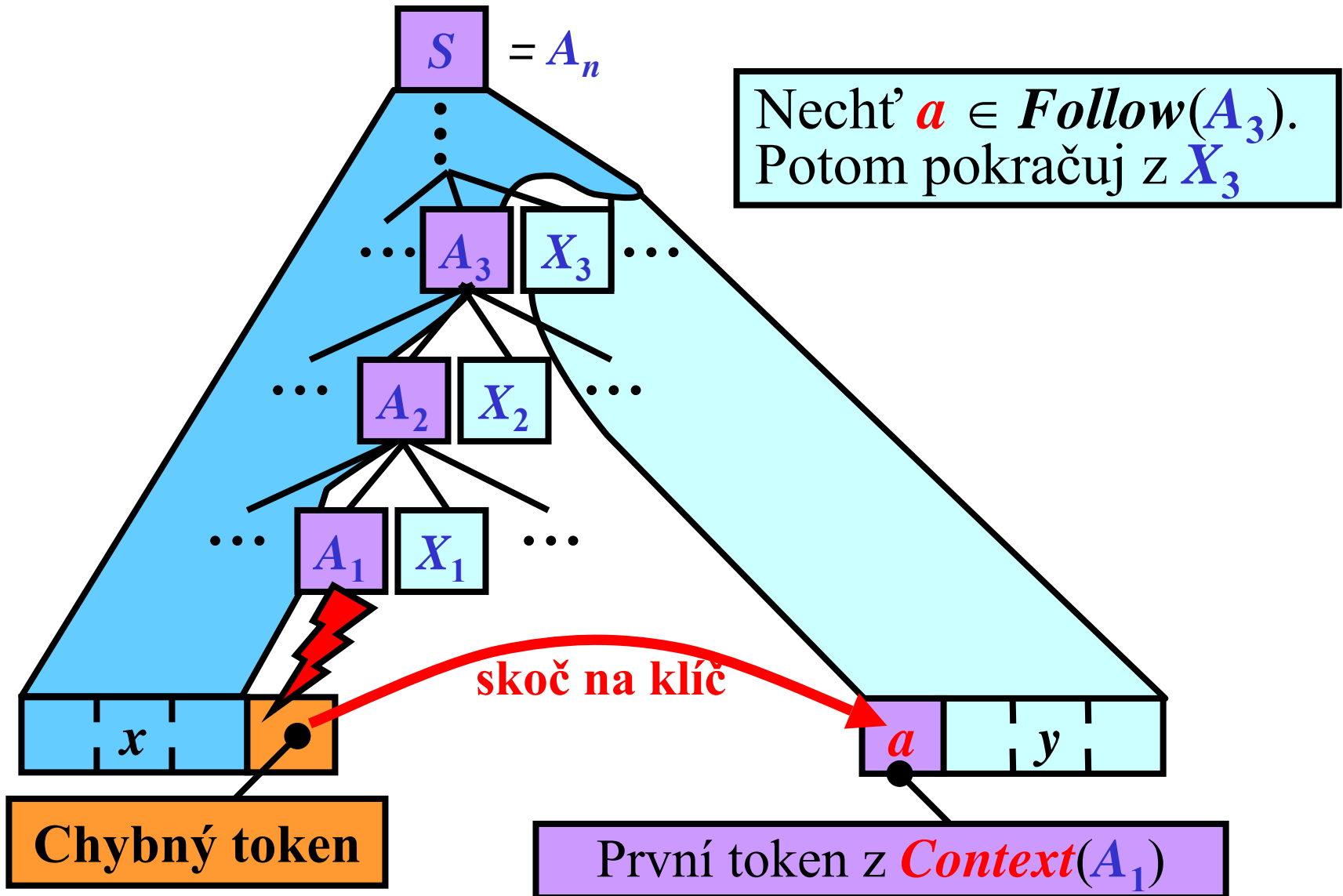
- $a := \text{GetNextToken};$   
 {Tyto tokeny přeskoč}
- until  $a$  v množ. **Context**( $A_1$ )
- if  $a$  v množ. **Follow**( $A_i$ ) then  
 pokračuj v syntaktické  
 analýze od symbolu  $X_i$ .

Chybný token

# Zotavení z chyb: Ilustrace 1/2



# Zotavení z chyb: Ilustrace 2/2



## *Context(X)* pro prediktivní SA: Varianta I

Pro  $G = (N, T, P, S)$ ,

***Context(A)*** = *Follow(A)* pro všechna  $A \in N$

---

- **Metoda:**

- Necht'  $A$  je vrchol zásobníku & žádné pravidlo nelze použít:

- **repeat**

  - $a := \text{GetNextToken};$

  - {Tyto tokeny jsou přeskočeny}

  - **until**  $a$  v množině ***Context(A)***

- odstraň  $A$  ze zásobníku;



## *Context(X)* pro prediktivní SA: Varianta II

Pro  $G = (N, T, P, S)$ ,

***Context(A)*** = *First(A)*  $\cup$  *Follow(A)* pro všechna  $A \in N$

---

- **Metoda:**

- Necht'  $A$  je vrchol zásobníku & žádné pravidlo nelze použít:

- **repeat**

- $a := \text{GetNextToken};$

- {Tyto tokeny jsou přeskočeny}

- **until**  $a$  v množině ***Context(A)***

- **if**  $a \in \text{First}(A)$  **then** ponech symbol  $A$  na zásobníku  
**else** odstraň  $A$  ze zásobníku; //  $a \in \text{Follow}(A)$



