

Cvičenie - Syntaktická analýza top-down, LL(1)

Ing. Viliam Hromada, PhD.

C-510
Ústav informatiky a matematiky
FEI STU

`viliam.hromada@stuba.sk`

Príklad č. 1

Je daná bezkontextová gramatika $G = (N, T, P, S)$, $N = \{S, A, B\}$, $T = \{a, b, c\}$, S je počiatočný neterminál. Pravidlá:

1. $S \rightarrow ABC$
2. $A \rightarrow a$
3. $A \rightarrow \varepsilon$
4. $B \rightarrow b$
5. $B \rightarrow \varepsilon$

Úlohy:

1. Zostrojte rozkladovú tabuľku $LL(1)$ -analyzátoru a na jej základe určte, či sa jedná o $LL(1)$ -gramatiku.
2. Ak áno, zistite, či majú deriváciu nasledovné reťazce: abc , ac , c , aac , aba , ca , ab .



Najprv vypočítame množiny *FIRST* a *FOLLOW* pre neterminály:

	<i>S</i>	<i>A</i>	<i>B</i>
<i>FIRST</i>	$\{a, b, c\}$	$\{a, \varepsilon\}$	$\{b, \varepsilon\}$
<i>FOLLOW</i>	$\{\varepsilon\}$	$\{b, c\}$	$\{c\}$

Následne určíme množiny *PREDICT* pre jednotlivé pravidlá:

1. $PREDICT(S \rightarrow ABc) = \{a, b, c\}$. Zdôvodnenie:

- $FIRST(ABc) = \{a, b, c\}$. A keďže $\varepsilon \notin FIRST(ABc)$, tak $PREDICT(S \rightarrow ABc) = FIRST(ABc)$.

2. $PREDICT(A \rightarrow a) = \{a\}$. Zdôvodnenie:

- $FIRST(a) = \{a\}$. A keďže $\varepsilon \notin FIRST(a)$, tak $PREDICT(A \rightarrow a) = FIRST(a)$.

3. $PREDICT(A \rightarrow \varepsilon) = \{b, c\}$. Zdôvodnenie:

- $FIRST(\varepsilon) = \{\varepsilon\}$. A keďže $\varepsilon \in FIRST(\varepsilon)$, tak $PREDICT(A \rightarrow \varepsilon) = (FIRST(\varepsilon) \setminus \{\varepsilon\}) \cup FOLLOW(A) = \emptyset \cup \{b, c\} = \{b, c\}$.



4. $PREDICT(B \rightarrow b) = \{b\}$. Zdôvodnenie:
- $FIRST(b) = \{b\}$. A keďže $\varepsilon \notin FIRST(b)$, tak $PREDICT(B \rightarrow b) = FIRST(b)$.
5. $PREDICT(B \rightarrow \varepsilon) = \{c\}$. Zdôvodnenie:
- $FIRST(\varepsilon) = \{\varepsilon\}$. A keďže $\varepsilon \in FIRST(\varepsilon)$, tak
 $PREDICT(B \rightarrow \varepsilon) = (FIRST(\varepsilon) \setminus \{\varepsilon\}) \cup FOLLOW(B) = \emptyset \cup \{c\} = \{c\}$.



Na základe množín *PREDICT* jednotlivých pravidiel gramatiky vieme vyplniť rozkladovú tabuľku:

RT	<i>a</i>	<i>b</i>	<i>c</i>	ϵ
<i>S</i>	1	1	1	
<i>A</i>	2	3	3	
<i>B</i>		4	5	

Keďže v rozkladovej tabuľke sa v každej bunke nachádza **maximálne jedno pravidlo**, gramatika **je** *LL(1)* gramatikou a *LL(1)* analyzátor **sa dá použiť**.

abc

Syntaktická analýza reťazca *abc*:

r.	Zvyšok vstupu	Zásobník □	Akcia
1	<i>abc</i>	<i>S</i>	E1
2	<i>abc</i>	<i>ABc</i>	E2
3	<i>abc</i>	<i>aBc</i>	P
4	<i>bc</i>	<i>Bc</i>	E4
5	<i>bc</i>	<i>bc</i>	P
6	<i>c</i>	<i>c</i>	P
7	ϵ	ϵ	A

Vstup bol celý prečítaný a zásobník celý vyprázdnený - reťazec *abc* teda gramatika **generuje**. Jeho odvodenie ľavou deriváciou vznikne aplikáciou pravidiel č. 1, 2, 4, v tomto poradí.

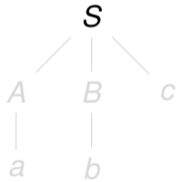


abc

Derivácia reťazca *abc*:

$$S \Rightarrow ABc \Rightarrow aBc \Rightarrow abc$$

Derivačný strom:

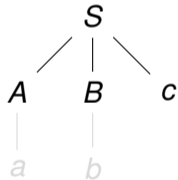


abc

Derivácia reťazca *abc*:

$$S \Rightarrow ABc \Rightarrow aBc \Rightarrow abc$$

Derivačný strom:

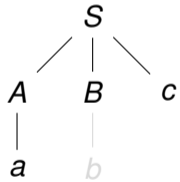


abc

Derivácia reťazca *abc*:

$$S \Rightarrow ABc \Rightarrow aBc \Rightarrow abc$$

Derivačný strom:

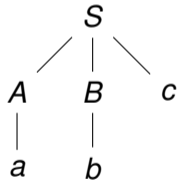


abc

Derivácia reťazca *abc*:

$$S \Rightarrow ABc \Rightarrow aBc \Rightarrow abc$$

Derivačný strom:



ac

Syntaktická analýza reťazca *ac*:

r.	Zvyšok vstupu	Zásobník □	Akcia
1	<i>ac</i>	<i>S</i>	E1
2	<i>ac</i>	<i>ABc</i>	E2
3	<i>ac</i>	<i>aBc</i>	P
4	<i>c</i>	<i>Bc</i>	E5
5	<i>c</i>	<i>c</i>	P
6	ε	ε	A

Vstup bol celý prečítaný a zásobník celý vyprázdnený - reťazec *ac* teda gramatika **generuje**. Jeho odvodenie ľavou deriváciou vznikne aplikáciou pravidiel č. 1, 2, 5, v tomto poradí.

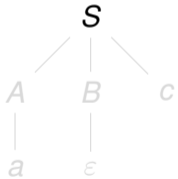


ac

Derivácia reťazca ac :

$$S \Rightarrow ABc \Rightarrow aBc \Rightarrow ac$$

Derivačný strom:

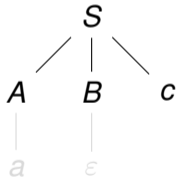


ac

Derivácia reťazca ac :

$$S \Rightarrow ABc \Rightarrow aBc \Rightarrow ac$$

Derivačný strom:

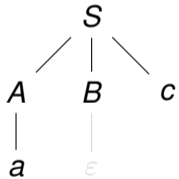


ac

Derivácia reťazca ac :

$$S \Rightarrow ABc \Rightarrow aBc \Rightarrow ac$$

Derivačný strom:

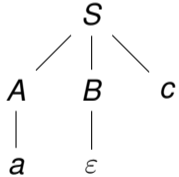


ac

Derivácia reťazca ac :

$$S \Rightarrow ABc \Rightarrow aBc \Rightarrow ac$$

Derivačný strom:



C

Syntaktická analýza reťazca c :

r.	Zvyšok vstupu	Zásobník \square	Akcia
1	c	S	E1
2	c	ABc	E3
3	c	Bc	E5
4	c	c	P
5	ε	ε	A

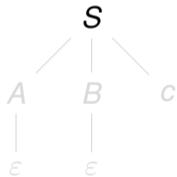
Vstup bol celý prečítaný a zásobník celý vyprázdnený - reťazec c teda gramatika **generuje**. Jeho odvodenie ľavou deriváciou vznikne aplikáciou pravidiel č. 1, 3, 5, v tomto poradí.

c

Derivácia reťazca *c*:

$$S \Rightarrow ABc \Rightarrow Bc \Rightarrow c$$

Derivačný strom:

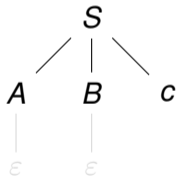


c

Derivácia reťazca *c*:

$$S \Rightarrow ABc \Rightarrow Bc \Rightarrow c$$

Derivačný strom:

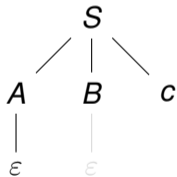


c

Derivácia reťazca *c*:

$$S \Rightarrow ABc \Rightarrow Bc \Rightarrow c$$

Derivačný strom:

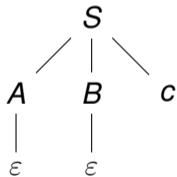


c

Derivácia reťazca *c*:

$$S \Rightarrow ABc \Rightarrow Bc \Rightarrow c$$

Derivačný strom:



aac

Syntaktická analýza reťazca *aac*:

r.	Zvyšok vstupu	Zásobník \square	Akcia
1	<i>aac</i>	<i>S</i>	E1
2	<i>aac</i>	<i>ABc</i>	E2
3	<i>aac</i>	<i>aBc</i>	P
4	<i>ac</i>	<i>Bc</i>	ERROR

Vidíme, že nastala situácia, že by sa mala vykonať expanzia pre neterminál *B*. Avšak v rozkladovej tabuľke **neexistuje** na pozícii [*B*, *a*] žiadne číslo pravidla. Teda v gramatike nie je možné expanziou neterminálu *B* získať na vrchu zásobníka symbol *a*. Nastala syntaktická chyba a teda reťazec *aac* v danej gramatike **nemá deriváciu**.



aba

Syntaktická analýza reťazca *aba*:

r.	Zvyšok vstupu	Zásobník □	Akcia
1	<i>aba</i>	<i>S</i>	E1
2	<i>aba</i>	<i>ABc</i>	E2
3	<i>aba</i>	<i>aBc</i>	P
4	<i>ba</i>	<i>Bc</i>	E4
5	<i>ba</i>	<i>bc</i>	P
6	<i>a</i>	<i>c</i>	ERROR

Vidíme, že nastala situácia, že by sa malo vykonať porovnanie, lebo na vrchu zásobníka je terminál *c*. Avšak terminál na vstupe *a* je **iný** než terminál na vrchu zásobníka *c*. Ich porovnanie teda **nemožno vykonať** a došlo k syntaktickej chybe, ktorá znamená, že reťazec *aba* **nemá v gramatike deriváciu**.



ca

Syntaktická analýza reťazca *ca*:

r.	Zvyšok vstupu	Zásobník \square	Akcia
1	<i>ca</i>	<i>S</i>	E1
2	<i>ca</i>	<i>ABc</i>	E3
3	<i>ca</i>	<i>Bc</i>	E5
4	<i>ca</i>	<i>c</i>	P
5	<i>a</i>	ϵ	ERROR

Vidíme, že nastala situácia, že zásobník bol **celý vyprázdnený**, avšak na vstupe zostala ešte **neprečítaná časť** - terminál *a*. Nastala syntaktická chyba, ktorá znamená, že reťazec *ca* **nemá v gramatike deriváciu**.



ab

Syntaktická analýza reťazca *ab*:

r.	Zvyšok vstupu	Zásobník □	Akcia
1	<i>ab</i>	<i>S</i>	E1
2	<i>ab</i>	<i>ABc</i>	E2
3	<i>ab</i>	<i>aBc</i>	P
4	<i>b</i>	<i>Bc</i>	E4
5	<i>b</i>	<i>bc</i>	P
6	ϵ	<i>c</i>	ERROR

Vidíme, že nastala situácia, že vstup bol **celý prečítaný**, avšak v zásobníku ešte zostali nejaké symboly, ktoré sa z neho **nedajú odstrániť**. Nastala teda syntaktická chyba, ktorá znamená, že reťazec *ab* **nemá v gramatike deriváciu**.



Príklad č. 2

Je daná bezkontextová gramatika $G = (N, T, P, S)$, $N = \{S, A, C\}$, $T = \{a, b\}$, S je počiatkový neterminál. Pravidlá:

1. $S \rightarrow aAb$
2. $A \rightarrow Ca$
3. $C \rightarrow aA$
4. $C \rightarrow aS$
5. $C \rightarrow b$

Úlohy:

1. Zostrojte rozkladovú tabuľku $LL(1)$ -analyzátoru a na jej základe určte, či sa jedná o $LL(1)$ -gramatiku.



Najprv vypočítame množiny *FIRST* a *FOLLOW* pre neterminály:

	<i>S</i>	<i>A</i>	<i>C</i>
<i>FIRST</i>	{ <i>a</i> }	{ <i>a, b</i> }	{ <i>a, b</i> }
<i>FOLLOW</i>	{ ϵ , <i>a</i> }	{ <i>a, b</i> }	{ <i>a</i> }

Následne určíme množiny *PREDICT* pre jednotlivé pravidlá:

1. $PREDICT(S \rightarrow aAb) = \{a\}$. Zdôvodnenie:
 - $FIRST(aAb) = \{a\}$. A keďže $\varepsilon \notin FIRST(aAb)$, tak $PREDICT(S \rightarrow aAb) = FIRST(aAb)$.
2. $PREDICT(A \rightarrow Ca) = \{a, b\}$. Zdôvodnenie:
 - $FIRST(Ca) = \{a, b\}$. A keďže $\varepsilon \notin FIRST(Ca)$, tak $PREDICT(A \rightarrow Ca) = FIRST(Ca)$.
3. $PREDICT(C \rightarrow aA) = \{a\}$. Zdôvodnenie:
 - $FIRST(aA) = \{a\}$. A keďže $\varepsilon \notin FIRST(aA)$, tak $PREDICT(A \rightarrow aA) = FIRST(aA)$.



4. $PREDICT(C \rightarrow aS) = \{a\}$. Zdôvodnenie:

- $FIRST(aS) = \{a\}$. A keďže $\varepsilon \notin FIRST(aS)$, tak $PREDICT(C \rightarrow aS) = FIRST(aS)$.

5. $PREDICT(C \rightarrow b) = \{b\}$. Zdôvodnenie:

- $FIRST(b) = \{b\}$. A keďže $\varepsilon \notin FIRST(b)$, tak $PREDICT(C \rightarrow b) = FIRST(b)$.



Na základe množín *PREDICT* jednotlivých pravidiel gramatiky vieme vyplniť rozkladovú tabuľku:

RT	<i>a</i>	<i>b</i>	ϵ
<i>S</i>	1		
<i>A</i>	2	2	
<i>C</i>	3,4	5	

Keďže v rozkladovej tabuľke sa nachádza bunka s **viac ako jedným pravidlom**, konkrétne na pozícii [*C*, *a*] gramatika **nie je** *LL*(1) gramatikou a *LL*(1) analyzátor **sa nedá použiť**.



Príklad č. 3

Je daná bezkontextová gramatika $G = (N, T, P, S)$, $S = \langle \text{príkazy} \rangle$ je počiatočný neterminál. Pravidlá:

1. $\langle \text{príkazy} \rangle \rightarrow \langle \text{príkaz} \rangle \langle \text{príkazy} \rangle$
2. $\langle \text{príkazy} \rangle \rightarrow \varepsilon$
3. $\langle \text{príkaz} \rangle \rightarrow \mathbf{if} \langle \text{podmienka} \rangle \mathbf{then} \langle \text{príkazy} \rangle \langle \text{elseČasť} \rangle \mathbf{fi}$
4. $\langle \text{príkaz} \rangle \rightarrow \mathbf{p1}$
5. $\langle \text{príkaz} \rangle \rightarrow \mathbf{p2}$
6. $\langle \text{elseČasť} \rangle \rightarrow \mathbf{else} \langle \text{príkazy} \rangle$
7. $\langle \text{elseČasť} \rangle \rightarrow \varepsilon$
8. $\langle \text{podmienka} \rangle \rightarrow \mathbf{podmienka}$

Terminály $T = \{\mathbf{if}, \mathbf{then}, \mathbf{else}, \mathbf{podmienka}, \mathbf{p1}, \mathbf{p2}, \mathbf{fi}\}$, ostatné symboly sú neterminály.

Zostrojte rozkladovú tabuľku $LL(1)$ -analyzátor a na jej základe určte, či sa jedná o $LL(1)$ -gramatiku.

Najprv vypočítame množiny *FIRST* a *FOLLOW* pre neterminály:

	<príkazy>	<príkaz>	<podmienka>	<elseČasť>
<i>FIRST</i>	{ ϵ , if , p1 , p2 }	{ if , p1 , p2 }	{ podmienka }	{ ϵ , else }
<i>FOLLOW</i>	{ ϵ , else , fi }	{ ϵ , if , p1 , p2 , else , fi }	{ then }	{ fi }

Následne určíme množiny *PREDICT* pre jednotlivé pravidlá:

1. $PREDICT(\langle \text{príkazy} \rangle \rightarrow \langle \text{príkaz} \rangle \langle \text{príkazy} \rangle) = \{\mathbf{if}, \mathbf{p1}, \mathbf{p2}\}$.
2. $PREDICT(\langle \text{príkazy} \rangle \rightarrow \varepsilon) = \{\mathbf{else}, \mathbf{fi}, \varepsilon\}$.
3. $PREDICT(\langle \text{príkaz} \rangle \rightarrow \mathbf{if} \langle \text{podmienka} \rangle \mathbf{then} \langle \text{príkazy} \rangle \langle \text{elseČasť} \rangle \mathbf{fi}) = \{\mathbf{if}\}$.
4. $PREDICT(\langle \text{príkaz} \rangle \rightarrow \mathbf{p1}) = \{\mathbf{p1}\}$.
5. $PREDICT(\langle \text{príkaz} \rangle \rightarrow \mathbf{p2}) = \{\mathbf{p2}\}$.
6. $PREDICT(\langle \text{elseČasť} \rangle \rightarrow \mathbf{else} \langle \text{príkazy} \rangle) = \{\mathbf{else}\}$.
7. $PREDICT(\langle \text{elseČasť} \rangle \rightarrow \varepsilon) = \{\mathbf{fi}\}$.
8. $PREDICT(\langle \text{podmienka} \rangle \rightarrow \mathbf{podmienka}) = \{\mathbf{podmienka}\}$.



Na základe množín *PREDICT* jednotlivých pravidiel gramatiky vieme vyplniť rozkladovú tabuľku:

RT	if	then	else	p1	p2	podmienka	fi	ϵ
<príkazy>	1		2	1	1		2	2
<príkaz>	3			4	5			
<elseČasť>			6				7	
<podmienka>						8		

Keďže v rozkladovej tabuľke sa nachádza vždy bunka s **maximálne jedným pravidlom**, gramatika **je** *LL(1)* gramatikou a *LL(1)* analyzátor **sa dá použiť**. Tak ho použijeme napr. na zistenie, či je reťazec

if podmienka then if podmienka then p1 fi else p1 fi

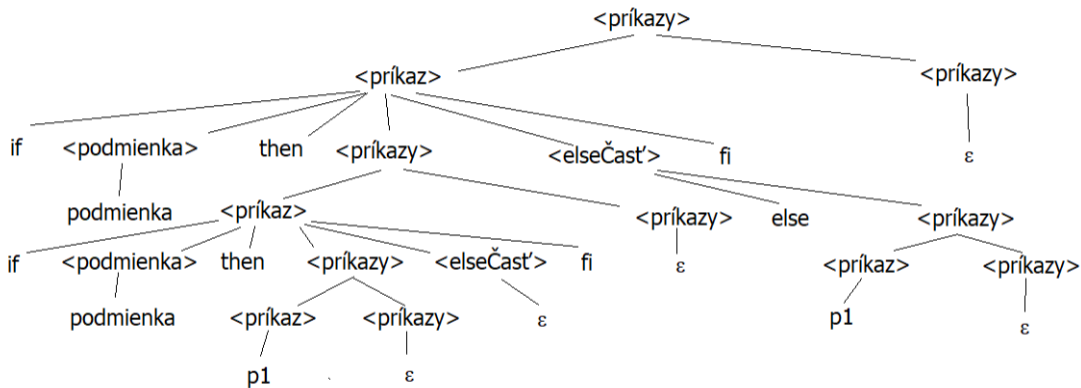
syntakticky v poriadku a ak áno, ako vyzerá jeho derivačný strom.

r.	Zvyšok vstupu	Zásobník	Akcia
1	if podmienka ...	<príkazy>	E1
2	if podmienka ...	<príkaz><príkazy>	E3
3	if podmienka ...	if <podmienka> then <príkazy><else Časť> fi <príkazy>	P
4	podmienka then ...	<podmienka> then <príkazy><else Časť> fi <príkazy>	E8
5	podmienka then ...	podmienka then <príkazy><else Časť> fi <príkazy>	P
6	then if ...	then <príkazy><else Časť> fi <príkazy>	P
7	if podmienka ...	<príkazy><else Časť> fi <príkazy>	E1
8	if podmienka ...	<príkaz><príkazy><else Časť> fi <príkazy>	E3
9	if podmienka ...	if <podmienka> then <príkazy><else Časť> fi <príkazy><else Časť> fi <príkazy>	P
10	podmienka then ...	<podmienka> then <príkazy><else Časť> fi <príkazy><else Časť> fi <príkazy>	E8
11	podmienka then ...	podmienka then <príkazy><else Časť> fi <príkazy><else Časť> fi <príkazy>	P
12	then p1 ...	then <príkazy><else Časť> fi <príkazy><else Časť> fi <príkazy>	P
13	p1 fi else ...	<príkazy><else Časť> fi <príkazy><else Časť> fi <príkazy>	E1
14	p1 fi else ...	<príkaz><príkazy><else Časť> fi <príkazy><else Časť> fi <príkazy>	E4
15	p1 fi else ...	p1 <príkazy><else Časť> fi <príkazy><else Časť> fi <príkazy>	P
16	fi else p1 fi	<príkazy><else Časť> fi <príkazy><else Časť> fi <príkazy>	E2
17	fi else p1 fi	<else Časť> fi <príkazy><else Časť> fi <príkazy>	E7
18	fi else p1 fi	fi <príkazy><else Časť> fi <príkazy>	P
19	else p1 fi	<príkazy><else Časť> fi <príkazy>	E2
20	else p1 fi	<else Časť> fi <príkazy>	E6
21	else p1 fi	else <príkazy> fi <príkazy>	P
22	p1 fi	<príkazy> fi <príkazy>	E1
23	p1 fi	<príkaz><príkazy> fi <príkazy>	E4
24	p1 fi	p1 <príkazy> fi <príkazy>	P
25	fi	<príkazy> fi <príkazy>	E2
26	fi	fi <príkazy>	P
27	ε	<príkazy>	E2
28	ε	ε	A



- Všimnite si, že na predchádzajúcom slajde je príklad výpočtu, v ktorom na r. 27 je vstup celý prečítaný, avšak zásobník **nie je vyprázdnený**. To však **neznamená**, že slovo nemá deriváciu, keďže v gramatike **existuje pravidlo**, ktoré môžeme použiť v LL(1) analyzátore aj keď je vstup prázdny.
- Tým sa zbavíme neterminálu <príkazy> zo zásobníka, čím ho definitívne vyprázdňime a teda zostane prázdny vstup aj zásobník a celý vstup **akceptujeme**.
- Zároveň vidíme, že vstup má v gramatike ľavú deriváciu v poradí aplikácie pravidiel 1,3,8,1,3,8,1,4,2,7,2,6,1,4,2,2.





- Daná gramatika popisuje syntax typu podmieneného príkazu, v ktorom je kľúčové slovo **if** spárované so slovom **fi** na označenie konca podmieneného príkazu.
- To znamená, že ak by sa na slovo **fi** v kóde "zabudlo", musí dôjsť k syntaktickej chybe počas syntaktickej analýzy.
- Skúsme to na príklade: **if podmienka then p1 else p2**

r.	Zvyšok vstupu	Zásobník	Akcia
1	if podmienka then p1 else p2	<príkazy>	E1
2	if podmienka then p1 else p2	<príkaz><príkazy>	E3
3	if podmienka then p1 else p2	if <podmienka> then <príkazy><elseČasť> fi <príkazy>	P
4	podmienka then p1 else p2	<podmienka> then <príkazy><elseČasť> fi <príkazy>	E8
5	podmienka then p1 else p2	podmienka then <príkazy><elseČasť> fi <príkazy>	P
6	then p1 else p2	then <príkazy><elseČasť> fi <príkazy>	P
7	p1 else p2	<príkazy><elseČasť> fi <príkazy>	E1
8	p1 else p2	<príkaz><príkazy><elseČasť> fi <príkazy>	E4
9	p1 else p2	p1 <príkazy><elseČasť> fi <príkazy>	P
10	else p2	<príkazy><elseČasť> fi <príkazy>	E2
11	else p2	<elseČasť> fi <príkazy>	E6
12	else p2	else <príkazy> fi <príkazy>	P
13	p2	<príkazy> fi <príkazy>	E1
14	p2	<príkaz><príkazy> fi <príkazy>	E5
15	p2	p2 <príkazy> fi <príkazy>	P
16	ε	<príkazy> fi <príkazy>	E2
17	ε	fi <príkazy>	ERROR

- Podľa stavu syntaktického analyzátora vidíme, že sa pokúšal urobiť **porovnanie** symbolu **fi** na vrchu zásobníka so vstupom
- To signalizuje, že syntaktický analyzátor **očakával** na vstupe symbol **fi**
- Avšak, vstup ho v danom momente neobsahoval - došlo teda k syntaktickej chybe a vstupný reťazec **nemá v gramatike deriváciu**, teda nie je syntakticky korektný.

Príklad č. 4

Je daná bezkontextová gramatika $G = (N, T, P, S)$, S je počiatočný neterminál.

Pravidlá:

1. $S \rightarrow Ad$
2. $A \rightarrow BCD$
3. $B \rightarrow D$
4. $B \rightarrow bB$
5. $C \rightarrow cD$
6. $C \rightarrow D$
7. $D \rightarrow \varepsilon$

Terminály $T = \{b, c, d\}$, neterminály $N = \{S, A, B, C, D\}$.

Zostrojte rozkladovú tabuľku $LL(1)$ -analyzátoru a na jej základe určte, či sa jedná o $LL(1)$ -gramatiku.



Najprv vypočítame množiny *FIRST* a *FOLLOW* pre neterminály:

	<i>S</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>FIRST</i>	$\{b, c, d\}$	$\{b, c, \varepsilon\}$	$\{b, \varepsilon\}$	$\{\varepsilon, c\}$	$\{\varepsilon\}$
<i>FOLLOW</i>	$\{\varepsilon\}$	$\{d\}$	$\{c, d\}$	$\{d\}$	$\{c, d\}$

Následne určíme množiny *PREDICT* pre jednotlivé pravidlá:

1. $PREDICT(S \rightarrow Ad) = \{b, c, d\}$. Zdôvodnenie:
 - $FIRST(Ad) = \{b, c, d\}$. A keďže $\varepsilon \notin FIRST(Ad)$, tak $PREDICT(S \rightarrow Ad) = FIRST(Ad) = \{b, c, d\}$.
2. $PREDICT(A \rightarrow BCD) = \{b, c, d\}$. Zdôvodnenie:
 - $FIRST(BCD) = \{b, c, \varepsilon\}$. A keďže $\varepsilon \in FIRST(BCD)$, tak $PREDICT(A \rightarrow BCD) = (FIRST(BCD) \setminus \{\varepsilon\}) \cup FOLLOW(A) = \{b, c\} \cup \{d\} = \{b, c, d\}$.
3. $PREDICT(B \rightarrow D) = \{c, d\}$. Zdôvodnenie:
 - $FIRST(D) = \{\varepsilon\}$. A keďže $\varepsilon \in FIRST(D)$, tak $PREDICT(B \rightarrow D) = (FIRST(D) \setminus \{\varepsilon\}) \cup FOLLOW(B) = \emptyset \cup \{c, d\} = \{c, d\}$.
4. $PREDICT(B \rightarrow bB) = \{b\}$. Zdôvodnenie:
 - $FIRST(bB) = \{b\}$. A keďže $\varepsilon \notin FIRST(bB)$, tak $PREDICT(B \rightarrow bB) = FIRST(bB) = \{b\}$.



5. $PREDICT(C \rightarrow cD) = \{c\}$. Zdôvodnenie:

- $FIRST(cD) = \{c\}$. A keďže $\varepsilon \notin FIRST(cD)$, tak $PREDICT(C \rightarrow cD) = FIRST(cD) = \{c\}$.

6. $PREDICT(C \rightarrow D) = \{d\}$. Zdôvodnenie:

- $FIRST(D) = \{\varepsilon\}$. A keďže $\varepsilon \in FIRST(D)$, tak $PREDICT(C \rightarrow D) = (FIRST(\varepsilon) \setminus \{\varepsilon\}) \cup FOLLOW(C) = \emptyset \cup \{d\} = \{d\}$.

7. $PREDICT(D \rightarrow \varepsilon) = \{c, d\}$. Zdôvodnenie:

- $FIRST(\varepsilon) = \{\varepsilon\}$. A keďže $\varepsilon \in FIRST(\varepsilon)$, tak $PREDICT(D \rightarrow \varepsilon) = (FIRST(\varepsilon) \setminus \{\varepsilon\}) \cup FOLLOW(D) = \emptyset \cup \{c, d\} = \{c, d\}$.

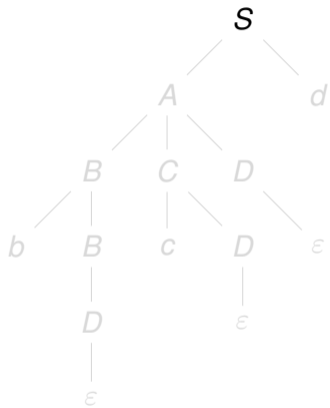
Na základe množín *PREDICT* jednotlivých pravidiel gramatiky vieme vyplniť rozkladovú tabuľku:

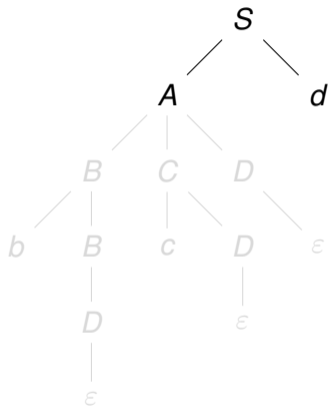
RT	<i>b</i>	<i>c</i>	<i>d</i>	ϵ
<i>S</i>	1	1	1	
<i>A</i>	2	2	2	
<i>B</i>	4	3	3	
<i>C</i>		5	6	
<i>D</i>		7	7	

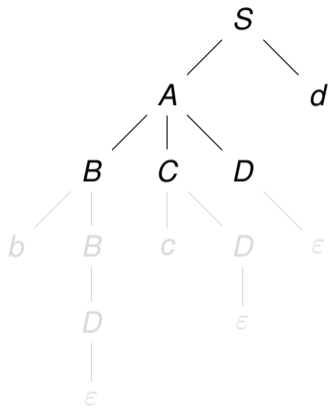
Keďže v rozkladovej tabuľke sa v každej bunke nachádza **maximálne jedno pravidlo**, gramatika **je** *LL(1)* gramatikou a *LL(1)* analyzátor **sa dá použiť**.

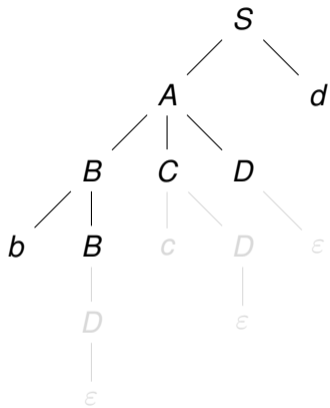
Zistite, či má v uvedenej gramatike deriváciu reťazec *bcd* pomocou LL(1) syntaktického analyzátora.

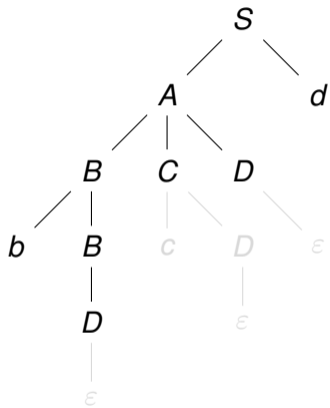
r.	Zvyšok vstupu	Zásobník □	Akcia
1	<i>bcd</i>	<i>S</i>	E1
2	<i>bcd</i>	<i>Ad</i>	E2
3	<i>bcd</i>	<i>BCDd</i>	E4
4	<i>bcd</i>	<i>bBCDd</i>	P
5	<i>cd</i>	<i>BCDd</i>	E3
6	<i>cd</i>	<i>DCDd</i>	E7
7	<i>cd</i>	<i>CDd</i>	E5
8	<i>cd</i>	<i>cDDd</i>	P
9	<i>d</i>	<i>DDd</i>	E7
10	<i>d</i>	<i>Dd</i>	E7
11	<i>d</i>	<i>d</i>	P
12	ϵ	ϵ	A

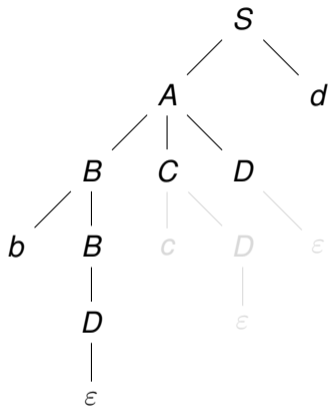


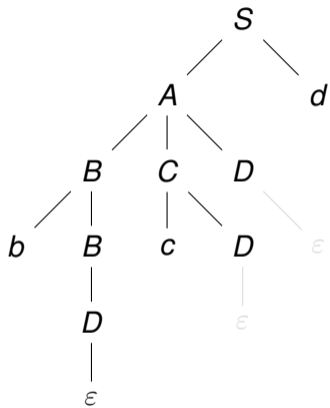


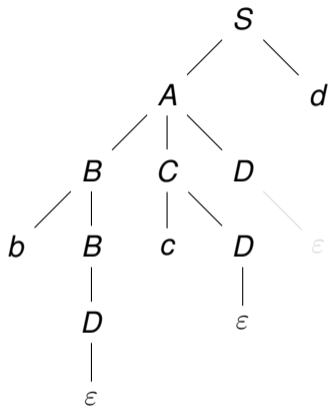


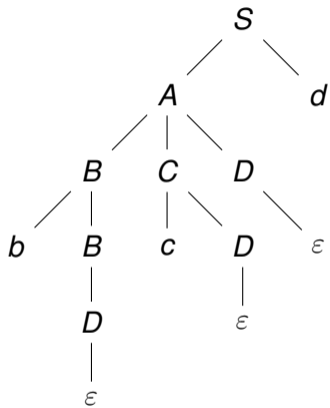












Príklad č. 5

Je daná bezkontextová gramatika $G = (N, T, P, S)$, S je počiatočný neterminál.

Pravidlá:

1. $S \rightarrow AB$
2. $S \rightarrow cSc$
3. $A \rightarrow aAb$
4. $A \rightarrow \varepsilon$
5. $B \rightarrow aBa$
6. $B \rightarrow bBb$
7. $B \rightarrow \varepsilon$

Terminály $T = \{a, b, c\}$, neterminály $N = \{S, A, B\}$.

Zostrojte rozkladovú tabuľku $LL(1)$ -analyzátoru a na jej základe určte, či sa jedná o $LL(1)$ -gramatiku.



Najprv vypočítame množiny *FIRST* a *FOLLOW* pre neterminály:

	<i>S</i>	<i>A</i>	<i>B</i>
<i>FIRST</i>	$\{a, b, c, \varepsilon\}$	$\{a, \varepsilon\}$	$\{a, b, \varepsilon\}$
<i>FOLLOW</i>	$\{c, \varepsilon\}$	$\{a, b, c, \varepsilon\}$	$\{a, b, c, \varepsilon\}$

Následne určíme množiny *PREDICT* pre jednotlivé pravidlá:

1. $PREDICT(S \rightarrow AB) = \{a, b, c, \varepsilon\}$. Zdôvodnenie:
 - $FIRST(AB) = \{a, b, \varepsilon\}$. A keďže $\varepsilon \in FIRST(AB)$, tak $PREDICT(S \rightarrow AB) = (FIRST(AB) \setminus \{\varepsilon\}) \cup FOLLOW(S) = \{a, b, c, \varepsilon\}$.
2. $PREDICT(S \rightarrow cSc) = \{c\}$. Zdôvodnenie:
 - $FIRST(cSc) = \{c\}$. A keďže $\varepsilon \notin FIRST(cSc)$, tak $PREDICT(S \rightarrow cSc) = FIRST(cSc) = \{c\}$.
3. $PREDICT(A \rightarrow aAb) = \{a\}$. Zdôvodnenie:
 - $FIRST(aAb) = \{a\}$. A keďže $\varepsilon \notin FIRST(aAb)$, tak $PREDICT(A \rightarrow aAb) = FIRST(aAb) = \{a\}$.
4. $PREDICT(A \rightarrow \varepsilon) = \{a, b, c, \varepsilon\}$. Zdôvodnenie:
 - $FIRST(\varepsilon) = \{\varepsilon\}$. A keďže $\varepsilon \in FIRST(\varepsilon)$, tak $PREDICT(A \rightarrow \varepsilon) = (FIRST(\varepsilon) \setminus \{\varepsilon\}) \cup FOLLOW(A) = \{a, b, c, \varepsilon\}$.



5. $PREDICT(B \rightarrow aBa) = \{a\}$. Zdôvodnenie:

- $FIRST(aBa) = \{a\}$. A keďže $\varepsilon \notin FIRST(aBa)$, tak $PREDICT(B \rightarrow aBa) = FIRST(aBa) = \{a\}$.

6. $PREDICT(B \rightarrow bBb) = \{b\}$. Zdôvodnenie:

- $FIRST(bBb) = \{b\}$. A keďže $\varepsilon \notin FIRST(bBb)$, tak $PREDICT(B \rightarrow bBb) = FIRST(bBb) = \{b\}$

7. $PREDICT(B \rightarrow \varepsilon) = \{a, b, c, \varepsilon\}$. Zdôvodnenie:

- $FIRST(\varepsilon) = \{\varepsilon\}$. A keďže $\varepsilon \in FIRST(\varepsilon)$, tak $PREDICT(B \rightarrow \varepsilon) = (FIRST(\varepsilon) \setminus \{\varepsilon\}) \cup FOLLOW(B) = \emptyset \cup \{a, b, c, \varepsilon\} = \{a, b, c, \varepsilon\}$.



Na základe množín *PREDICT* jednotlivých pravidiel gramatiky vieme vyplniť rozkladovú tabuľku:

RT	<i>a</i>	<i>b</i>	<i>c</i>	ϵ
<i>S</i>	1	1	1,2	1
<i>A</i>	3,4	4	4	4
<i>B</i>	5,7	6,7	7	7

Keďže v rozkladovej tabuľke sú bunky, kde sa nachádza **viac než jedno pravidlo**, gramatika **nie je** *LL(1)* gramatikou a *LL(1)* analyzátor **sa nedá použiť**.

Príklad č. 6

Je daná bezkontextová gramatika $G = (N, T, P, S)$, S je počiatočný neterminál.

Pravidlá:

1. $S \rightarrow dA$
2. $S \rightarrow aSb$
3. $A \rightarrow BC$
4. $B \rightarrow \varepsilon$
5. $C \rightarrow \varepsilon$
6. $C \rightarrow aC$

Terminály $T = \{a, b, d\}$, neterminály $N = \{S, A, B, C\}$.

Zostrojte rozkladovú tabuľku $LL(1)$ -analyzátoru a na jej základe určte, či sa jedná o $LL(1)$ -gramatiku.



Najprv vypočítame množiny *FIRST* a *FOLLOW* pre neterminály:

	<i>S</i>	<i>A</i>	<i>B</i>	<i>C</i>
<i>FIRST</i>	$\{a, d\}$	$\{a, \varepsilon\}$	$\{\varepsilon\}$	$\{a, \varepsilon\}$
<i>FOLLOW</i>	$\{\varepsilon, b\}$	$\{\varepsilon, b\}$	$\{a, b, \varepsilon\}$	$\{\varepsilon, b\}$



Následne určíme množiny *PREDICT* pre jednotlivé pravidlá:

1. $PREDICT(S \rightarrow dA) = \{d\}$. Zdôvodnenie:

- $FIRST(dA) = \{d\}$. A keďže $\varepsilon \notin FIRST(dA)$, tak $PREDICT(S \rightarrow dA) = FIRST(dA) = \{d\}$.

2. $PREDICT(S \rightarrow aSb) = \{a\}$. Zdôvodnenie:

- $FIRST(aSb) = \{a\}$. A keďže $\varepsilon \notin FIRST(aSb)$, tak $PREDICT(S \rightarrow aSb) = FIRST(aSb) = \{a\}$.

3. $PREDICT(A \rightarrow BC) = \{a, b, \varepsilon\}$. Zdôvodnenie:

- $FIRST(BC) = \{a, \varepsilon\}$. A keďže $\varepsilon \in FIRST(BC)$, tak $PREDICT(A \rightarrow BC) = (FIRST(BC) \setminus \{\varepsilon\}) \cup FOLLOW(A) = \{a\} \cup \{\varepsilon, b\} = \{a, b, \varepsilon\}$.

4. $PREDICT(B \rightarrow \varepsilon) = \{a, b, \varepsilon\}$. Zdôvodnenie:

- $FIRST(\varepsilon) = \{\varepsilon\}$. A keďže $\varepsilon \in FIRST(\varepsilon)$, tak $PREDICT(B \rightarrow \varepsilon) = (FIRST(\varepsilon) \setminus \{\varepsilon\}) \cup FOLLOW(B) = \emptyset \cup \{a, b, \varepsilon\} = \{a, b, \varepsilon\}$.



5. $PREDICT(C \rightarrow \varepsilon) = \{b, \varepsilon\}$. Zdôvodnenie:

- $FIRST(\varepsilon) = \{\varepsilon\}$. A keďže $\varepsilon \in FIRST(\varepsilon)$, tak
 $PREDICT(C \rightarrow \varepsilon) = (FIRST(\varepsilon) \setminus \{\varepsilon\}) \cup FOLLOW(C) = \emptyset \cup \{b, \varepsilon\} = \{b, \varepsilon\}$.

6. $PREDICT(C \rightarrow aC) = \{a\}$. Zdôvodnenie:

- $FIRST(aC) = \{a\}$. A keďže $\varepsilon \notin FIRST(aC)$, tak
 $PREDICT(C \rightarrow aC) = FIRST(aC) = \{a\}$.



Na základe množín *PREDICT* jednotlivých pravidiel gramatiky vieme vyplniť rozkladovú tabuľku:

RT	<i>a</i>	<i>b</i>	<i>d</i>	ϵ
<i>S</i>	2.		1.	
<i>A</i>	3.	3.		3.
<i>B</i>	4.	4.		4.
<i>C</i>	6.	5.		5.

Keďže v rozkladovej tabuľke **nie sú bunky**, kde by sa nachádzalo **viac než jedno pravidlo**, gramatika **je** *LL(1)* gramatikou a *LL(1)* analyzátor **sa dá použiť**.

Zistite, či má v uvedenej gramatike deriváciu reťazec d pomocou LL(1) syntaktického analyzátora.

r.	Zvyšok vstupu	Zásobník \square	Akcia
1	d	S	E1
2	d	dA	P
3	ε	A	E3
4	ε	BC	E4
5	ε	C	E5
6	ε	ε	A

Všimnite si, že výpočet sa v treťom kroku nezasekol - hoci vstup bol celý prečítaný a zásobník nebol vyprázdnený! Vďaka stĺpcu rozkladovej tabuľky so záhlavím ε , ktorý sa používa v situácii, že bol vstup celý prečítaný, bolo možné ďalej expandovať neterminály a dospieť do situácie, že sa podarilo zásobník vyprázdniť!

