

Zásobníkový automat a gramatika

Analýza odvození slova

Analýza podle gramatiky

- Ověření, zda lze slovo odvodit = analýza
- Dva typy podle startu a cíle:
 - Zdola: od slova k počátečnímu symbolu
 - Shora: od počátečního symbolu ke slovu
- Zásobník:
 - Záznam odvození – zde se přepisují symboly dle typu analýzy
- Rozkladová tabulka:
 - Pravidla gramatiky přepsaná do řeči zásobníkového automatu
 - Různá pro analýzu shora a zdola

Analýza shora

- **Start: Počáteční symbol, Cíl: Slovo**
- **LL analýza: přes levé derivace** hledáme levou derivaci, vstupní slovo analyzujeme zleva
 - Přesně určuje volbu pravidel při analýze a umožňuje jednoznačný postup při odvození
 - Gramatika, která je jednoznačná a lze ji takto analyzovat: LL gramatika
- **LL(k)** označení gramatiky pro LL analýzu, číslo k určuje, kolik následujících symbolů je nutné znát pro analýzu slova
- **LL(1)**: nejpoužívanější gramatika, stačí znát jeden následující symbol
- **LL(0)**: umožňuje jen jazyky s konečným počtem slov
- **LL** gramatiky s $k > 1$ lze převést na LL gramatiky s $k = 1$
 - Existují přesné popisy, jak jednotlivá pravidla nahrazovat (přidávají se neterminály a pravidla se upravují, aby při analýze stačilo znát jeden další symbol)

Zásobníkový automat pro analýzu shora

- $Q = \{q_0\}$: jen jeden vnitřní stav
- Abeceda gramatiky a automatu je shodná
- Abeceda zásobníku jsou terminály i neterminály z gramatiky
- Rozkladová tabulka: zobrazení přiřazující změnu situace; konstruuje se z pravidel gramatiky
- Rozpoznávání do prázdného zásobníku

Rozkladová tabulka při rozpoznávání shora

- Rozkladová tabulka je zobrazení:
 $(\Pi \cup \Sigma \cup \#) \times (\Sigma \cup \epsilon \cup \$) \rightarrow \{\text{expand } i, \text{pop}, \text{accept}, \text{error}\}$
- Význam funkčních hodnot:
 - **Expand i:** Na vrcholu zásobníku je netereminál A, který je přepsán na pravou stranu pravidla, jež má A na levé straně
 - **Pop:** Na vrcholu zásobníku a na vstupu je stejný terminál, ten je z vrcholu zásobníku odstraněn a čte se další znak.
 - **Accept:** Konec rozpoznávání s přijetím; prázdný zásobník.
 - **Error:** Chyba při rozpoznávání, vstupní řetězec do jazyka nepatří.

Gramatika pro přednášku

- Algebraické výrazy:

$G = (\Sigma, \Pi, P, S)$ kde:

- $\Sigma = \{a, +, *, (,)\}$
- $\Pi = \{S, T, F\}$
- S je počáteční symbol.
- Produkční pravidla:

1) $S \rightarrow T + S$

2) $S \rightarrow T$

3) $T \rightarrow F * T$

4) $T \rightarrow F$

5) $F \rightarrow (S)$

6) $F \rightarrow a$

Levá strana	Pravé strany	Číslo pravidel
S	$T + S \mid T$	(1) (2)
T	$F * T \mid F$	(3) (4)
F	$(S) \mid a$	(5) (6)

První znaky a následníci

- Příklad: $(a + a) * a$.
- Při hledání derivačního stromu kresleného shora:
 - Dosazujeme zleva (intuitivně předpokládáme gramatiku LL).
 - Chceme vědět, zda řetězec vůbec může začínat symbolem, jímž začíná, hledáme totiž neterminál, jenž mohl takový začátek generovat.
 - Chceme vědět, zda za již analyzovanou částí řetězce smí následovat zbytek - pro něj také hledáme neterminály.

First(N): Množina možných začátků

N je libovolná větná forma generovaná gramatikou G.
First(N) je množina všech terminálních symbolů, jimiž začínají řetězce odvozené z N.

Pokud v dané gramatice existuje odvození
 $N \rightarrow^* \varepsilon$, potom i ε patří do First (N).

*Množinu First hledáme pro pravé strany pravidel -
zajímá nás, čím to vlastně může začínat.*

Follow(N): Množina možných pokračování neterminálu N

Do množiny Follow(N) patří ty terminály, které se při libovolném odvození mohou ocitnout bezprostředně vpravo za neterminálem N:

Pokud $S \rightarrow^* XNaY$, potom $a \in \text{Follow}(N)$.

Poznámka: Jestliže za symbolem N v nějakém odvození nic nenásleduje, patří do Follow(N) také symbol pro konec řetězce (nepatří mezi terminály, přidáváme navíc).

Konstrukce množin $\text{First}(N)$

Algoritmus pro řešení konkrétní pravé strany jednoho pravidla:

1. **Je-li $N = \epsilon$:** $\text{First}(\epsilon) = \{\epsilon\}$. Konec.
2. **Je-li N ve tvaru aX , potom:** $\text{First}(N) = \{a\}$. Konec, jiný začátek není možný.
3. **N začíná neterminálem ($N=AX$):**
 - Nalezneme všechna A_i , jež lze odvodit z A .
 - Zjistíme, čím mohou začínat (nalezneme $\text{First}(A_i)$).
 - Množina $\text{First}(N)$ je sjednocením všech dílčích množin $\text{First}(A_i)$.
4. **N začíná neterminálem ($N=AX$) a současně lze A přepsat na prázdný symbol**
 - Jako další symbol zvolíme to, co za A následuje - a hledáme množinu $\text{First}(X)$ - tím může řetězec začínat.
 - Nalezneme také množiny $\text{First}(A_i)$, pokud lze A přepsat i na jiné neterminály.
 - Hledaná množina je sjednocení $\text{First}(X)$ a množin $\text{First}(A_i)$

Příklad pro G

Levá strana	Pravé strany	Číslo pravidel
S	$T + S \mid T$	(1) (2)
T	$F * T \mid F$	(3) (4)
F	$(S) \mid a$	(5) (6)

„N“	First(N)
T+S	{(, a}
T	{(, a}
F*T	{(, a}
F	{(,a}
(S)	{(}
a	{a}

Konstrukce množin Follow(N)

1. $\$ \in \text{Follow}(S)$
 - $\$$ - konec řetězce, S poč. symbol
2. $\text{First}(X) \subseteq \text{Follow}(N)$, pokud $Y \rightarrow ZNX$
 - Procházíme všechny pravé strany G a všude hledáme výskyt N .
 - U jednotlivých výskytů zjistíme, co následuje za N , pro tuto část pravidla hledáme $\text{First}(X)$ - to jsou možní bezprostřední následníci N .
3. $\text{Follow}(Y) \subseteq \text{Follow}(N)$, pokud $Y \rightarrow ZNX$ a současně $X \rightarrow^* \varepsilon$
 - Lze-li X přepsat do prázdného slova, musíme ještě zjistit $\text{Follow}(Y)$ - levé strany pravidla.

Příklad pro G

Levá strana	Pravé strany	Číslo pravidel
S	$T + S \mid T$	(1) (2)
T	$F * T \mid F$	(3) (4)
F	$(S) \mid a$	(5) (6)

N	Follow(N)	Odkud?
S	\$,)	Z definice. Pravidlo (5), X je zde řetězec „)“.
T	+,)	Pravidlo (1), X je zde řetězec „+ S“, $\text{First}(+S) = +$; Pravidlo(1), X je nyní prázdný řetězec, bereme tedy celé S z levé strany.
F	*,)	Pravidlo (3): X je *T, Pravidlo (5): X je)

Jak poznáme gramatiku LL(1)?

- Pro její pravidla platí:
 - **Vlastnost First-First:**
Pokud $Y \rightarrow X1 \mid X2 \dots \mid Xn$, potom musí platit:
 $\text{First}(X_i) \cap \text{First}(X_j) = \{\}$ pro všechna $i \neq j$
 - **Vlastnost First-Follow (někdy též FFL):**
Lze-li z X_i odvodit prázdný symbol, potom:
 $\text{First}(X_i) \cap \text{Follow}(Y) = \{\}$
- Poznámka: Naše gramatika G není typu LL(1), protože tato kritéria nesplňuje (příklad gramatiky generující stejný jazyk bude dále).

Konstrukce rozkladové tabulky

- Rozkladová tabulka je zobrazení:
 $(\Pi \cup \#) \times (\Sigma \cup \varepsilon) \rightarrow \{\text{expand } i, \text{pop}, \text{accept}, \text{error}\}$
- Význam funkčních hodnot:
 - **Expand i:** Na vrcholu zásobníku je netereminál A, který je přepsán na pravou stranu pravidla, jež má A na levé straně
 - **Pop:** Na vrcholu zásobníku a na vstupu je stejný terminál, ten je z vrcholu zásobníku odstraněn a čte se další znak.
 - **Accept:** Konec rozpoznávání s přijetím; prázdný zásobník.
 - **Error:** Chyba při rozpoznávání, vstupní řetězec do jazyka nepatří.

Rozkladová tabulka prakticky I.

- Tabulka určuje, podle kterého pravidla se provádí expanze jednotlivých neterminálů, když chceme získat konkrétní symboly.
- Řádky tabulky tvoří vše, co může obsahovat zásobník: neterminály, terminály, symbol konce zásobníku (#).
- Sloupce tvoří to, co může být vstup: terminály a symbol konce řetězce (\$).

Rozkladová tabulka prakticky II.

- Pro každé pravidlo s číslem (i) $A \rightarrow X$:
 - Nalezneme množinu $U = \text{First}(X.\text{Follow}(A))$
(Zjistíme, čím může začínat rozklad a jak může pokračovat).
 - V řádku označeném A ve sloupcích, v nichž jsou prvky množiny U uvedeme hodnotu *expand(i)*.
- V řádcích označených terminály uvádíme *pop* tam, kde je sloupec nadepsán stejným terminálem.
- Buňka tabulky v řádku (#,\$) má hodnotu *accept*.
- Buňky, jež jsou dosud prázdné, mají hodnotu *error*.

Struktura obecné rozkladové tabulky

Zásobník



	T (sloupce terminálů)	\$
N (řádky neterminálů)	Dle pravidel a algoritmu hodnoty expand(i) nebo error.	
T (řádky terminálů)	pop . . pop	
# (řádek konce zásobníku)		accept

Vstup



Příklad – rozkladová tabulka pro LL(1) gramatiku

- Algebraické výrazy, $\Pi = \{S, A, B, C, D\}$, $\Sigma = \{a, *, +, \varepsilon\}$, S
 1. $S \rightarrow AB$
 2. $B \rightarrow + AB$
 3. $B \rightarrow \varepsilon$
 4. $A \rightarrow CD$
 5. $D \rightarrow *CD$
 6. $D \rightarrow \varepsilon$
 7. $C \rightarrow (S)$
 8. $C \rightarrow a$

Levá strana	Pravé strany	Číslo pravidel	Follow(L)	First(P.Foll(L))
S	AB	(1)	\$,)	First(AB .Foll(S))={a}
B	+AB ϵ	(2) (3)	\$,)	First(+AB .Foll(B))={+} First(ϵ .Follow(B))={\$,)}
A	CD	(4)	+	First(CD .Follow(A))={a, (}
D	*CD ϵ	(5) (6)	+, \$	First(*CD .Follow(D))={*} First(ϵ .Follow(D))={+, \$}
C	(S) a	(7) (8)	+, *, \$	First((S) .Follow(C))={({} First(a .Follow(C))={a}

	a	+	*	()	\$
S	e1					
A	e4			e4		
B		e2			e3	e3
C	e8			e7		
D		e6	e5			e6

Celá rozkladová tabulka

	a	+	*	()	\$
S	e1					
A	e4			e4		
B		e2			e3	e3
C	e8			e7		
D		e6	e5			e6
a	pop					
+		pop				
*			pop			
(pop		
)					pop	
#						acc

Analýza slova $a+a*a$

S	AB	(1)
B	+AB ϵ	(2) (3)
A	CD	(4)
D	*CD ϵ	(5) (6)
C	(S) a	(7) (8)

	a	+	*	()	\$
S	e1					
A	e4			e4		
B		e2			e3	e3
C	e8			e7		
D		e6	e5			e6

Vstup	Zásobník	Akce
$a+a*a\$$	S#	e1
$a+a*a\$$	AB#	e4
$a+a*a\$$	CDB#	e8
$a+a*a\$$	aDB#	pop
$+a*a\$$	DB#	e6
$+a*a\$$	B#	e2
$+a*a\$$	+AB#	pop
$a*a\$$	AB#	e4
$a*a\$$	CDB#	e8
$a*a\$$	aDB#	pop
$*a\$$	DB#	e5
$*a\$$	*CDB#	pop
$a\$$	CDB#	e8
$a\$$	aDB#	pop
$\$$	DB#	e6
$\$$	B#	e3
$\$$	#	acc

Analýza slova a^+a

S	AB	(1)
B	$+AB \mid \varepsilon$	(2) (3)
A	CD	(4)
D	$*CD \mid \varepsilon$	(5) (6)
C	(S) a	(7) (8)

	a	+	*	()	\$
S	e1					
A	e4			e4		
B		e2			e3	e3
C	e8			e7		
D		e6	e5			e6

Vstup	Zásobník	Akce
$a^+a\$$	S#	e1
$a^+a\$$	AB#	e4
$a^+a\$$	CDB#	e8
$a^+a\$$	aDB#	pop
$+^+a\$$	DB#	e6
$+^+a\$$	B#	e2
$+^+a\$$	+AB#	pop
$*a\$$	AB#	error

Buňka rozkladové tabulky A,* je prázdná. Nelze tedy použít žádné pravidlo a dále pracovat se zásobníkem. Proto zásobník nebude prázdný.

Analýza zdola - LR(k)

- **Left-to-right:** vstup čteme zleva
- **Right** od „pravý rozklad“ - odpovídá analýze zdola
- Při rozhodování mezi pravidly potřebujeme znát nejvýše k symbolů

Jak poznáme LR(k) gramatiku?

- Pro pravidla platí:

V případě existence dvou pravých derivací:

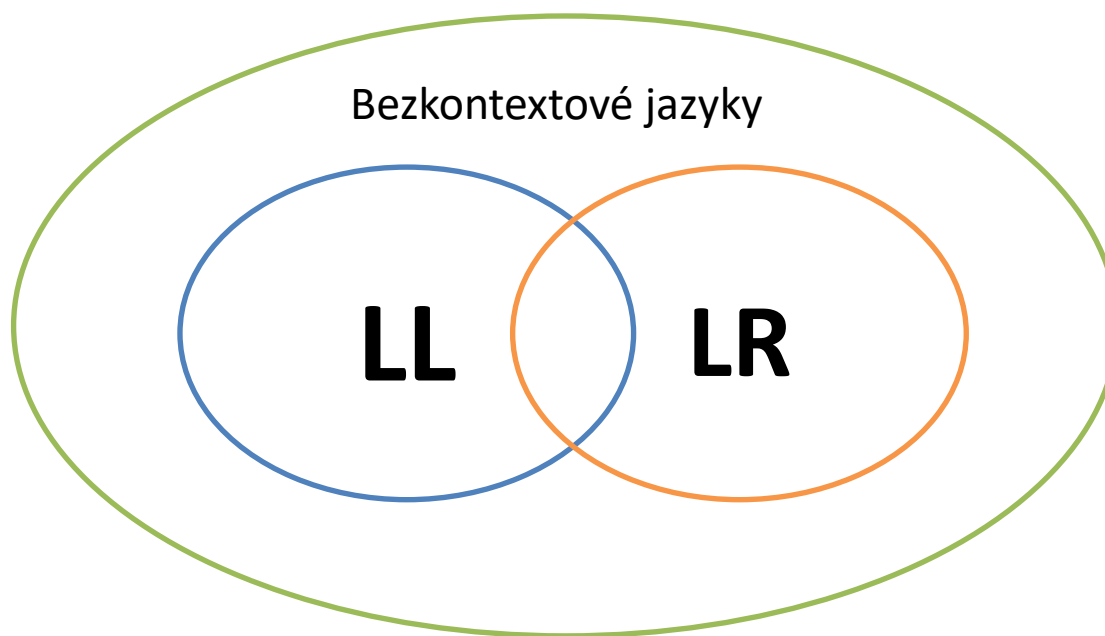
$$S \rightarrow^* \alpha Ax \rightarrow \alpha \gamma x$$

$$S \rightarrow^* \beta By \rightarrow \beta \gamma y$$

takových, že $\text{First}_k(x) = \text{First}_k(y)$,

vždy platí: $\alpha A = \beta B$

Souvislost mezi LL a LR



Čím se liší analýza zdola?

- Postup od řetězce k počátečnímu symbolu
- Jdeme „proti“ šipkám v derivačním stromu
- Rozkladová tabulka má akci *redukce* pro přepis části řetězce na nějaký terminál. (Analýza shora má akci *expanze*.)
- V řetězci na zásobníku hledáme strukturu, která půjde redukovat. (Při analýze shora hledáme pravidlo, které použijeme pro expanzi.)

Pomocné množiny

- Before(N):
 - N je neterminál.
 - Analogie follow, nyní ale hledáme vše, co neterminálu v pravidlech předchází.
- EFF(X):
 - Vycházíme z množin First(X), u každého prvku zkoumáme derivaci, díky níž je prvek do množiny First zařazen.
 - Vyřadíme ty symboly, které se do First dostaly s použitím ϵ pravidla pro nejlevější symbol.

Rozkladová tabulka při rozpoznávání zdola

- Rozkladová tabulka je zobrazení:
 $(\Pi \cup \Sigma \cup \#) \times (\Sigma \cup \varepsilon \cup \$) \rightarrow \{\text{push, reduce, accept, error}\}$
- Význam funkčních hodnot:
 - **Push:** Čtený symbol se přesouvá do zásobníku
 - **Reduce:** Provádí se redukce podle nějakého z pravidel gramatiky, nahrazujeme řetězec neterminály z levé strany
 - **Accept:** Konec rozpoznávání s přijetím; prázdný zásobník.
 - **Error:** Chyba při rozpoznávání, vstupní řetězec do jazyka nepatří.

Stručně ke konstrukci rozkladové tabulky

- Analogické s analýzou LL
- Pro každé pravidlo $(i) A \rightarrow \alpha X$ a pro všechny řetězce $u \in \text{Follow}(A)$ je buňka $(X,u) = \text{reduce}(i)$
- Pro každé pravidlo $(i) A \rightarrow \varepsilon$, pro všechny symboly $X \in \text{Before}(A)$ a pro všechny řetězce $u \in \text{Follow}(A)$ je buňka $(X,u) = \text{reduce}(i)$
- Pro každé $B \rightarrow \beta X \gamma$, $u \in \text{EFF}(\gamma.\text{Follow}(B))$ je buňka $(X,u) = \text{push}$
- Buňka $(S,\$) = \text{accept}$
- Ostatní buňky generují chybu.