Algoritmy syntaktické analýzy (pomocí CFG)

Vladimír Kadlec, Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

Základní postupy pro syntaktickou analýzu obecných bezkontextových gramatik

- všechny uvedené algoritmy pracují s polynomiální časovou a prostorovou složitostí
- algoritmus CKY – *Cocke*, *Kasami*, *Younger*;
- tabulková (chart) analýza (neplést s LR tabulkou):
  - shora dolů (*top-down*);
  - zdola nahoru (*bottom-up*);
  - analýza řízená hlavou pravidla (*head-driven*);
- Tomitův zobecněný algoritmus LR.

Syntaktická analýza

- Vstupy:
  - řetězec lexikálních kategorií (preterminálních symbolů) $a_1\ a_2\ \dots\ a_n$;
  - bezkontextová gramatika $G = \langle N, \Sigma, P, S \rangle$.
- Výstup:
  - efektivní reprezentace derivačních stromů.

Algoritmus CKY

- Gramatika musí být v Chomského normální formě.

  CNF (každá CFG jde do ní převést): $A \rightarrow BC$
  $D \rightarrow$ 'd'
- Derivujeme řetězce symbolů délky q na p pozici, značíme $w_{p,q}$.
- Derivace řetězců délky 1, $A \Rightarrow w_{p,1}$, je prováděno prohledáváním terminálních pravidel.
- Derivace delších řetězců $A \Rightarrow^* w_{p,q}$, $q \geq 2$ vyžaduje aby platilo $A \Rightarrow BC \Rightarrow^* w_{p,q}$. Tedy z B derivujeme řetězec délky k, $1 \leq k \leq q$, a z C derivujeme zbytek, řetězec délky $q - k$. Tzn. $B \Rightarrow^* w_{p,k}$ a $C \Rightarrow^* w_{p+k,q-k}$. Kratší řetězce máme tedy vždy "předočítané."

Algoritmus CKY pokrač.

program CKY Parser;
begin
  for $p := 1$ **to** n **do** $V[p, 1] := \{A|A \rightarrow a_p \in P\}$;
  for $q := 2$ **to** n **do**
    for $p := 1$ **to** $n - q + 1$ **do**
      $V[p, q] = \emptyset$;
      for k :=1 **to** $q - 1$ **do**
        $V[p, q] =$
          $V[p, q] \cup$
          $\cup \{A|A \rightarrow BC \in P, B \in V[p, k], C \in V[p + k, q - k]\}$
      od
    od
  od
end

Algoritmus CKY, příklad – zadání

- vstupní gramatika je:

  $S \rightarrow AA|BB|AX|BY|a|b$
  $X \rightarrow SA$
  $Y \rightarrow SB$
  $A \rightarrow a$
  $B \rightarrow b$

- vstupní řetězec je $w = abaaba$.

Algoritmus CKY, příklad – řešení (matice V)

$S \rightarrow AA|BB|AX|BY|a|b$
$X \rightarrow SA$
$Y \rightarrow SB$
$A \rightarrow a$
$B \rightarrow b$

`a b a a b a`

p – pozice, q – délka

q \ p	1	2	3	4	5	6
1	S, A	S, B	S, A	S, A	S, B	S, A
2	Y	X	S, X	Y	X	
3	S	$\emptyset$	Y	S		
4	X	S	$\emptyset$			
5	$\emptyset$	X				
6	S					

Tabulkové (chart) analyzátory

- Rozlišujeme tři základní typy **tabulkových analyzátorů**:
  - shora dolů;
  - zdola nahoru;
  - analýza řízená hlavou pravidla.
- Mnoho dalších variant je popsáno v:
  Sikkel Klaas: Parsing Schemata: A Framework for Specification and Analysis of Parsing Algorithm, 1997.
- Neklade se žádné omezení na gramatiku.
- Analyzátory typu "chart" v sobě většinou obsahují dvě datové struktury **chart** a **agendu**. Chart a agenda obsahují tzv. *hrany*.
- **Hrana** je trojice $[A \rightarrow \alpha_\bullet\beta,\ i,\ j]$, kde:
  - i, j jsou celá čísla, $0 \leq i \leq j \leq n$ pro n slov ve vstupní větě
  - a $A \rightarrow \alpha\beta$ je pravidlem vstupní gramatiky.

$[A \rightarrow BC \bullet DE, 0, 3]$

0 a 1 b 2 a 3 a 4 b 5 a 6

Obecný analyzátor typu "chart"

program Chart Parser;
begin
    inicializuj (*CHART*);
    inicializuj (*AGENDA*);
    while (*AGENDA* není prázdná) **do**
        E := vezmi hranu z *AGENDA*;
        for each (hrana F, která může být vytvořena pomocí
          hrany E a nějaké jiné hrany z *CHART*) **do**
          if ((F není v *AGENDA*) **and** (F není v *CHART*) **and**
            (F je různá od E)
            then přidej F do *AGENDA*;
          fi;
        od;
        přidej E do *CHART*;
    od;
end;

Varianta shora dolů

Inicializace:

- $\forall\ p \in P \mid p = S \rightarrow \alpha$ přidej hranu $[S\rightarrow {\bullet}\alpha,\ 0,\ 0]$ do agendy.
- počáteční chart je prázdný.

Iterace – vezmi hranu E z agendy a pak:

a) pokud je E ve tvaru $[A\rightarrow \alpha_\bullet,\ j,\ k]$, potom pro každou hranu $[B\rightarrow \gamma_\bullet\ A\ \beta,\ i,\ j]$ v chartu vytvoř hranu $[B\rightarrow \gamma\ A\ {\bullet}\beta,\ i,\ k]$.

b) pokud je E ve tvaru $[B\rightarrow \gamma_\bullet\ A\ \beta,\ i,\ j]$, potom pro každou hranu $[A\rightarrow \alpha_\bullet,\ j,\ k]$ v chartu vytvoř hranu $[B \rightarrow\ \gamma\ A\ {\bullet}\beta,\ i,\ k]$.

c) pokud je E ve tvaru $[A\rightarrow \alpha_\bullet a_{j+1}\beta,\ i,\ j]$, vytvoř hranu $[A \rightarrow\ \alpha\ a_{j+1\bullet}\beta,\ i,\ j{+}1]$.

d) (*predikční pravidlo*) pokud je E ve tvaru $[A\rightarrow \alpha_\bullet\ B\ \beta,\ i,\ j]$ potom pro každé pravidlo $B\rightarrow \gamma \in P$, vytvoř hranu $[B\rightarrow {\bullet}\ \gamma,\ i,\ i]$.

Příklad – tabulkové analýzy (typu chart)

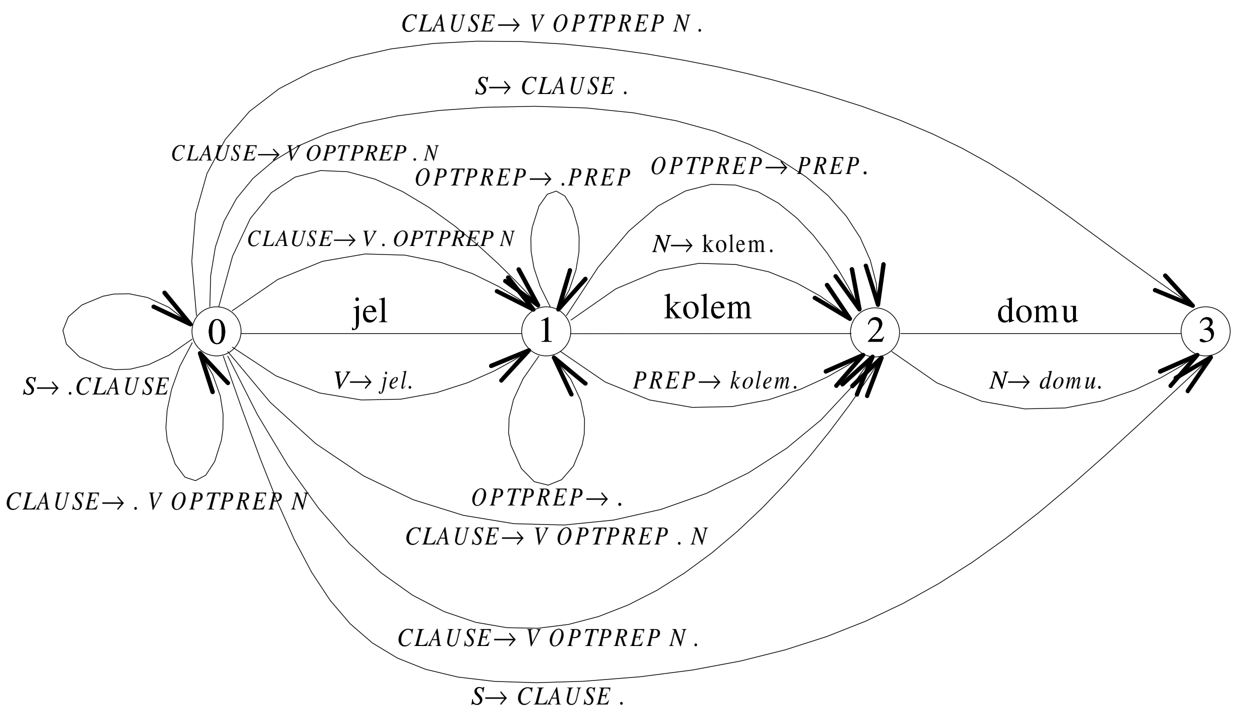
Gramatika:

$$
\begin{array}{rcl}
S & \longrightarrow & CLAUSE \\
CLAUSE & \longrightarrow & V\ OPTPREP\ N \\
OPTPREP & \longrightarrow & \epsilon \\
OPTPREP & \longrightarrow & PREP \\
V & \longrightarrow & jel \\
PREP & \longrightarrow & kolem \\
N & \longrightarrow & domu \\
N & \longrightarrow & kolem
\end{array}
$$

Věta:

"jel kolem domu" (a_1=*jel*, a_2=*kolem*, a_3=*domu*).

Příklad – chart po analýze shora dolů



Varianta zdola nahoru

Inicializace:

- $\forall\ p \in P \mid p = A \rightarrow \epsilon$ přidej hrany $[A \rightarrow \bullet, 0, 0]$, $[A \rightarrow \bullet, 1, 1]$, ..., $[A \rightarrow \bullet, n, n]$ do agendy.
- $\forall\ p \in P \mid p = A \rightarrow a_i\alpha$ přidej hranu $[A \rightarrow \bullet a_i\alpha$, i-1, i-1$]$ do agendy.
- počáteční chart je prázdný.

Iterace – vezmi hranu E z agendy a pak:

a) pokud je E ve tvaru $[A \rightarrow \alpha_\bullet, j, k]$, potom pro každou hranu $[B \rightarrow \gamma_\bullet\ A\ \beta, i, j]$ v chartu vytvoř hranu $[B \rightarrow \gamma\ A\ \bullet\beta, i, k]$.

b) pokud je E ve tvaru $[B \rightarrow \gamma_\bullet\ A\ \beta, i, j]$, potom pro každou hranu $[A \rightarrow \alpha_\bullet, j, k]$ v chartu vytvoř hranu $[B \rightarrow\ \gamma\ A\ \bullet\beta, i, k]$.

c) pokud je E ve tvaru $[A \rightarrow \alpha_\bullet a_{j+1}\beta, i, j]$, potom vytvoř hranu $[A \rightarrow\ \alpha\ a_{j+1\bullet}\beta, i, j{+}1]$.

d) pokud je E ve tvaru $[A \rightarrow \alpha_\bullet, i, j]$, potom pro každé pravidlo $B \rightarrow A\gamma$ vstupní gramatiky vytvoř hranu $[B \rightarrow \bullet A\gamma, i, i]$.

Analýza řízená hlavou pravidla

- *head-driven chart parsing*
- **Hlava pravidla** – libovolný (určený) symbol z pravé strany pravidla.

  Například pravidlo $CLAUSE \rightarrow V\ \underline{OPTPREP}\ N$ může mít hlavy V, $OPTPREP$, N.
- Epsilon pravidlo má hlavu ϵ.
- Hrana v analyzátoru řízeném hlavou pravidla – trojice $[A \rightarrow \alpha_\bullet\beta_\bullet\gamma, i, j]$, kde i, j jsou celá čísla, $0 \leq i \leq j \leq n$ pro n slov ve vstupní větě a $A \rightarrow \alpha\beta\gamma$ je pravidlo vstupní gramatiky a hlava je v β.
- Algoritmus vlastní analýzy (varianta zdola nahoru) je podobný jednoduchému přístupu. Analýza neprobíhá zleva doprava, ale začíná na hlavě daného pravidla.

Analyzátor řízený hlavou pravidla

Inicializace:

- $\forall\ p \in P \mid p = A\rightarrow \epsilon$ přidej hrany $[A\rightarrow {\bullet}{\bullet}$, 0, 0$]$, $[A\rightarrow {\bullet}{\bullet}$, 1, 1$]$, ..., $[A\rightarrow {\bullet}{\bullet}$, n, n$]$ do agendy.
- $\forall\ p \in P \mid p = A\rightarrow \alpha\underline{a_i}\beta$ (a_i je hlavou pravidla) přidej hranu $[A\rightarrow \alpha_\bullet a_{i\bullet}\beta$, i-1, i$]$ do agendy.
- počáteční chart je prázdný.

Je tato inicializace v pořádku?

Analyzátor řízený hlavou pravidla pokrač.

Iterace – vezmi hranu E z agendy a pak:

a_1) pokud je E ve tvaru $[A\rightarrow {}_\bullet\alpha_\bullet,\ j,\ k]$, potom pro každou hranu: $[B\rightarrow \beta_\bullet\gamma_\bullet A\delta,\ i,\ j]$ v chartu vytvoř hranu $[B\rightarrow \beta_\bullet\gamma A_\bullet\delta,\ i,\ k]$.

a_2) $[B\rightarrow \beta A_\bullet\gamma_\bullet\delta,\ k,\ l]$ v chartu vytvoř hranu $[B\rightarrow \beta_\bullet A\gamma_\bullet\delta,\ j,\ l]$.

b_1) pokud je E ve tvaru $[B\rightarrow \beta_\bullet\gamma_\bullet A\delta,\ i,\ j]$, potom pro každou hranu $[A\rightarrow {}_\bullet\alpha_\bullet,\ j,\ k]$ v chartu vytvoř hranu $[B\rightarrow \beta_\bullet\gamma A_\bullet\delta,\ i,\ k]$.

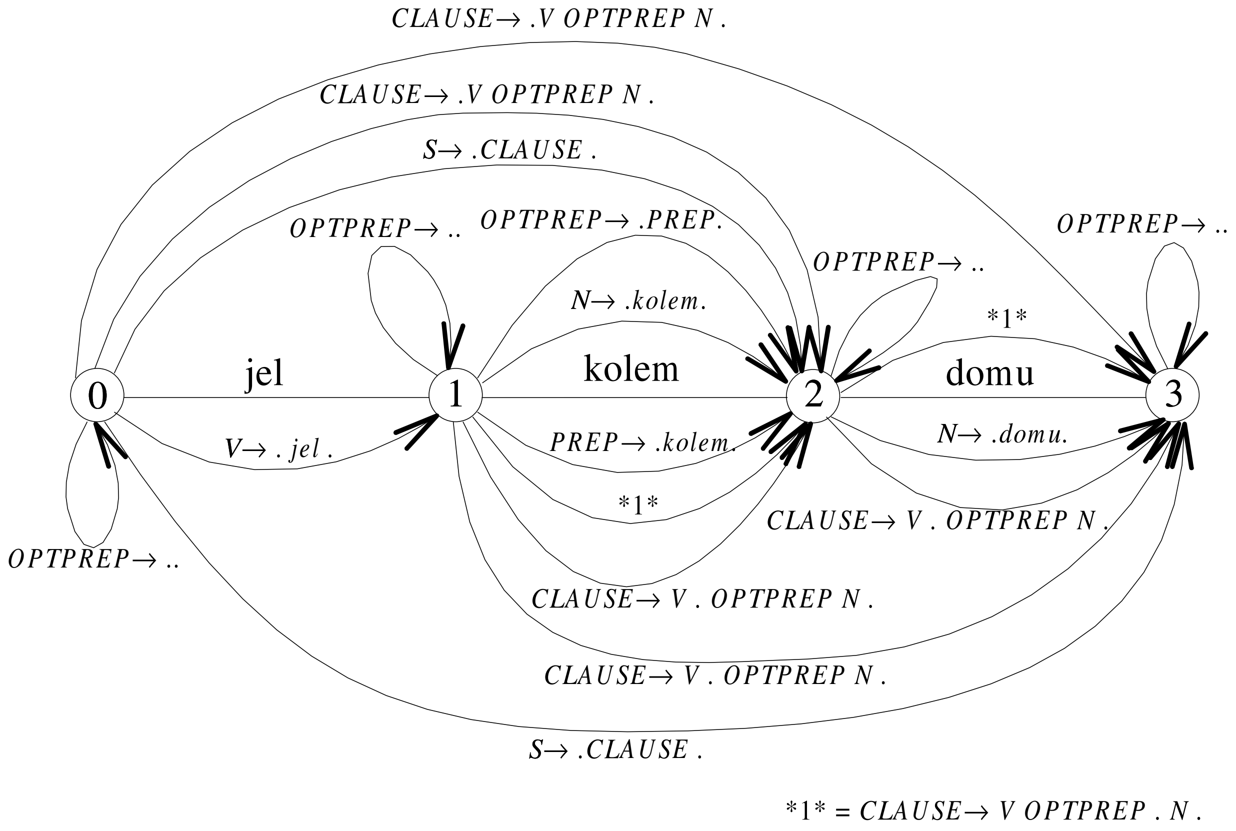
b_2) pokud je E ve tvaru $[B\rightarrow \beta A_\bullet\gamma_\bullet\delta,\ k,\ l]$, potom pro každou hranu $[A\rightarrow {}_\bullet\alpha_\bullet,\ j,\ k]$ v chartu vytvoř hranu $[B\rightarrow \beta_\bullet A\gamma_\bullet\delta,\ j,\ l]$.

c_1) pokud je E ve tvaru $[A\rightarrow \beta a_{i\bullet}\gamma_\bullet\delta,\ i,\ j]$, potom vytvoř hranu $[A\rightarrow \beta_\bullet a_i\gamma_\bullet\delta,\ i\text{-}1,\ j]$.

c_2) pokud je E ve tvaru $[A\rightarrow \beta_\bullet\gamma_\bullet a_{j+1}\delta,\ i,\ j]$, potom vytvoř hranu $[A\rightarrow \beta_\bullet\gamma a_{j+1\bullet}\delta,\ i,\ j+1]$.

d) pokud je E ve tvaru $[A\rightarrow {}_\bullet\alpha_\bullet,\ i,\ j]$, potom pro každé pravidlo $B\rightarrow \beta\ \underline{A}\ \gamma$ ve vstupní gramatice vytvoř hranu $[B\rightarrow \beta_\bullet A_\bullet\gamma,\ i,\ j]$ (symbol A je hlavou pravidla).

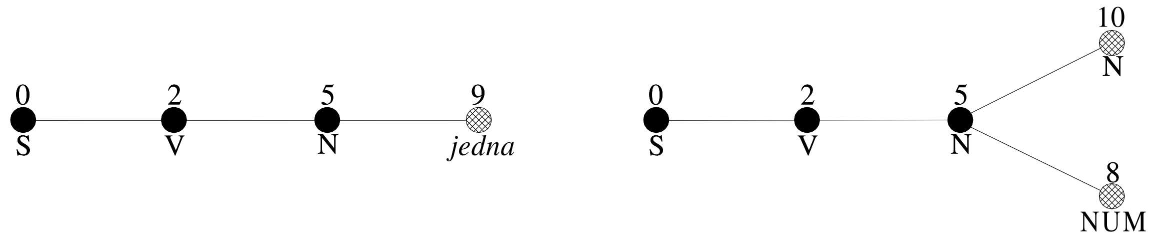
Příklad – chart po analýze řízené hlavou pravidla



Tomitův zobecněný analyzátor LR

- *generalized LR parser (GLR)*
- Masaru Tomita: Efficient parsing for natural language, 1986
- standardní LR tabulka, která může obsahovat konflikty;
- zásobník je reprezentován acyklickým orientovaným grafem (*DAG*);
- derivační stromy jsou uloženy ve sbaleném "lese" stromů.
- v podstatě stejný, jako algoritmus LR;
- udržujeme si seznam aktivních uzlů zásobníku (grafu);
- akce redukce provádíme vždy před akcemi čtení;
- akci čtení provádíme pro všechny aktivní uzly najednou;
- kde je to možné, tam uzly slučujeme.

Příklad konfliktu redukce/redukce



stav	položka	akce	symbol	další stav
5	*CLAUSE*→*V N*•*NUM*	shift	*NUM*	8
	NN→ *N*•*N*		*N*	10
	NUM→ •*jedna*		*jedna*	9
	N→ •*tramvaj*		*tramvaj*	7
	N→ •*jedna*			
9	*NUM*→*jedna*•	reduce (6)		
	N→*jedna*•	reduce (5)		

Porovnání jednotlivých algoritmů