460-4005/01: Teoretická informatika (TI)
přednáška 5

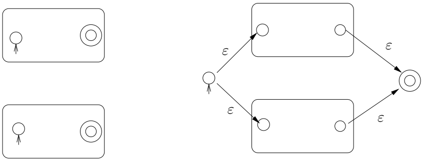
prof. RNDr Petr Jančar, CSc.

katedra informatiky FEI VŠB-TUO
www.cs.vsb.cz/jancar

LS 2010/2011

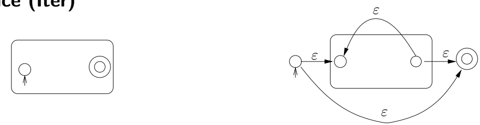
(Procedury) konstrukce ZNKA k regulárnímu výrazu

Sjednocení (Union)



Zřetězení (Conc)

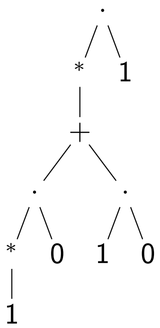
Iterace (Iter)

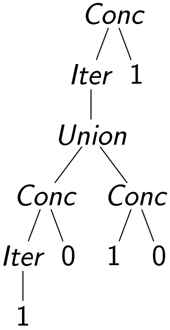


Syntaxí řízený překlad RV na ZNKA

regulární výraz $(1^*0 + 10)^*1$

syntaktický strom





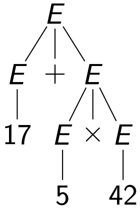
lineární zápis

$$Conc(Iter(Union(Conc(Iter(1),0),Conc(1,0))),1)$$

Syntaxe, sémantika

Gramatika slouží nejen pro popis syntaxe jazyka; syntaktická struktura (derivační strom) konkrétního slova je obvykle základem také pro vyhodnocení sémantiky (významu) slova.

Tím významem může být např. hodnota z nějaké množiny (domény). Např. u dříve uvedených aritmetických výrazů je to (celé) číslo či program (cílový kód) k jeho výpočtu ... (interpret, kompilátor ...)



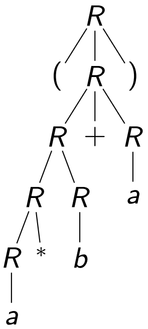
Překlad RV na (ZN)KA na základě derivačního stromu BG

Omezme se na abecedu $\Sigma = \{a, b\}$. Množina regulárních výrazů nad Σ je dána např. následující gramatikou (symboly $\emptyset$, ϵ zde neuvažujeme)

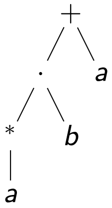
$$R \longrightarrow a \mid b \mid R + R \mid R \cdot R \mid RR \mid R^* \mid (R)$$

regulární výraz $(a^* b + a)$

derivační strom

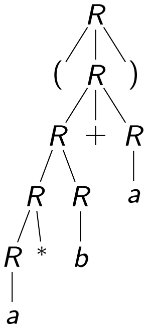


syntaktický strom



Konstrukce (ZN)KA na základě derivačního stromu

derivační strom



pro vrchol označený

- a, b: vydej příslušný (elementární) automat;
- R s následníky $R, +, R$: sestav automat A_1 pro 1. následníka, A_2 pro 3. následníka, a pak sestav a vydej automat A pro $L(A_1) \cup L(A_2)$;
- R s následníky $R, \cdot, R$ (podobně pro R, R): sestav automat A_1 pro 1. následníka, A_2 pro 3. následníka, a pak sestav a vydej automat A pro $L(A_1) \cdot L(A_2)$;
- R s následníky $R, {}^*$: sestav automat A_1 pro 1. následníka, a pak sestav a vydej automat A pro $L(A_1)^*$;
- R s následníky $(, R,)$: vydej automat sestavený pro 2. následníka.

Jednoznačné gramatiky a jazyky

BG G je *jednoznačná* $\Leftrightarrow_{df}$ každé slovo z $L(G)$ má právě jeden derivační strom (tj. právě jednu levou derivaci).
V opačném případě je G *nejednoznačná* (či *víceznačná*).

Bezkontextový jazyk L je *jednoznačný* $\Leftrightarrow_{df}$ ex. jednoznačná G tž. $L(G) = L$; jinak se L nazývá (*vnitřně*) *nejednoznačný* (*víceznačný*).

Např.:

$L_1 = \{ a^n b^n \mid n \geq 0 \}$: $\quad S \longrightarrow aSb \mid \varepsilon$ (je jednoznačný)

$L_2 = \{ a^i b^j c^k \mid (i = j) \vee (j = k) \}$:

$$S \longrightarrow S_1 C \mid AS_2$$

$$S_1 \longrightarrow aS_1 b \mid \varepsilon \qquad S_2 \longrightarrow bS_2 c \mid \varepsilon$$

$$C \longrightarrow c\,C \mid \varepsilon \qquad A \longrightarrow aA \mid \varepsilon$$

Fakt: Neex. jednoznačná BG G tž. $L(G) = L_2$. (L_2 je víceznačný.)

Pozn.: problém jednoznačnosti bezkontextové gramatiky je algoritmicky nerozhodnutelný (ukážeme později ...).

Jednoznačné gramatiky a jazyky

K (nejednoznačné) gramatice

$$R \longrightarrow a \mid b \mid R + R \mid RR \mid R^* \mid (R)$$

lze sestrojit ekvivalentní gramatiku, která je jednoznačná:

$$\begin{aligned}
R &\longrightarrow T + R \mid T \\
T &\longrightarrow FT \mid F \\
F &\longrightarrow F^* \mid (R) \mid C \\
C &\longrightarrow a \mid b
\end{aligned}$$

Gramatika pro booleovské formule (cvičení)

Uvažujme jazyk sestávající ze všech booleovských formulí s proměnnými $x1, x2, \ldots$ a logickými spojkami $\neg, \wedge, \vee$; mohou se v nich používat závorky $(,)$, ale není nutné plně závorkovat. Každá taková formule je tedy řetězcem v abecedě

$$\Sigma = \{\, x, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \neg, \wedge, \vee, (,)\,\}\,;$$

jako příklad může sloužit řetězec $(\neg x15 \vee x2 \wedge x5) \wedge \neg x21 \vee \neg(x2 \vee x5)$, který do jazyka patří. (Samozřejmě zde můžeme preferovat přehlednější zápis $(\neg x_{15} \vee x_2 \wedge x_5) \wedge \neg x_{21} \vee \neg(x_2 \vee x_5)$, ale to není podstatné.)
Navrhněte co nejjednodušší bezkontextovou gramatiku generující uvedený jazyk.
Takto navržená (jednoduchá) gramatika asi není jednoznačná; ověřte.
Zkonstruujte pak pro stejný jazyk jednoznačnou gramatiku, u níž derivační stromy přirozeně odpovídají obvyklé prioritě operátorů: negace váže silněji než konjunkce a konjunkce váže silněji než disjunkce.

LR syntaktická analýza (pravá derivace pozpátku)

$$R \longrightarrow T + R \mid T$$
$$T \longrightarrow FT \mid F$$
$$F \longrightarrow F^* \mid (R) \mid C$$
$$C \longrightarrow a \mid b$$

vrchol zásobníku	další čtený symbol	redukce podle pravidla či přesun
a		$C \longrightarrow a$
C		$F \longrightarrow C$
F	$*$	přesun a $F \longrightarrow F^*$
F	$+,)$, konec	$T \longrightarrow F$
F	$a, b, ($	přesun
FT		$T \longrightarrow FT$
T (ne FT)	$+$	přesun
T (ne FT)	ne $+$	$R \longrightarrow T$
b		$C \longrightarrow b$
$T + R$		$R \longrightarrow T + R$
(R)		$F \longrightarrow (R)$
	(jinak přesun)	

$$
\begin{aligned}
R &\longrightarrow T + R \mid T \\
T &\longrightarrow FT \mid F \\
F &\longrightarrow F^* \mid (R) \mid C \\
C &\longrightarrow a \mid b
\end{aligned}
$$

- $First(\varepsilon) = \{\varepsilon\}$
- $a \in \Sigma, \beta \in (\Pi \cup \Sigma)^* \Longrightarrow First(a\beta) = \{a\}$
- $(X \rightarrow \beta) \in P, a \in First(\beta) \Longrightarrow a \in First(X)$
  (tedy $First(\beta) \subseteq First(X)$)
- $a \in \Sigma, a \in First(X), \beta \in (\Pi \cup \Sigma)^* \Longrightarrow a \in First(X\beta)$
- $\varepsilon \in First(X), \beta \in (\Pi \cup \Sigma)^* \Longrightarrow First(\beta) \subseteq First(X\beta)$

- $\varepsilon \in Follow(S)$
- $(X \rightarrow \alpha Y \beta) \in P \Longrightarrow Follow(Y) \supseteq First(\beta \cdot Follow(X))$