5. přednáška LS 2006/2007

- Bezkontextové gramatiky a jazyky ...
- Derivační stromy
- Jednoznačnost a nejednoznačnost gramatik a jazyků
- Syntaxí řízený překlad ...
- 'Překlad' regulárních výrazů na konečné automaty
- Realizace levého průchodu derivačním stromem ...
- Zásobníkové automaty
-
- Ukázka 2. zápočtové písemky

Aritmetické výrazy v abecedě $\{ a, +, \times, (,) \}$
Například: $a + a \times a$, $(a + a) \times a$

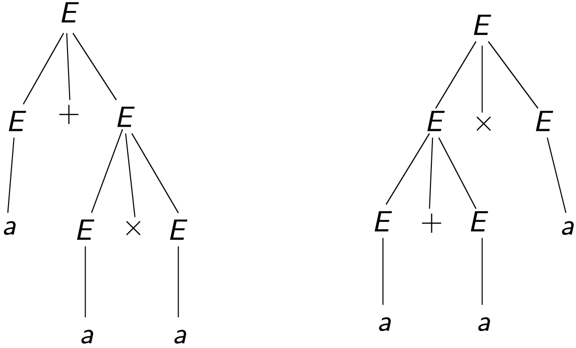
$$\langle EXPR \rangle \longrightarrow a$$
$$\langle EXPR \rangle \longrightarrow \langle EXPR \rangle + \langle EXPR \rangle$$
$$\langle EXPR \rangle \longrightarrow \langle EXPR \rangle \times \langle EXPR \rangle$$
$$\langle EXPR \rangle \longrightarrow (\langle EXPR \rangle)$$

$$E \longrightarrow a \mid E + E \mid E \times E \mid (E)$$

Odvození (*derivace*), slova $a + a \times a$:
$E \Rightarrow E + E \Rightarrow a + E \Rightarrow a + E \times E \Rightarrow a + a \times E \Rightarrow a + a \times a$
levá derivace, *pravá* derivace ...
$E \Rightarrow E + E \Rightarrow E + E \times E \Rightarrow E + E \times a \Rightarrow E + a \times a \Rightarrow a + a \times a$
A ještě příklad derivace, která není ani levá ani pravá:
$E \Rightarrow E + E \Rightarrow E + E \times E \Rightarrow E + a \times E \Rightarrow a + a \times E \Rightarrow a + a \times a$

Derivační strom

strom odvození (*derivační strom*)



Těmto různým stromům odpovídají různé *levé* derivace:

$E \Rightarrow E + E \Rightarrow a + E \Rightarrow a + E \times E \Rightarrow a + a \times E \Rightarrow a + a \times a$

$E \Rightarrow E \times E \Rightarrow E + E \times E \Rightarrow a + E \times E \Rightarrow a + a \times E \Rightarrow a + a \times a$

Bezkontextová gramatika - formálně

Bezkontextová gramatika
$G = (\Pi, \Sigma, S, P)$:
Π ... konečná množina *neterminálů*,
Σ ... konečná množina *terminálů* $\quad (\Pi \cap \Sigma = \emptyset)$,
$S \in \Pi$ ... *počáteční neterminál*
P ... konečná množina pravidel typu

$$A \rightarrow \beta, \quad kde \quad A \in \Pi, \, \beta \in (\Pi \cup \Sigma)^*.$$

Relace $\Rightarrow$ (resp. $\Rightarrow_G$) na $(\Pi \cup \Sigma)^*$:
$\gamma \Rightarrow \delta$ právě když $\gamma = \mu_1 A \mu_2$, $\delta = \mu_1 \beta \mu_2$, $(A \rightarrow \beta) \in P$

$\Rightarrow^*$ ... refl. a tranz. uzávěr $\Rightarrow$
Tedy $\gamma \Rightarrow^* \delta \quad \Leftrightarrow_{df}$ ex. posloupnost $\mu_0, \mu_1, \ldots, \mu_n$ tak, že
$\gamma = \mu_0 \Rightarrow \mu_1 \Rightarrow \cdots \Rightarrow \mu_n = \delta$.
(*Derivace* (délky n) slova δ ze slova γ.)

Jazyk generovaný gramatikou G: $L(G) = \{\, w \in \Sigma^* \mid S \Rightarrow^* w \,\}$

Jazyk L je *bezkontextový* $\quad \Leftrightarrow_{df}$ ex. BG G tak, že $L(G) = L$.

Značení:
$a, b, c, \ldots$ ... terminály
$u, v, w, \ldots$ ... řetězce terminálů
$A, B, C, \ldots, X, Y, Z$ ... neterminály
$\alpha, \beta, \gamma, \ldots$ ... řetězce neterminálů a terminálů

Levá derivace

$\alpha \Rightarrow^L \beta$ (*levé přepsání*) $\Leftrightarrow_{df}$
lze psát $\alpha = uX\delta$ a $\beta = u\gamma\delta$, kde $(X \rightarrow \gamma) \in P$
$(u \in \Sigma^*,\ \delta \in (\Pi \cup \Sigma)^*)$.

$\alpha_0 \Rightarrow \alpha_1 \Rightarrow \cdots \Rightarrow \alpha_n$ je *levou derivací* $\Leftrightarrow_{df}$ $\alpha_i \Rightarrow^L \alpha_{i+1}$ $(0 \leq i \leq n-1)$.

Derivační strom - formálně

Derivační strom (vztahující se ke $G = (\Pi, \Sigma, S, P)$), je uspořádaný kořenový strom, v němž
- vrcholy ohodnoceny prvky $\Pi \cup \Sigma$,
- kořen ohodnocen S,
- vrchol ohodnocený $X\,(\in \Pi)$ má následníky ohodnocené $Y_1, Y_2, \ldots, Y_n$ $(Y_i \in \Pi \cup \Sigma)$, kde $(X \to Y_1 Y_2 \ldots Y_n) \in P$,
- vrchol ohodnocený $a\,(\in \Sigma)$ je listem.

Derivační strom pro $w = a_1 a_2 \ldots a_n$ ... má listy zleva doprava ohodnoceny $a_1, a_2, \ldots, a_n$.

Věta. Každému derivačnímu stromu pro slovo w (v dané gramatice G) přirozeně odpovídá právě jedná levá derivace slova w (vyvozená z levého průchodu stromem); naopak každé levé derivaci slova w přirozeně odpovídá právě jeden derivační strom pro w.

Gramatika ovšem může sloužit nejen pro popis (syntaxe) jazyka; syntaktická struktura (derivační strom) konkrétního slova má obvykle sloužit také pro vyhodnocení sémantiky (významu) slova.

Tím významem může být např. hodnota z nějaké množiny (domény). Např. u dříve uvedených aritmetických výrazů je to (řekněmě přirozené) číslo.

Každému vrcholu derivačního stromu podle gramatiky

$$E \longrightarrow a \mid E + E \mid E \times E \mid (E)$$

můžeme přiřadit tento návod k výpočtu jeho hodnoty (tj. k výpočtu hodnoty podstromu, jehož je kořenem); návod závisí na označení vrcholu a jeho následníků:

- Je-li označen a, vydá hodnotu (číslo) přiřazené a (v této chvíli předpokládáme, že hodnota a byla už vyhodnocena dříve; teď je to pro nás elementární operace).
- Je-li označen E a následníci jsou označeni $E, +, E$, pak se vyhodnotí první a třetí následník a vydá se součet výsledných hodnot.
- Je-li označen E a následníci jsou označeni $E, \times, E$, pak se vyhodnotí první a třetí následník a vydá se součin výsledných hodnot.
- Je-li označen E a následníci jsou označeni $(, E,)$, pak se vyhodnotí druhý následník a jeho hodnota se vydá.
- Je-li označen $(,), +, \times$, nevydává se žádná hodnota.

Pro úplnost si u těch aritmetických výrazů můžeme představit, že terminál a je nahrazen neterminálem A a jsou dodána pravidla

$$
\begin{aligned}
&A \longrightarrow D \mid AD \\
&D \longrightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{aligned}
$$

Jistě snadno dodáme pravidla pro vyhodnocení vrcholů derivačního stromu označených (terminálními) symboly $0, 1, 2, \ldots, 9$ a (neterminálními) symboly A, D.
Ale: lze si rozumně představit, že ‘vyhodnocovač’ (překladač) provádí dvě fáze (případně zároveň); ilustrujme např. na rozboru řetězce

$$25801 + 562 \times 803 + (9401 \times 82 + 65).$$

V první fázi přečte zadaný řetězec konečný automat, který najde a zahlásí každý výskyt celého čísla – takový výskyt je pro druhou fázi chápán jako jediný symbol (atom, terminál) s přiřazenou hodnotou onoho čísla.

Ve druhé fázi je konstruován derivační strom (syntaktická struktura) de facto pro řetězec $a_1 + a_2 \times a_3 + (a_4 \times a_5 + a_6)$.
(Symboly $a_1, a_2, \ldots, a_6$ jsou např. uloženy v tabulce, která udává i jejich přiřazené hodnoty $25801, 562, \ldots, 65$.)

Tím jsme ilustrovali určitý interpret, který zadaný aritmetický výraz přečte a vydá jeho vypočtenou hodnotu.
Jinou možností je překladač (kompilátor); sémantika (cílový objekt) není v tomto případě číselná hodnota, ale např. program v jazyku nižší úrovně, který ji počítá.

Ilustrujme si takový překladač:

Role bezkontextových gramatik při překladu

Překlad přiřazovacích příkazů typu
$V := E$

$$S \longrightarrow \langle id \rangle := E$$
$$E \longrightarrow \langle id \rangle \mid E + E \mid E * E \mid (E)$$

do asembleru

Instrukce	Efekt
LOAD m	$c(m) \rightarrow ACC$
STORE m	$c(ACC) \rightarrow m$
ADD m	$c(ACC) + c(m) \rightarrow ACC$
MPY m	$c(ACC) * c(m) \rightarrow ACC$
LOAD $= m$	$m \rightarrow ACC$
ADD $= m$	$c(ACC) + m \rightarrow ACC$
MPY $= m$	$c(ACC) * m \rightarrow ACC$

Např. $\quad Zisk := (Cena + Dan) * 0.12$

$LOAD = 0.12$;
$STORE\ \$2$;
$LOAD\ Dan$;
$STORE\ \$1$;
$LOAD\ Cena$;
$ADD\ \$1$;
$MPY\ \$2$;
$STORE\ Zisk$

Práce překladače:

- lexikální analýza
- syntaktická analýza
- generování kódu

Důležité 'zotavení z chyb' (error recovery) zde pomíjíme.

Pro

$$Zisk := (Cena + Dan) * 0.12$$

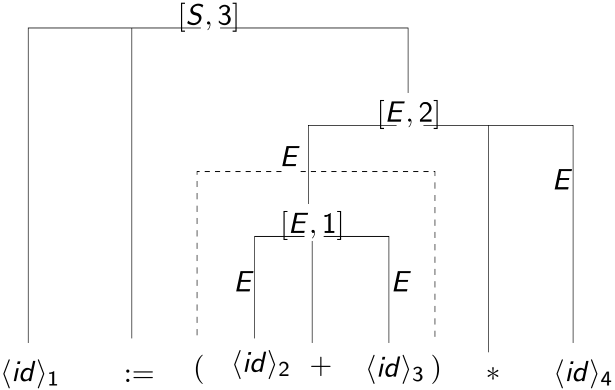
je výsledkem lexikální analýzy

$$\langle id \rangle_1 := (\langle id \rangle_2 + \langle id \rangle_3) * \langle id \rangle_4$$

zároveň s tabulkou TAB:

Poř. číslo	Identifikátor	Informace
1	Zisk	prom. real
2	Cena	prom. real
3	Dan	prom. real
4	0.12	konst. real

Pro
$\langle id\rangle_1 := (\langle id\rangle_2 + \langle id\rangle_3) * \langle id\rangle_4$
je výsledkem syntaktické analýzy

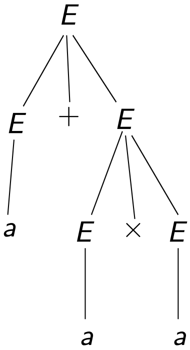


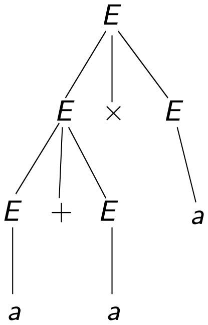
Pravidla pro ‘vyhodnocení’ vrcholů derivačního stromu

Vyhodnotit vrchol u znamená sestavit a vydat kód $Cod(u)$

- u ohodnocen $\langle id \rangle_i$:
  $Cod(u)$ z TAB (např. pro $\langle id_1 \rangle$ ‘Zisk’, pro $\langle id \rangle_4$ ‘$= 0.12$’)
- u ohodnocen $:=, +, *$: $Cod(u)$ je prázdný
- u s následníky u_1, u_2, u_3 ohodnocen $[S, n]$: $Cod(u)$ je
  ‘LOAD’ $Cod(u_3)$ ‘; STORE’ $Cod(u_1)$
- u s následníky u_1, u_2, u_3 ohodnocen $[E, n]$, přičemž u_2 ohodnocen $+$:
  $Cod(u)$ je
  $Cod(u_3)$ ‘; STORE \$’n ‘; LOAD’ $Cod(u_1)$ ‘; ADD \$’n
- podobně pro $*$: $Cod(u)$ je
  $Cod(u_3)$ ‘; STORE \$’n ‘; LOAD’ $Cod(u_1)$ ‘; MPY \$’n

Existence dvou různých derivačních stromů (levých derivací) pro jedno a totéž slovo . . . pro překlad (vyhodnocení sémantiky) závada!
Připomeňme si





u gramatiky

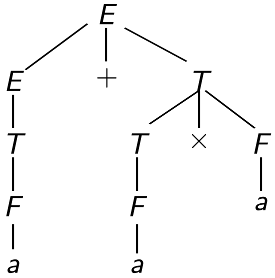
$$E \longrightarrow a \mid E + E \mid E \times E \mid (E)$$

‘Opravená’ (jednoznačná) gramatika (1. verze)

$$
\begin{aligned}
E &\longrightarrow E + T \mid T \\
T &\longrightarrow T \times F \mid F \\
F &\longrightarrow (E) \mid a
\end{aligned}
$$

Jediná levá derivace pro $a + a \times a$:
$E \Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \Rightarrow a + T \Rightarrow a + T \times F \Rightarrow a + F \times F \Rightarrow a + a \times F \Rightarrow a + a \times a$

Jediný derivační strom pro $a + a \times a$:



Jednoznačné gramatiky a jazyky

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

‘Překlad’ regulárních výrazů na konečné automaty

Omezme se na abecedu $\Sigma = \{a, b\}$. Množina regulárních výrazů nad Σ je dána např. gramatikou

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

Pravidla pro konstrukci konečného automatu použitelná na zadaný derivační strom jsou jasná: pro vrchol označený

- $\emptyset$, ε, 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 $R, \cdot, R$:
  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.

Předchozí návod lze zase chápat jako interpret (rovnou vydá příslušný automat); lze ovšem opět upravit na kompilátor – vydávající jen návod k sestavení výsledného automatu.

Všimněme si ale, že gramatika

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

není jednoznačná.
To lze vyřešit např. takto:

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

Zůstává ovšem důležitý problém: jak konstruovat derivační strom k zadanému řetězci ?
Vlastně nemusíme strom fyzicky zkonstruovat, stačí umět realizovat např. levý průchod oním stromem!
No jo, ale to jde přece pomocí zásobníku ('stacku'), ne?

Ukázka 2. zápočtové písemky

Příklad 1 je zamýšlen asi na 15 minut. Bude prověřovat např. schopnost konstrukce jednoduché bezkontextové gramatiky (BG) či zásobníkového automatu (ZA), nebo převod BG na ZA, apod.

---

1. Navrhněte bezkontextovou gramatiku G tak, že $L(G) = L_1 \cdot L_2$ kde
$L_1 = \{w \in \{a, b\}^* \mid w \text{ obsahuje podřetězec } bab\}$
$L_2 = \{a^n u \mid u \in \{a, b\}^* \text{ a } 1 \leq n \leq \text{délka}(u) \leq 2n\ \}$
Přitom použijte S jako počáteční neterminál, a dále neterminály A, B, C.
Musíte navrhnout právě jedno pravidlo s S na levé straně; snažte se, aby jste použili celkově nejvýše 10 pravidel.

Každý z příkladů 2,3,4 je zamýšlen na 3-4 minuty a je hodnocen 1 bodem. Tyto příklady mají prověřit kratší formou něco z výše uvedeného a také např. znalost uzávěrových vlastností třídy bezkontextových jazyků, schopnost zařazení jazyků do (Chomského) hierarchie (regulární, bezkontextové neregulární, nebezkontextové), porozumění obecným operacím s jazyky apod.

---

2. Definujte použitím pojmu pravé odvození, co to znamená, že

- gramatika G (rozumí se bezkontextová) je víceznačná

- bezkontextový jazyk L je víceznačný

3. Vyjmenujte, které z daných jazyků ($|w|$ označuje délku slova w, $|w|_a$ označuje počet výskytů symbolu a ve w; w^R značí zrcadl. obraz slova w)

$$L_1 = \{w \in \{a\}^* \mid |w| \textit{ je druhou mocninou celého čísla } \},$$
$$L_2 = \{0^m 1^n 0^m \mid n \leq m\},$$
$$L_3 = \{w \in \{a,b\}^* \mid |w|_a \textit{ je dělitelné pěti } \},$$
$$L_4 = \{w \in \{a,b\}^* \mid w \textit{ obsahuje podslovo abba nebo končí bbb}\}$$

jsou regulární:

jsou bezkontextové, ale ne regulární:

nejsou bezkontextové:

---

4. Charakterizujte co nejjednodušeji jazyk $\{a\}^* \backslash L$, (kde $\backslash$ označuje operaci levého kvocientu):

$\{a\}^* \backslash L = \{w \mid \ldots$

a uveďte příklad jazyků L_1, L_2, kde $\{a\}^* \backslash L_1 = L_1$ a $\{a\}^* \backslash L_2 \neq L_2$.