

Bezkontextové jazyky, derivačné stromy, redukovaná gramatika

Ing. Viliam Hromada, PhD.

C-510
Ústav informatiky a matematiky
FEI STU

`viliam.hromada@stuba.sk`



Pre pripomenutie:

- **Regulárne jazyky** - jazyky generovateľné pomocou tzv. regulárnych gramatík.
- **Bezkontextové jazyky** - jazyky generovateľné pomocou tzv. bezkontextových gramatík.

Bezkontextové jazyky

- Regulárne jazyky sa ľahko spracujú z výpočtového hľadiska, ich použitie je však obmedzené - napr. popis a identifikácia základných symbolov programovacích jazykov.
- Na špecifikáciu zložitejších konštrukcií programovacích jazykov (výrazy, deklarácie, podprogramy) sa používajú bezkontextové jazyky, resp. gramatiky.
- Pre pripomenutie, bezkontextová gramatika $G = (N, T, P, S)$ obsahuje pravidlá v tvare:

$$A \rightarrow \alpha,$$

kde $A \in N, \alpha \in (N \cup T)^*$.



Príklad bezkontextovej gramatiky

Nech $G = (\{E, T, F\}, \{(\,), +, *, \text{id}\}, P, E)$, kde pravidlá P :

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid \text{id}.$$

Zjednodušená gramatika na generovanie výrazov v programovacích jazykoch.

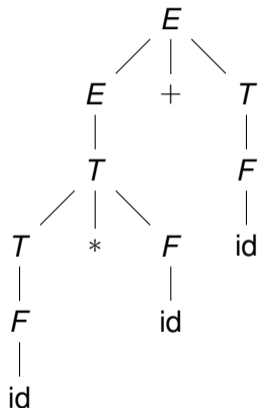
Existuje v nej napr. odvodenie:

$$E \Rightarrow E + T \Rightarrow T + T \Rightarrow T * F + T \Rightarrow F * F + T \Rightarrow \text{id} * F + T$$

$$\Rightarrow \text{id} * F + F \Rightarrow \text{id} * F + \text{id} \Rightarrow \text{id} * \text{id} + \text{id}$$



V bezkontextovej gramatike vieme ku každému odvodeniu kresliť **derivačný strom** (strom odvodenia):



Strom odvodenia

Definícia

Nech $G = (N, T, P, S)$ je bezkontextová gramatika. Ohodnotený strom sa nazýva **strom odvodenia** (derivačný strom) slova w v gramatike G , ak spĺňa nasledovné podmienky:

- Každý vrchol stromu je ohodnotený symbolom z množiny $N \cup T_\epsilon$.
- Koreň stromu je ohodnotený začiatočným symbolom S .
- Ak má nejaký vrchol stromu potomkov, potom je ohodnotený symbolom z množiny N .
- Ak X_1, X_2, \dots, X_k sú ohodnotenia priamych potomkov vrcholu A , potom v P musí existovať pravidlo $A \rightarrow X_1 X_2 \dots X_k$.
- Listy stromu sú ohodnotené symbolmi z množiny T_ϵ .
- Slovo $w \in T^*$ sa dá získať zreťazením ohodnotení listov stromu zľava doprava.



Definícia

Bezkontextová gramatika G sa nazýva **jednoznačná**, ak ku každému slovu $w \in L(G)$ existuje jediný strom ododenia. V opačnom prípade sa gramatika nazýva **nejednoznačná** (viacznačná, ambiguous).



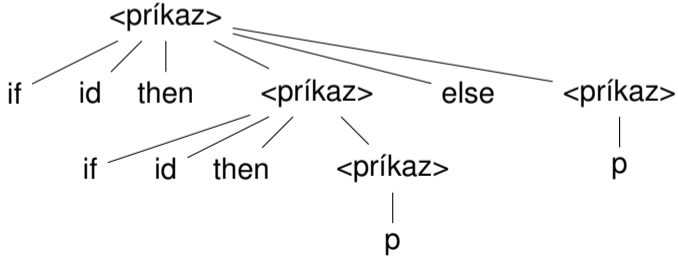
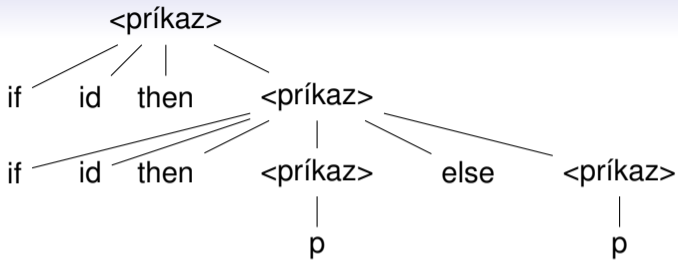
Nejednoznačná gramatika - príklad

Gramatika $G = (\{\langle \text{príkaz} \rangle\}, \{\text{id}, \text{p}, \text{if}, \text{then}, \text{else}\}, P, \langle \text{príkaz} \rangle)$, pravidlá P :

1. $\langle \text{príkaz} \rangle \rightarrow \text{if id then } \langle \text{príkaz} \rangle$
2. $\langle \text{príkaz} \rangle \rightarrow \text{if id then } \langle \text{príkaz} \rangle \text{ else } \langle \text{príkaz} \rangle$
3. $\langle \text{príkaz} \rangle \rightarrow \text{p}$

Zjednodušená gramatika reprezentujúca podmienené príkazy, **id** zastupuje identifikátor (premennú) predstavujúci podmienku, **p** zastupuje nejaký príkaz. Potom pre slovo (reťazec terminálov): **if id then if id then p else p** existujú 2 rôzne stromy odvodenia:





Gramatika pre regulárne výrazy nad $A = \{a, b\}$

Gramatika $G = \{\{R\}, \{*, |, a, b\}, P, R\}$, kde pravidlá:

$R \rightarrow R|R$ (zjednotenie)

$R \rightarrow RR$ (zreťazenie)

$R \rightarrow R^*$ (iterácia)

$R \rightarrow a$

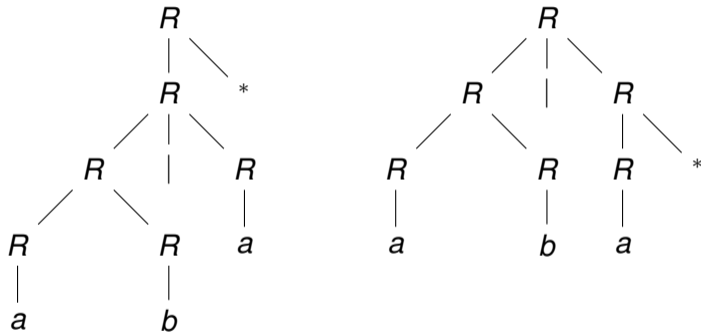
$R \rightarrow b$

$R \rightarrow \varepsilon$



Gramatika pre regulárne výrazy nad $A = \{a, b\}$

V takejto gramatike by mal regulárny výraz $ab|a^*$ viacero derivačných stromov, napr.



Čo by teoreticky mohlo znamenať nejednoznačnosť príslušného regulárneho výrazu pri jeho počítačovom spracovaní, pretože prvý regulárny výraz vyzerá ako $((ab)|a)^*$ a druhý ako $(ab)|(a^*)$.

Jednoznačné regexy

Našťastie, v tomto prípade existuje aj jednoznačná gramatika pre regulárne výrazy. Uvedieme príklad gramatiky pre regexy, ktoré budú popisovať reťazce nad abecedou $\{a, b\}$. **POZOR!!! Všimnite si, že v tejto gramatike sú zvislá čiara $|$, hviezdica či ε terminálne symboly - čo je v protiklade s tým, ako sme ich doteraz používali!**. Je to dané tým, že chceme popísať gramatiku, ktorá generuje reťazce predstavujúce regulárne výrazy a v nich majú tieto symboly svoj špecifický význam.

Pre jednoduchosť, na rozdiel od klasickej definície regexov, naša gramatika nebude vedieť generovať regexy, ktoré obsahujú symbol prázdnej množiny.

Gramatika $G =$

$(\{\langle \text{Regex} \rangle, \langle \text{Term} \rangle, \langle \text{Factor} \rangle, \langle \text{Base} \rangle, \langle \text{Char} \rangle\}, \{*, |, a, b, \varepsilon, (,)\}, P, \langle \text{Regex} \rangle)$,

kde pravidlá sú na ďalšom slajde:



Jednoznačné regexy

$\langle \text{Regex} \rangle \rightarrow \langle \text{Term} \rangle$

$\langle \text{Regex} \rangle \rightarrow \langle \text{Term} \rangle \mid \langle \text{Regex} \rangle$

$\langle \text{Term} \rangle \rightarrow \langle \text{Factor} \rangle$

$\langle \text{Term} \rangle \rightarrow \langle \text{Factor} \rangle \langle \text{Term} \rangle$

$\langle \text{Factor} \rangle \rightarrow \langle \text{Base} \rangle$

$\langle \text{Factor} \rangle \rightarrow \langle \text{Base} \rangle^*$

$\langle \text{Base} \rangle \rightarrow (\langle \text{Regex} \rangle)$

$\langle \text{Base} \rangle \rightarrow \langle \text{Char} \rangle$

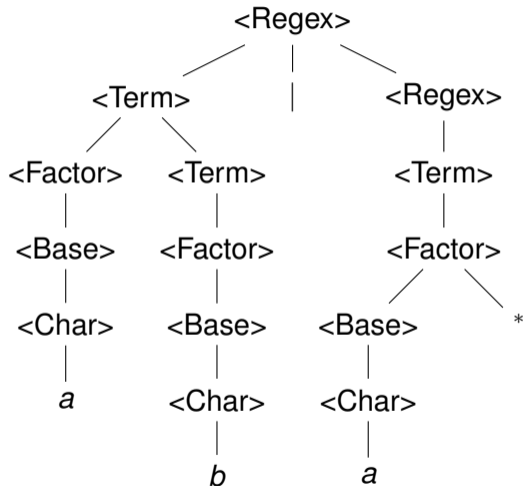
$\langle \text{Char} \rangle \rightarrow a$

$\langle \text{Char} \rangle \rightarrow b$

$\langle \text{Char} \rangle \rightarrow \varepsilon$



Derivačný strom reťazca $ab \mid a^*$ v tejto gramatike:



Problém určenia nejednoznačnosti gramatiky

- Ak máme zadanú nejakú bezkontextovú gramatiku $G = (N, T, P, S)$, tak problém zistenia, či je gramatika **jednoznačná/nejednoznačná** je **vo všeobecnosti nerozhodnuteľný!**
- To znamená, že **neexistuje** algoritmus, ktorý by pre ľubovoľnú bezkontextovú gramatiku vedel v konečnom čase povedať, či je jednoznačná alebo nejednoznačná!



- Nejednoznačné gramatiky predstavujú problém - ak existujú pre 1 slovo viaceré stromy odvodenia, asi má slovo viacero sémantických interpretácií.
- Je zrejmé, že odvodenie slova a derivačný strom sú úzko prepojené - ak existuje odvodenie, existuje i derivačný strom (a naopak).
- Pri bezkontextových gramatikách si môžeme vybrať, na ktorý neterminál vo vetnej forme aplikujeme príslušné prepisovacie pravidlo v danom kroku odvodenia - bez rizika, že tým ovplyvníme výsledok.



2 špeciálne odvodenia

Príklad: Všimnite si tieto 2 odvodenia:

1. $E \Rightarrow E + T \Rightarrow T + T \Rightarrow T * F + T \Rightarrow F * F + T \Rightarrow \text{id} * F + T \Rightarrow \text{id} * \text{id} + T \Rightarrow \text{id} * \text{id} + F \Rightarrow \text{id} * \text{id} + \text{id}$
2. $E \Rightarrow E + T \Rightarrow E + F \Rightarrow E + \text{id} \Rightarrow T + \text{id} \Rightarrow T * F + \text{id} \Rightarrow T * \text{id} + \text{id} \Rightarrow F * \text{id} + \text{id} \Rightarrow \text{id} * \text{id} + \text{id}$

V odvodení č. 1 sme nahradili vždy prvý neterminál zľava, v odvodení č. 2 sme nahradili vždy prvý neterminál sprava. Oba derivačné stromy by však boli rovnaké (a také isté, ako derivačný strom na slajde č. 5).



Ľavé (pravé) odvodenie

Definícia

*Nech G je bezkontextová gramatika. Potom odvodenie $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \dots \Rightarrow \alpha_k$ v G , kde každý krok odvodenia $\alpha_i \Rightarrow \alpha_{i+1}$ pre $0 \leq i \leq k$ realizujeme tak, že v reťazci α_i nahradíme prvý neterminál zľava, nazývame **ľavé odvodenie (ľavá derivácia)**; ak nahradíme prvý neterminál sprava, nazývame **pravé odvodenie (pravá derivácia)**. Príslušné vznikajúce vetné formy nazývame **ľavé (pravé) vetné formy**. Krok odvodenia v ľavom (pravom) odvodení budeme označovať symbolom \Rightarrow_l (\Rightarrow_r).*



L'avé (pravé) odvodenie

Teda podľa uvedeného značenia:

$$1. E \Rightarrow_l E + T \Rightarrow_l T + T \Rightarrow_l T * F + T \Rightarrow_l F * F + T \Rightarrow_l id * F + T \Rightarrow_l id * id + T \Rightarrow_l id * id + F \Rightarrow_l id * id + id$$

$$2. E \Rightarrow_r E + T \Rightarrow_r E + F \Rightarrow_r E + id \Rightarrow_r T + id \Rightarrow_r T * F + id \Rightarrow_r T * id + id \Rightarrow_r F * id + id \Rightarrow_r id * id + id$$



Ak G je jednoznačná gramatika, potom ľavé (pravé) odvodenie každého slova je určené jednoznačne. Túto vlastnosť využívajú **algoritmy hľadajúce deriváciu slova v gramatike - tzv. syntaktické analyzátory**.

Transformácie gramatík

- Gramatiky možno **transformovať** na **ekvivalentné** gramatiky (t.j. generujúce ten istý jazyk), ktoré budú navyše **spĺňať** dodatočné **požiadavky** (napr. na tvar pravidiel).
- Základom pri transformácii gramatík je, aby sa po transformácii **nezmenil jazyk generovaný gramatikou**.



3 základné úpravy

Existujú 3 základné úpravy bezkontextových gramatík:

1. Odstránenie nadbytočných symbolov - toto si vysvetlíme.
2. Odstránenie ε -pravidiel - zatiaľ necháme tak...
3. Odstránenie jednoduchých pravidiel - zatiaľ necháme tak...



Odstránenie nadbytočných symbolov

Môže sa stať, že pri definícii gramatiky sa použijú symboly (či už N alebo T), ktoré sa alebo nikdy počas derivácie nepoužijú, alebo ak sa použijú, nikdy sa nepodarí odvodiť reťazec terminálov. Ich prítomnosť v gramatike potom zbytočne zväčšuje jej veľkosť a navyše môžu viesť k zlým výsledkom rôznych algoritmov, pracujúcich s gramatikami.

Definícia

Nech $G = (N, T, P, S)$ je bezkontextová gramatika. Symbol $X \in N \cup T$ sa nazýva:

- **nedostupný**, ak v G neexistuje odvodenie $S \Rightarrow^* \alpha X \beta$ pre nejaké $\alpha, \beta \in (N \cup T)^*$,
- **nadbytočný**, ak v G neexistuje odvodenie $S \Rightarrow^* xXz \Rightarrow^* xyz$ pre nejaké $x, y, z \in T^*$.



Odstránenie nadbytočných symbolov

Definícia

*Gramatika bez nadbytočných symbolov sa nazýva **redukovaná** gramatika.*

Každý nedostupný symbol je zároveň aj nadbytočný. Vytvorenie redukovanej gramatiky pozostáva z 2 hlavných fáz:

1. Odstránenie neterminálov, z ktorých nie je možné odvodiť reťazec terminálov alebo ε .
2. Odstránenie neterminálov a terminálov, ktoré sa nemôžu vyskytnúť vo vetných formách gramatiky.

Symbols sa odstránia spolu s príslušnými pravidlami, v ktorých vystupujú.



Príklad

Nech gramatika G má pravidlá:

$$S \rightarrow AB \mid A$$

$$A \rightarrow AA \mid a$$

$$B \rightarrow bB$$

$$C \rightarrow c$$

Neterminál B je nadbytočný symbol. Neterminál C a terminál c sú nedostupné symboly. Po ich odstránení dostávame ekvivalentnú redukovanú gramatiku:

$$S \rightarrow A$$

$$A \rightarrow AA \mid a$$



Odstránenie nadbytočných symbolov - množina N_T

V prvej fáze hľadáme neterminály, z ktorých je možné odvodiť reťazec terminálov (prípadne ε). Hľadáme množinu:

$$N_T = \{A \mid A \Rightarrow^* w, w \in T^*\} \quad (1)$$



Odstránenie nadbytočných symbolov - množina N_T

Vstup: bezkontextová gramatika $G = (N, T, P, S)$

Výstup: množina N_T podľa (1)

1: $N_T \leftarrow \emptyset$

2: **opakuj**

3: $\dot{N}_T \leftarrow N_T$

4: $N_T \leftarrow \dot{N}_T \cup \{A \mid A \rightarrow \alpha \in P \wedge \alpha \in (\dot{N}_T \cup T)^*\}$

5: **pokiaľ** $N_T \neq \dot{N}_T$



Odstránenie nadbytočných symbolov - množina N_T

Príklad: Nech $G = (\{S, A, B, C\}, \{a, b, c\}, P, S)$, kde P obsahuje pravidlá:

$$S \rightarrow ABc \mid abA$$

$$A \rightarrow aAa \mid bBb \mid b$$

$$B \rightarrow aBb \mid caBC$$

$$C \rightarrow \varepsilon \mid ab \mid BcCa$$

Podľa algoritmu by sa premenná N_T postupne menila: $\emptyset, \{A, C\}, \{S, A, C\}$.
Neterminál B je teda taký, že sa z neho nedá odvodiť terminálny reťazec. Jeho výskyt v gramatike je teda **nadbytočný**, pretože jeho odstránením sa **nijako nezmení jazyk** $L(G)$, ktorý gramatika generuje.



Odstránením neterminálu B a pravidiel s jeho výskytom by sme dostali gramatiku:

$$S \rightarrow abA$$

$$A \rightarrow aAa \mid b$$

$$C \rightarrow \varepsilon \mid ab$$



Odstránenie nadbytočných symbolov - množina V_D

V druhej fáze hľadáme neterminály a terminály, ktoré sa môžu vyskytnúť v nejakej vetnej forme danej gramatiky, t.j. sú to **dostupné symboly gramatiky**. Hľadáme množinu:

$$V_D = \{X \mid S \Rightarrow^* \alpha X \beta, \alpha, \beta \in (N \cup T)^*\}. \quad (2)$$



Odstránenie nadbytočných symbolov - množina V_D

Vstup: bezkontextová gramatika $G = (N, T, P, S)$

Výstup: množina V_D podľa (2)

1: $V_D \leftarrow \{S\}$

2: **opakuj**

3: $\dot{V}_D \leftarrow V_D$

4: $V_D \leftarrow \dot{V}_D \cup \{X \mid A \rightarrow \alpha X \beta \in P \wedge A \in \dot{V}_D\}$

5: **pokiaľ** $V_D \neq \dot{V}_D$



Odstránenie nadbytočných symbolov - množina V_D

Príklad: Nech $G = (\{S, A, B, C, D\}, \{a, b, c, d\}, P, S)$, kde P obsahuje pravidlá:

$$S \rightarrow AB \mid abS \mid \varepsilon$$

$$A \rightarrow aAa \mid b$$

$$B \rightarrow aCb \mid cC$$

$$C \rightarrow cS$$

$$D \rightarrow abC \mid \varepsilon \mid dD$$

Podľa algoritmu by sa premenná V_D postupne menila:

$\{S\}, \{S, A, B, a, b\}, \{S, A, B, C, a, b, c\}$. Nedostupné symboly sú teda D, d .



Odstránenie nadbytočných symbolov - výsledný algoritmus

Vstup: bezkontextová gramatika $G = (N, T, P, S)$

Výstup: gramatika bez nadbytočných symbolov

- 1: Vytvor množinu N_T .
- 2: **ak** $S \in N_T$ **potom**
- 3: odstráň z gramatiky G všetky neterminály, ktoré nepatria do N_T ;
- 4: vytvor množinu V_D
- 5: odstráň z gramatiky G všetky symboly, ktoré nepatria do V_D ;
- 6: **koniec ak**

Poradie krokov je **dôležité!**. Najprv sa musia odstrániť neterminály, ktoré nevedú na terminálne reťazce (N_T) a až potom sa odstránia nedostupné symboly (V_D).



Odstránenie nadbytočných symbolov

Príklad: Pokračujme v gramatike, pre ktorú sme hľadali množinu N_T na slajde č. 28. Po odstránení neterminálu B (pretože nebol v množine N_T) dostávame gramatiku $G_1 = (\{S, A, C\}, \{a, b, c\}, P_1, S)$, kde P_1 obsahuje pravidlá:

$$S \rightarrow abA$$

$$A \rightarrow aAa \mid b$$

$$C \rightarrow ab \mid \varepsilon$$

Pre túto gramatiku je množina $V_D = \{S, A, a, b\}$, t.j. symboly C, c sú nedostupné. Po ich odstránení dostávame redukovanú gramatiku:

$$G_{red} = (\{S, A\}, \{a, b\}, \{S \rightarrow abA, A \rightarrow aAa \mid b\}, S).$$

T.j. $L(G) = L(G_1) = L(G_{red}) = \{aba^n ba^n \mid n \in \mathbb{Z}_0^+\}$.



Ak by sme gramatiku zo slajdu č. 28 upravovali **v opačnom poradí**, potom:

1. Množina $V_D = \{S, A, B, C, a, b, c\}$, t.j. všetky symboly sú dostupné a nič sa neodstráni.
2. Množina $N_T = \{A, C, S\}$, t.j. odstránime len neterminál B a výsledná gramatika:

$$S \rightarrow abA$$

$$A \rightarrow aAa \mid b$$

$$C \rightarrow \varepsilon \mid ab$$

pričom vidíme, že neterminál C nie je dostupný a terminál c ani len nevystupuje v niektorom z pravidiel.



Použitá literatúra

Dedera, L': Počítačové jazyky a ich spracovanie.

Linz, P.: An Introduction to Formal Languages and Automata.

Molnár, L': Gramatiky a jazyky.

