Automaty a gramatiky (BI-AAG)

1. Základní pojmy

Jan Holub

Katedra teoretické informatiky

Fakulta informačních technologií

ČVUT v Praze

Motivace

Motivace

- Správné použití teorie jazyků a automatů šetří čas a prostředky.
- Jedna ze základních teorií Computer Science.
- Pochopení efektivních řešení pro celou řadu základních problémů.

Cíle předmětu

- Seznámit se s teorií jazyků a automatů.
- Umět ji efektivně používat.
- Rozeznat třídy jazyků.

Hodnocení předmětu

Cvičení

- 2 domácí úkoly po 0 bodech (Je to na Vás.)
- 2 testy po 20 bodech
- celkem 40 bodů
- zápočet: minimálně 20 bodů.
- Jeden z testů je možné na konci semestru opakovat k získání zápočtu a 20 bodů.

Hodnocení předmětu

Zkouška

- „rozstřel“: max. 10 bodů
- zkouškový test: max. 50 bodů
- ústní část: ± 5 bodů + právo veta
- zkouška: rozstřel min. 7 bodů, test min. 25 bodů, projít ústní zkouškou

Upozornění

- Slajdy jsou pouze pomocným materiálem k přednáškám a nemohou být jediným zdrojem pro přípravu k testům a ke zkoušce.

Obsah předmětu

1. Základní pojmy, Chomského hierarchie.
2. Deterministické a nedeterministické konečné automaty (DKA a NKA), NKA s epsilon přechody.
3. Operace s automaty (převod na NKA bez epsilon přechodů, na DKA, minimalizace, průnik, sjednocení).
4. Reg. výrazy, převody mezi RV, KA a RG, Kleenova věta.
5. Operace s regulárními gramatikami, převody na KA.
6. Jazyky bezkontextové, zasobníkový automat.
7. Analýza bezkontextových jazyků (nedeterm. versus determ.).
8. Vlastnosti reg. jazyků (pumping lemma, Nerodova věta).
9. Rozšíření o překlad, Mealey, Moore, převody.
10. Programová realizace DKA a NKA, obvodová realizace.
11. KA jako lexikální analyzátor, lex/flex generátory.
12. Jazyky kontextové a neomezené, Turingův stroj.

Doporučená literatura

- Aho, A. V., Lam, M. S., Sethi, R., Ullman, J. D. “Compilers: Principles, Techniques, and Tools” (2nd Edition). Addison Wesley, 2007. ISBN 0321486811.
- Kozen, D. C. “Automata and Computability”. Springer, 1997. ISBN 0387949070.
- Melichar, B.: Jazyky a překlady. Praha, Vydavatelství ČVUT 2007.
- Melichar, B., Antoš, J., Holub, J., Šimánek, M.: Jazyky a překlady – cvičení. Praha, Vydavatelství ČVUT 2004.

BI-AAG v angličtině

- přednášky a cvičení v angličtině
- testy a zkouška v češtině
- menší skupina studentů
- zájemci kontaktujte doc. Jana Holuba

Základní pojmy

Abeceda — konečná množina *symbolů* (značíme Σ, někdy též T)

- binární {0,1}, ternární {Yes, No, Maybe},
  DNA {A, C, G, T}, klíčová slova {while, do, begin, end, to, for, false, true, ...}

Řetězec nad abecedou — konečná posloupnost symbolů abecedy. Např. „0110101“, „ACCCGT“, „while true do“

Prázdná posloupnost = *prázdný řetězec* = ε.

Σ^* — množina všech řetězců nad Σ

Σ^+ — množina všech neprázdných řetězců nad Σ

$\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$

Základní pojmy

Operace *zřetězení*:

- $x, y \in \Sigma^*$, připojením řetězce y za řetězec x vznikne řetězec xy
- je asociativní, t.j. $(xy)z = x(yz)$
- není komutativní, t.j. $xy \neq yx$
- ε se chová vzhledem k operaci zřetězení jako neutrální prvek: $x\varepsilon = \varepsilon x = x$

Délka řetězce x:

- značíme $|x|$
- $|x| \geq 0,\ |\varepsilon| = 0$
- $a^0 = \varepsilon,\ a^1 = a,\ a^2 = aa,\ a^3 = aaa,\ \ldots$

Formální jazyk

Formální jazyk L nad Σ: $L \subseteq \Sigma^*$ (množina řetězců).

operace:

- množinové operace *sjednocení, průnik a rozdíl*
- *komplement* (doplněk) jazyka L_1: $L_2 = \Sigma^* \setminus L_1$ ($L_2 \cup L_1 = \Sigma^*, L_1 \cap L_2 = \emptyset$).
- *součin* (zřetězení) jazyků: $L = L_1.L_2 = \{xy : x \in L_1, y \in L_2\}$ (L je definován nad abecedou $\Sigma = \Sigma_1 \cup \Sigma_2$)
- $N-$tá *mocnina* jazyka L: $L^n = L.L^{n-1}$, $L^0 = \{\varepsilon\}$.
  Iterace L^* jazyka L: $L^* = \cup L^n$, pro všechna n od nuly do nekonečna.
  $L^* = L^+ \cup \{\varepsilon\}$,
  $L^+ = L.L^* = L^*.L$ (pozitivní iterace).

Gramatika

Definice

Gramatika je čtveřice $G = (N, T, P, S)$, kde

- N je konečná množina neterminálních symbolů,
- T je konečná množina terminálních symbolů ($T \cap N = \emptyset$, T je vlastně vstupní abeceda Σ),
- P je konečná podmnožina množiny $(N \cup T)^*.N.(N \cup T)^* \times (N \cup T)^*$, (element (α, β) z P zapíšeme $\alpha \to \beta$ a nazveme *pravidlo*),
- $S \in N$ je *počáteční symbol* gramatiky (též větný nebo startovací symbol).

Gramatika

Příklad

Gramatika $G_1 = (\{A, S\}, \{0, 1\}, P, S)$, kde P:

- $S \rightarrow 0A$
- $A \rightarrow 1A$
- $A \rightarrow 0.$

Poznámka:

$\alpha \rightarrow \beta_1,\ \alpha \rightarrow \beta_2, \ldots, \alpha \rightarrow \beta_n$, lze zkrátit na:
$\alpha \rightarrow \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n.$

Gramatika

Další možnosti zápisu pomocí Backus-Naurovy formy (BNF) nebo rozšířené Backus-Naurovy formy (EBNF).

Příklad

Gramatika G generuje jazyk celých čísel bez znaménka. V této gramatice je použita BNF.

$G =$
$(\{\langle \textit{celé číslo} \rangle, \langle \textit{číslice} \rangle\}, \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, P, \langle \textit{celé číslo} \rangle)$
Množina P obsahuje pravidla:
$\langle \textit{celé číslo} \rangle \ ::= \ \langle \textit{číslice} \rangle \langle \textit{celé číslo} \rangle \mid \langle \textit{číslice} \rangle$
$\langle \textit{číslice} \rangle \ ::= \ 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9.$

Gramatika

Definice

$G = (N, T, P, S)$, $x, y \in (N \cup T)^*$. Říkáme, že x *přímo derivuje* y ($x \Rightarrow y$), jestliže existuje $\alpha \to \beta \in P$ a $\gamma, \delta \in (N \cup T)^*$ takové, že $x = \gamma\alpha\delta, y = \gamma\beta\delta$.

Příklad

$G_1 = (\{A, S\}, \{0, 1\}, P, S)$, $P = \{S \to 0A, A \to 1A, A \to 0\}$.
$01A \Rightarrow 011A$

Definice

$\alpha \Rightarrow^k \beta$, jestliže existuje posloupnost $\alpha_0, \alpha_1, \ldots, \alpha_k$, pro $k \geq 0, k+1$ řetězců takových, že $\alpha = \alpha_0, \alpha_{i-1} \Rightarrow \alpha_i$ pro $1 \leq i \leq k$, a $\alpha_k = \beta$. Tuto posloupnost nazveme *derivací* řetězce β z řetězce α, která má délku k v gramatice G.

Příklad

$G_1 \colon 01A \Rightarrow^4 011111A$

Gramatika

Definice

Tranzitivní uzávěr relace $\Rightarrow$: $\alpha \Rightarrow^+ \beta$, když $\alpha \Rightarrow^i \beta$ pro nějaké $i \geq 1$.

Definice

Tranzitivní a reflexivní uzávěr relace $\Rightarrow$: $\alpha \Rightarrow^* \beta$, když $\alpha \Rightarrow^i \beta$ pro nějaké $i \geq 0$.

Gramatika

Definice

$L(G) = \{w : w \in T^*, S \Rightarrow^* w\}$ je *jazyk generovaný gramatikou* G.

Příklad

$G_1 = (\{A, S\}, \{0, 1\}, \{S \to 0A, A \to 1A, A \to 0\}, S)$ generuje jazyk $L(G_1) = \{01^n0 : n \geq 0\}$.

V gramatice G_1 existují například derivace

$S \Rightarrow 0A \Rightarrow 00$

$S \Rightarrow 0A \Rightarrow 01A \Rightarrow 010$

$S \Rightarrow 0A \Rightarrow 01A \Rightarrow 011A \Rightarrow 0110$

Gramatika

Definice
$G = (N, T, P, S)$. Řetězec α nazveme *větnou formou* v gramatice G, jestliže platí $S \Rightarrow^* \alpha, \alpha \in (N \cup T)^*$.

Příklad
G_1: $S \Rightarrow^* 01A$

Definice
Větná forma v gramatice $G = (N, T, P, S)$, která neobsahuje neterminální symboly se nazývá *věta generovaná gramatikou* G.

Příklad
G_1: $S \Rightarrow^* 0110$

Jazyk generovaný gramatikou G je množina všech vět generovaných gramatikou G.

Gramatika

Definice

Gramatiky G_1 a G_2 jsou *ekvivalentní*, když generují stejný jazyk. To znamená, že $L(G_1) = L(G_2)$.

Klasifikace gramatik

Noam Chomsky (*7.12.1928 Philadelphia)

- práce na poli gramatik jak formálních tak i přirozených jazyků

Definice

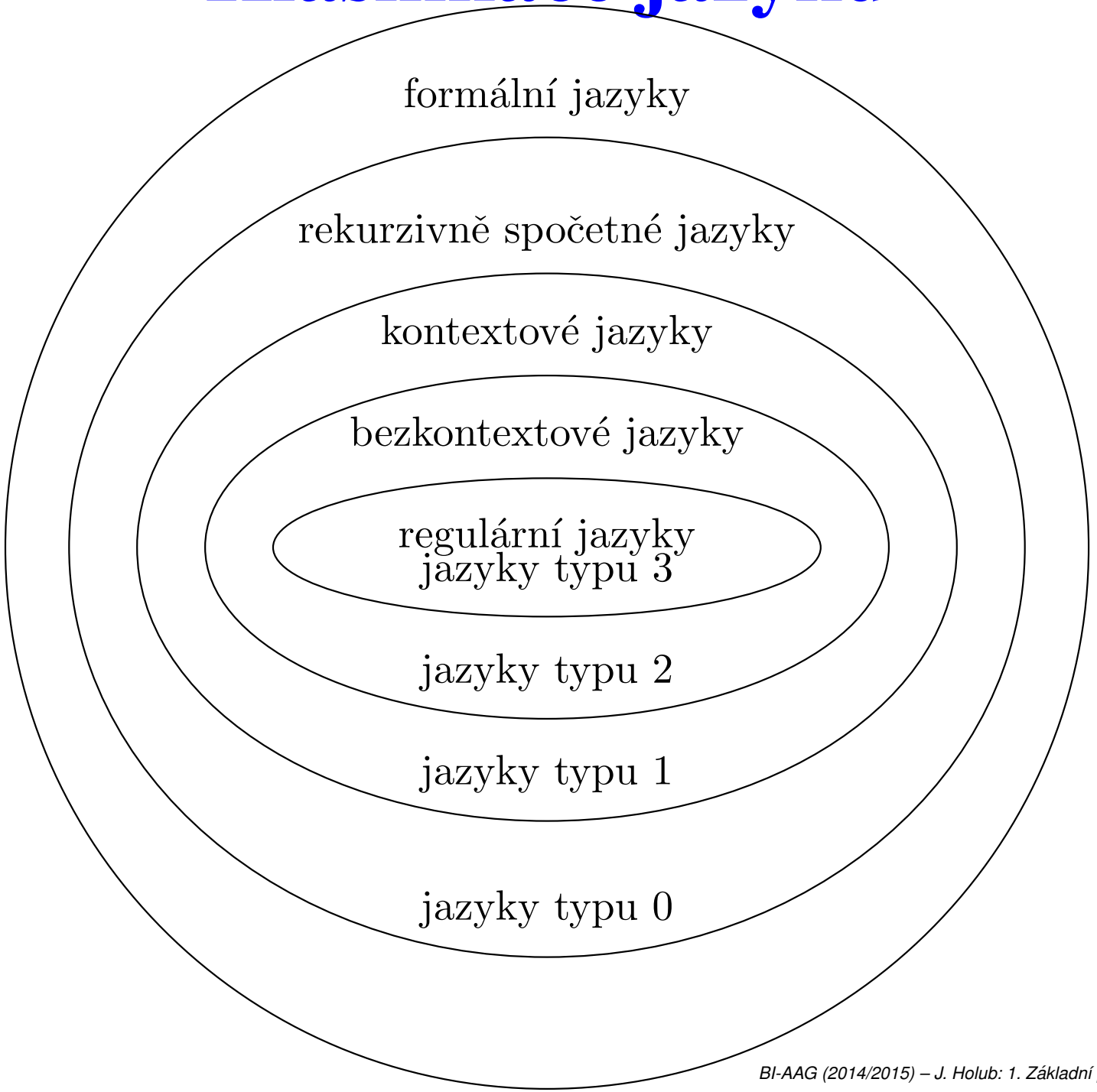
$G = (N, T, P, S)$. Říkáme, že G je:

1. *Neomezená* (typu 0), jestliže odpovídá obecné definici gramatiky.
2. *Kontextová* (typu 1), jestliže každé pravidlo z P má tvar $\gamma A\, \delta \to \gamma\alpha\delta$, kde $\gamma, \delta \in (N \cup T)^*, \alpha \in (N \cup T)^+, A \in N$. Výjimku tvoří pravidlo $S \to \varepsilon$ v případě, že S se neobjeví na pravé straně žádného pravidla.
3. *Bezkontextová* (typu 2), jestliže každé pravidlo má tvar $A \to \alpha$, kde $A \in N, \alpha \in (N \cup T)^*$.
4. *Regulární* (typu 3), jestliže každé pravidlo má tvar $A \to aB$ nebo $A \to a$, kde $A, B \in N, a \in T$. Výjimku tvoří pravidlo $S \to \varepsilon$ v případě, že S se neobjeví na pravé straně žádného pravidla.

Klasifikace jazyků

1. *Rekurzivně spočetné jazyky* (typu 0) generované neomezenými gramatikami.
   - rozpoznatelné Turingovým strojem
2. *Kontextové jazyky* (typu 1) generované kontextovými gramatikami.
   - rozpoznatelné lineárně omezeným Turingovým strojem
3. *Bezkontextové jazyky* (typu 2) generované bezkontextovými gramatikami.
   - rozpoznatelné zásobníkovým automatem
4. *Regulární jazyky* (typu 3) generované regulárními gramatikami.
   - rozpoznatelné konečným automatem

Klasifikace jazyků



Klasifikace jazyků

Příklad

$G_1 = (\{S\}, \{0,1\}, \{S \to 0S, S \to 1S, S \to 1, S \to 0\}, S)$ je regulární gramatika a generuje jazyk $L(G_1) = \{0,1\}^+$.

$G_2 = (\{S\}, \{0,1\}, \{S \to 0S1, S \to 01\}, S)$ je bezkontextová a generuje jazyk $L(G_2) = \{0^n 1^n : n \geq 1\}$.

$G_3 = (\{S, A\}, \{0,1\}, \{S \to 0A1, S \to 01, 0A \to 00A1, A \to 01\}, S)$ je kontextová a generuje jazyk $L(G_3) = \{0^n 1^n : n \geq 1\}$.

$G_4 = (\{S\}, \{0,1\}, \{S \to 0S1, S \to 01, 0S1 \to S\}, S)$ je neomezená a generuje jazyk $L(G_4) = \{0^n 1^n : n \geq 1\}$.

Gramatiky G_2, G_3 a G_4 jsou ekvivalentní, protože $L(G_2) = L(G_3) = L(G_4) = \{0^n 1^n : n \geq 1\}$.

Zařazení jazyka

- nalezení gramatiky,
- nalezení automatu,
- Pumping Lemma, Nerodova věta

Pumping lemma pro regulární jazyky

Pumping lemma pro regulární jazyky

Nechť L je regulární jazyk. Pak existuje taková konstanta $p > 1$, že pro $w \in L$ a $|w| \geq p$ může být w zapsáno ve tvaru $w = xyz$, kde $1 \leq |y| < p$ a $xy^i z \in L$ pro všechna $i \geq 0$.

Použití: Důkaz, že nějaký jazyk není regulární.

Příklad

$L = \{a^n b^n : n \geq 0\}$

Zvolíme $w = a^{p+1}b^{p+1}$. Pak $x = a^k, y = a^{p+1-k}b^l, z = b^{p+1-l}, k < p$.

$xz = a^k b^{p+1-l} \notin L$

$xyz = a^k a^{p+1-k} b^l b^{p+1-l} = a^{p+1}b^{p+1} \in L$

$xyyz = a^k a^{p+1-k} b^l a^{p+1-k} b^l b^{p+1-l} \notin L$

Proto L není regulární.

Pumping lemma pro bezkontext. jazyky

Pumping lemma pro bezkontextové jazyky
Nechť L je bezkontextový jazyk. Pak existuje taková konstanta $p > 1$, že pro $w \in L$ a $|w| \geq p$ může být w zapsáno ve tvaru $w = xuyvz$, kde $1 \leq |uyv| < p$ a $xu^iyv^iz \in L$ pro všechna $i \geq 0$.

Použití: Důkaz, že nějaký jazyk není bezkontextový.

Příklad
$L = \{a^n b^n c^n : n \geq 0\}$

Zvolíme $w = a^p b^p c^p$.

Pak ...

$\vdots$

Proto L není bezkontextový.

Gramatiky a operace nad jazyky

Algoritmus Konstrukce gramatiky pro *sjednocení* jazyků.
Vstup: Bezkontextové gramatiky G_1 a G_2 generující jazyky L_1 a L_2.
Výstup: Bezkontextová gramatika G, že $L(G) = L_1 \cup L_2$.
Metoda: $G_1 = (N_1, T, P_1, S_1)$, $G_2 = (N_2, T, P_2, S_2)$
Za předpokladu, že množiny terminálních symbolů jsou stejné a $N_1 \cap N_2 = \emptyset$:
$G = (N_1 \cup N_2 \cup \{S\}, T, P_1 \cup P_2 \cup \{S \to S_1 \mid S_2\}, S)$, kde $S \notin N_1 \cup N_2$.

Gramatiky a operace nad jazyky

Algoritmus Konstrukce gramatiky pro *součin* jazyků.
Vstup: Bezkontextové gramatiky G_1 a G_2 generující jazyky L_1 a L_2.
Výstup: Bezkontextová gramatika G, že $L(G) = L_1.L_2$.
Metoda: $G_1 = (N_1, T, P_1, S_1)$,
$G_2 = (N_2, T, P_2, S_2)$, $N_1 \cap N_2 = \emptyset$. Gramatika G se vytvoří takto:
$G = (N_1 \cup N_2 \cup \{S\}, T, P_1 \cup P_2 \cup \{S \to S_1 S_2\}, S)$, kde $S \notin N_1 \cup N_2$.

Gramatiky a operace nad jazyky

Algoritmus Konstrukce gramatiky pro *iteraci* jazyka.
Vstup: Bezkontextová gramatika G generující jazyk L.
Výstup: Bezkontextová gramatika G', že $L(G') = L^*$.
Metoda: $G = (N, T, P, S)$. Výslednou gramatiku G' zkonstruujeme takto:
$G' = (N \cup \{S'\}, T, P \cup \{S' \to SS', S' \to \varepsilon\}, S')$, kde $S' \notin N$.

Konečné jazyky

Definice

Konečný jazyk $L \subset \Sigma^*$ je konečný právě tehdy, když L je konečná množina. □

Definice

Nejmenší množina jazyků, která obsahuje všechny konečné jazyky a jazyky vzniklé pomocí operací

a) sjednocení,

b) součinu,

c) iterace,

je množina všech regulárních jazyků. □

Derivační strom pro bezkontext. jazyky

Derivační strom je grafickým vyjádřením syntaktické struktury věty.

Derivační strom pro bezkontext. jazyky

Definice

Derivační strom je orientovaný strom.
Jestliže máme dánu gramatiku $G = (N, T, P, S)$, pak derivační strom v této gramatice musí mít tyto vlastnosti:

1. Uzly derivačního stromu jsou ohodnoceny terminálními a neterminálními symboly.
2. Kořen stromu je ohodnocen počátečním symbolem.
3. Jestliže uzel má alespoň jednoho následovníka, je ohodnocen neterminálním symbolem.
4. Jestliže $n_1, n_2, \ldots, n_k$ jsou bezprostřední následovníci uzlu n, který je ohodnocen symbolem A, a tyto uzly jsou zleva doprava ohodnoceny symboly $A_1, A_2, \ldots, A_k$, pak $A \to A_1 A_2 \ldots A_k$ je pravidlo v P.
5. Koncové uzly derivačního stromu tvoří zleva doprava větnou formu nebo větu v gramatice G, která je *výsledkem* derivačního stromu.

Derivační strom pro bezkontext. jazyky

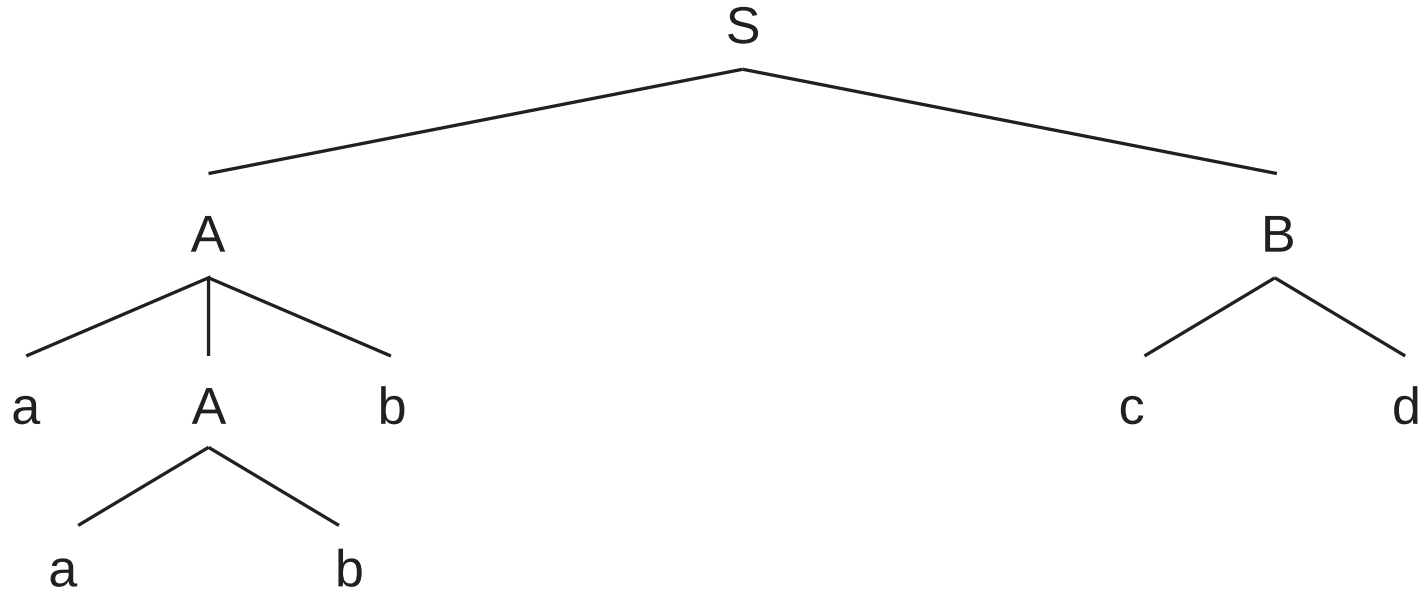
Příklad

$G = (\{S, A, B\}, \{a, b, c, d\}, P, S)$, kde P:

$(1) S \rightarrow AB \quad (2) A \rightarrow aAb \quad (3) A \rightarrow ab$

$(4) B \rightarrow cBd \quad (5) B \rightarrow cd$

$S \Rightarrow AB \Rightarrow aAbB \Rightarrow aabbB \Rightarrow aabbcd.$



$S \Rightarrow AB \Rightarrow Acd \Rightarrow aAbcd \Rightarrow aabbcd$

$S \Rightarrow AB \Rightarrow aAbB \Rightarrow aAbcd \Rightarrow aabbcd$