Příklady pro cvičení 4. z IFJ:
Bezkontextové gramatiky, zásobníkové automaty

Příklad 1.
Vytvořte bezkontextovou gramatiku, která generuje jazyk $L_1 = \{a^n b^n : n \geq 1\}$. Pomocí této gramatiky proveďte derivaci řetězce ***aaabbb***.

Řešení:
Gramatika $G = (N, T, P, S)$, kde:

- $N = \{S\}$,
- $T = \{a, b\}$,
- $P = \{1: S \rightarrow aSb, 2: S \rightarrow ab\}$

generuje tento jazyk. Řetězec ***aaabbb*** potom můžeme vygenerovat pomocí následující posloupností derivací:

$$S \Rightarrow aSb\ [1] \Rightarrow aaSbb\ [1] \Rightarrow aaabbb\ [2]$$

Nebo zkráceně, pokud nás explicitně nezajímá použité pravidlo, můžeme jejich označení vynechat:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$$

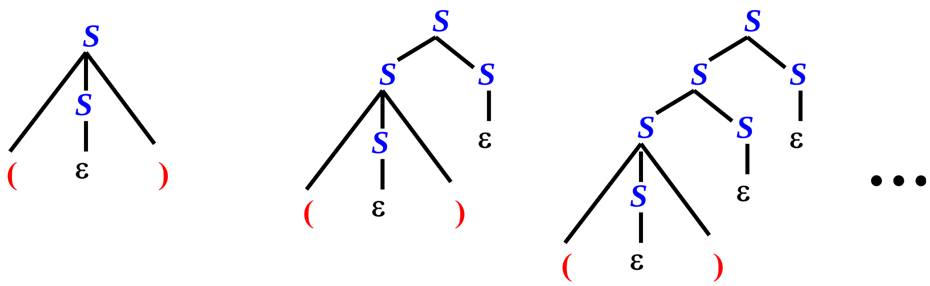
Příklad 2.
Nechť L_2 je jazyk nad abecedou Σ = { „(“ , „)“ } obsahující všechny řetězce, které vzniknou z aritmetických výrazů vypuštěním všech symbolů kromě závorek. Například ( )(( )( )) ∈ L_2, neboť tento řetězec vznikl například z aritmetického výrazu $(a + b).((a + b) – (c + d))$. Je vámi navržená gramatika jednoznačná? Pokud ne, zkuste ji zmodifikovat tak, aby jednoznačná byla.

Řešení:
Gramatika $G = (N, T, P, S)$, kde:

- $N = \{S\}$,
- $T = \{(,)\}$,
- $P = \{1: S \rightarrow SS, 2: S \rightarrow (S), 3: S \rightarrow \varepsilon\}$

generuje tento jazyk.

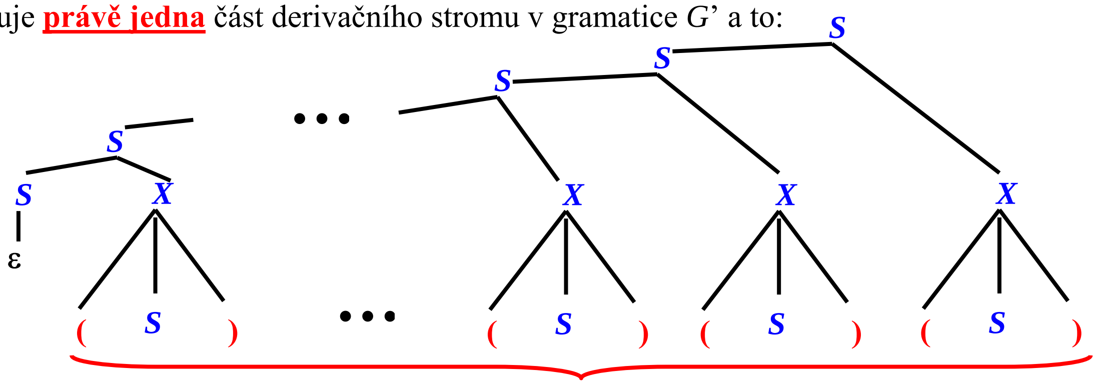
Tato gramatika <u>**není jednoznačná**</u>, protože již například pro řetězec ( ) existuje nekonečně mnoho derivačních stromů (pro nejednoznačnost gramatiky by stačily pouze dva):



Ekvivalentní jednoznačná gramatika k vytvořené gramatice G je například:
$G' = (N, T, P, S)$, kde:

- $N = \{S, X\}$,
- $T = \{(,)\}$,
- $P = \{1: S \rightarrow SX, 2: X \rightarrow (S), 3: S \rightarrow \varepsilon\}$

Uvažujme, že závorková struktura obsahuje na nejvyšší úrovni právě n párů závorek. Potom existuje **právě jedna** část derivačního stromu v gramatice G' a to:



Tyto derivační stromy můžeme do sebe zanořovat a tím budou vznikat libovolné závorkové struktury různých úrovní. Na závěr budou přepsány všechny zbývající nonterminální symboly pomocí pravidla 3: $S \to \varepsilon$.
Je tedy vidět, že pro libovolnou závorkovou strukturu můžeme vytvořit v G' pouze **jeden** derivační strom. Gramatika G' je tedy jednoznačná. Jazyk L_2 **není jazyk s inherentní nejednoznačností**, protože existuje jednoznačná gramatika G', která jej generuje.

Příklad 3.
Vytvořte bezkontextovou gramatiku, která generuje jazyk $L_3 = \{xy: x, y \in \{a, b\}^* \wedge y = \text{reversal}(x)\}$.
Neformálně, jazyk L_3 obsahuje všechny řetězce obsahující pouze symboly a, b, pro které platí, že přečtená druhá polovina řetězce od konce tvoří první polovinu řetězce.

Řešení:
Gramatika $G = (N, T, P, S)$, kde:

- $N = \{S\}$,
- $T = \{a, b\}$,
- $P = \{1: S \to aSa,\ 2: S \to bSb,\ 3: S \to \varepsilon\}$

generuje tento jazyk.

Poznámka: Řetězec, který se čte stejně zepředu i pozpátku, se nazývá palindrom. Jazyk L_3 tedy obsahuje všechny palindromy sudé délky nad abecedou $\Sigma = \{a, b\}$.

Příklad 4.
Vytvořte bezkontextovou gramatiku, která generuje jazyk $L_4 = \{x: x \in \{a, b\}^* \wedge \#_a x > \#_b x\}$.

Poznámka: $\#_a x$ obecně znamená počet výskytů symbolů a v řetězci x, tedy podmínka $\#_a x > \#_b x$ popisuje skutečnost, že počet výskytů symbolů a v řetězci x je větší než počet symbolů b.

Řešení:
Gramatika $G = (N, T, P, S)$, kde:

- $N = \{S\}$,
- $T = \{a, b\}$,
- $P = \{1: S \to a,\ 2: S \to aS,\ 3: S \to bSS\ \ 4: S \to SbS,\ 5: S \to SSb\}$

generuje tento jazyk.

Pravidla 3, 4 a 5 vygenerují libovolnou větnou formu, která obsahuje pouze nonterminální symboly ***S*** a terminální symboly ***b***, přičemž počet nonterminálních symbolů ***S*** je o jedničku větší než počet terminálních symbolů ***b***. Potom pravidla 1 a 2 dokončí derivaci řetězce tak, že každý nonterminál ***S*** je přepsán sekvencí terminálních symbolů ***a***, přičemž tato sekvence obsahuje aspoň jeden terminální symbol ***a***. Celkem můžeme tedy vygenerovat všechny řetězce, které obsahují více symbolů ***a*** než symbolů ***b***.

Příklad 5.
Sestrojte zásobníkový automat, který přijímá jazyk $L_1 = \{a^n b^n : n \geq 1\}$ z příkladu 1. s vyprázdněním zásobníku a s přechodem do koncového stavu.

Řešení:
Základní myšlenka činnosti tohoto automatu:

1) Zásobníkový automat bude číst symboly ***a*** ze vstupu a všechny je ukládat na zásobník. Tuto skutečnost zařídíme pravidly:
   Ssa → ***Sas***, ***asa*** → ***aas***
   (**Poznámka:** první pravidlo je použito při čtení prvního symbolu ***a***, protože na zásobníku je uložen pouze startující zásobníkový symbol ***S***. Při dalším čtení budou již na zásobníku nějaké symboly ***a*** a použito bude tedy druhé pravidlo.)

2) Jakmile automat přečte na vstupu první symbol ***b***, vytáhne jeden symbol ***a*** ze zásobníku a přejde do jiného stavu ***q***. Přechodem do jiného stavu ***q*** zabezpečíme, že automat již nikdy nebude číst ze vstupu symboly ***a***. Při dalším čtení symbolu ***b*** opět vytáhneme jeden symbol ***a*** ze zásobníku ale již setrváme ve stavu ***q***. Tuto skutečnost zařídíme pravidly:
   asb → ***q***, ***aqb*** → ***q***

3) Jakmile je celý řetězec přečtený, zkontrolujeme stav zásobníku a to následovně. Pokud řetězec patří do daného jazyka L_1, musí opět zásobník obsahovat pouze startující zásobníkový symbol. (Zásobník totiž obsahoval přesně tolik symbolů ***a***, kolik jich bylo v první části řetězce. Při čtení symbolů ***b*** druhé části řetězce se vždy jeden symbol ***a*** ze zásobníku odstranil, tedy pokud jsou tyto počty stejné, nesmí být na něm žádný symbol ***a***) Tuto kontrolu můžeme provést následujícím pravidlem, které dostane automat do koncového stavu ***f*** a vyprázdní mu zásobník:
   Sq → ***f***

Formální popis tohoto zásobníkového automatu:
$M = (Q, \Sigma, \Gamma, R, s, S, F)$, kde

- $Q = \{s, q, f\}$,
- $\Sigma = \{a, b\}$,
- $\Gamma = \{S, a, b\}$,
- $R = \{Ssa \to Sas,\ asa \to aas,\ asb \to q,\ aqb \to q,\ Sq \to f\}$,
- $F = \{f\}$

Příklad 6.
Sestrojte zásobníkový automat, který přijímá jazyk L_2 z příkladu 2. s vyprázdněním zásobníku a přechodem do koncového stavu.

Řešení:

Základní myšlenka činnosti tohoto automatu:

1) Zásobníkový automat bude číst symboly **(** ze vstupu a všechny je ukládat na zásobník. Tuto skutečnost zařídíme pravidly:
   $S s(\rightarrow S(s,\ (s(\rightarrow ((s$
2) Pokud zásobníkový automat přečte symbol **)**, musí mít na zásobníku symbol **(**, který tvoří s tímto přečteným symbolem **)** závorkový pár. Tato závorka tedy bude odstraněna ze zásobníku pravidlem:
   $(s) \rightarrow s$

3) Jakmile je celý řetězec přečtený, zkontrolujeme, zda zásobník obsahuje pouze startující zásobníkový symbol. Pouze v tomto případě byly závorky správně popárovány. Přidáme tedy pravidlo:
   $Ss \rightarrow f$

Formální popis tohoto zásobníkového automatu:

$M = (Q, \Sigma, \Gamma, R, s, S, F)$, kde

- $Q = \{s, f\}$,
- $\Sigma = \{(,)\}$,
- $\Gamma = \{S, (,)\}$,
- $R = \{Ss(\rightarrow S(s,\ (s(\rightarrow ((s,\ (s) \rightarrow s,\ Ss \rightarrow f\}$,
- $F = \{f\}$

Příklad 7.

Sestrojte rozšířený zásobníkový automat, který přijímá jazyk L_3 z příkladu 3. s vyprázdněním zásobníku a přechodem do koncového stavu.

Řešení:

Základní myšlenka činnosti tohoto automatu:

Automat bude pracovat nedeterministicky a to následujícím způsobem:

1) Rozšířený zásobníkový automat bude číst symboly ***a***, ***b*** ze vstupu a všechny je ukládat na zásobník. Tuto skutečnost zařídíme pravidly:
   $sa \rightarrow as,\ sb \rightarrow bs$
   (**Poznámka:** Všimněte si, že u rozšířeného zásobníkového automatu nemusíme číst žádný symbol ze zásobníku)

2) Až bude mít automat přečtenu přesně polovinu řetězce, vloží na zásobník symbol ***C***. POZOR, všimněte si, že rozšířený zásobníkový automat nemá žádný prostředek pro najití středu řetězce, neví, jak je ještě nepřečtená část dlouhá, proto je tento krok proveden !!!**NEDETERMINISTIKY!!!** pomocí pravidla:
   $s \rightarrow Cs$

3) Pokud má být druhá polovina řetězce stejná jako byla první reverzovaná první polovina, stačí postupně porovnávat následující čtené symboly se symboly na zásobníku a ty odstraňovat. Pozor, na vrcholu zásobníku je uložen symbol ***C***, který nám říká, že jsme za polovinou čteného řetězce. První skutečný symbol je na zásobníku až pod tímto symbolem. Porovnávání můžeme tedy provést následujícími pravidly:
   $aCsa \rightarrow Cs,\ bCsb \rightarrow Cs$

4) Jakmile je celý řetězec přečtený, zkontrolujeme, zda zásobník obsahuje pod symbolem ***C*** pouze startující zásobníkový symbol pomocí pravidla:
   $SCs \to f$

Formální popis tohoto rozšířeného zásobníkového automatu:

$M = (Q, \Sigma, \Gamma, R, s, S, F)$, kde

- $Q = \{s, f\}$,
- $\Sigma = \{a, b\}$,
- $\Gamma = \{S, a, b\}$,
- $R = \{sa \to as,\ sb \to bs,\ s \to Cs,\ aCsa \to Cs,\ bCsb \to Cs,\ SCs \to f\}$,
- $F = \{f\}$

Příklad 8.

Sestrojte zásobníkový automat, který přijímá jazyk L_4 z příkladu 4. s vyprázdněním zásobníku a přechodem do koncového stavu.

Řešení:

Základní myšlenka činnosti tohoto automatu:

Automat bude v každém kroku obsahovat na zásobníku kromě startujícího symbolu pouze symboly ***a*** nebo pouze symboly ***b***, ale nikdy nebude obsahovat oba druhy symbolů současně! A to následujícím způsobem: Nechť bylo do teď přečteno ze vstupu právě *m* symbolů ***a*** a *n* symbolů ***b***. Potom, pokud je $m > n$, bude zásobník obsahovat právě $m - n$ symbolů ***a***, pokud je $m < n$, bude zásobník obsahovat právě $n - m$ symbolů ***b*** a konečně pro případ $m = n$ nebude zásobník obsahovat žádné symboly ***a***, ***b***. Stručně a neformálně řečeno, zásobník obsahuje ty symboly, kterých bylo doposud přečteno více a tolik, o kolik jich bylo přečteno více než druhých. Toho dosáhneme následujícím způsobem:

1) Pokud je přečten symbol ze vstupu symbol ***a*** nebo ***b*** a zásobník je prázdný, vložíme tento symbol na zásobník pomocí pravidel:
   $Ssa \to Sas$, $Ssb \to Sbs$

2) Pokud je přečten ze vstupu symbol ***a*** a na vrcholu zásobníků již je symbol ***a***, přidáme symbol ***a*** na zásobník pomocí pravidla:
   $asa \to aas$

3) Pokud je přečten ze vstupu symbol ***b*** a na vrcholu zásobníků již je symbol ***b***, přidáme symbol ***b*** na zásobník pomocí pravidla:
   $bsb \to bbs$

4) Pokud je přečten ze vstupu symbol ***a*** a na vrcholu zásobníků je symbol ***b***, ubereme jeden symbol ***b*** ze zásobníku pro vyrovnání počtů pomocí pravidla:
   $bsa \to s$

5) Pokud je přečten ze vstupu symbol ***b*** a na vrcholu zásobníků je symbol ***a***, ubereme jeden symbol ***a*** ze zásobníku pro vyrovnání počtů pomocí pravidla:
   $asb \to s$

6) Jakmile je celý řetězec přečtený, zkontrolujeme, zda zásobník obsahuje „nějaké“ symboly a. Jedině tehdy bylo totiž přečteno více symbolů a než b. To zkontrolujeme následujícím pravidlem, pomocí kterého přejde automat do koncového stavu f:
   $as \rightarrow f$

7) Pokud je automat v koncovém stavu, necháme jej vyprázdnit zásobník užitím následujících dvou pravidel:
   $af \rightarrow f$, $Sf \rightarrow f$,

Formální popis tohoto zásobníkového automatu:

$M = (Q, \Sigma, \Gamma, R, s, S, F)$, kde

- $Q = \{s, f\}$,
- $\Sigma = \{a, b\}$,
- $\Gamma = \{S, a, b\}$,
- $R = \{Ssa \rightarrow Sas,\ Ssb \rightarrow Sbs,\ asa \rightarrow aas,\ bsb \rightarrow bbs,\ bsa \rightarrow s,\ asb \rightarrow s,\ as \rightarrow f,$
  $af \rightarrow f, Sf \rightarrow f\}$,
- $F = \{f\}$

(**Poznámka:** Pokud by stačilo, aby automat přijímal jazyk pouze přechodem do koncového stavu, mohli bychom poslední dvě pravidla odstranit)

Příklad 9.

Sestrojte gramatiku, která generuje jazyk $L_5 = \{a^m b^n c^n\colon m, n \geq 1\}$. Ke gramatice vytvořte:

a) Zásobníkový automat přijímající jazyk L_5 s vyprázdněním zásobníku simulující syntaktickou analýzu shora dolů.

b) Rozšířený zásobníkový automat přijímající jazyk L_5 s přechodem do koncového stavu simulující syntaktickou analýzu zdola nahoru.

Demonstrujte přijetí řetězce *abbcc* oběma automaty.

Řešení:

a) Gramatika $G = (N, T, P, S)$, kde:

- $N = \{S, A\}$,
- $T = \{a, b, c\}$,
- $P = \{1\colon S \rightarrow aS,\ 2\colon S \rightarrow aA,\ 3\colon A \rightarrow bAc,\ 4\colon A \rightarrow bc\ \}$

generuje tento jazyk.

b) Výsledný zásobníkový automat simulující syntaktickou analýzu shora dolů bude vypadat následovně:

1. Vytvoříme **porovnávací pravidla** následujícím způsobem: Pro každý terminální symbol t gramatiky G vytvoříme pravidlo tvaru $tst \rightarrow s$. Pro danou gramatiku G tedy vytvoříme pravidla:
   $asa \rightarrow s$, $bsb \rightarrow s$, $csc \rightarrow s$

2. Vytvoříme **expanzivní pravidla** následujícím způsobem: Pro každé pravidlo gramatiky tvaru $A \rightarrow x$ vytvoříme odpovídající pravidlo $As \rightarrow ys$ pro zásobníkový automat, přičemž y je reverzovaný řetězec x, formálně: $y = reversal(x)$. Pro danou gramatiku G tedy vytvoříme pravidla:
   $Ss \rightarrow Sas$, $Ss \rightarrow Aas$, $As \rightarrow cAbs$, $As \rightarrow cbs$

Formální popis tohoto zásobníkového automatu:
$M = (Q, \Sigma, \Gamma, R, s, S, F)$, kde

- $Q = \{s\}$,
- $\Sigma = T = \{a, b, c\}$,
- $\Gamma = N \cup T = \{S, A, a, b\}$,
- $R = \{asa \to s, bsb \to s, csc \to s, Ss \to Sas, Ss \to Aas, As \to cAbs, As \to cbs\}$
- $F = \varnothing$

Poznámka: Všimněte si, že množina koncových stavů je prázdná, protože automat přijímá jazyk s vyprázdněním zásobníku a tedy koncové stavy tu nemají žádný význam

Simulace přijetí řetězce *abbcc* tímto zásobníkovým automatem:
$Ssabbcc \vdash Aasabbcc \vdash Asbbcc \vdash cAbsbbcc \vdash cAsbcc \vdash ccbsbcc \vdash ccscc \vdash csc \vdash s$

b) Výsledný rozšířený zásobníkový automat simulující syntaktickou analýzu zdola nahoru bude vypadat následovně:

1. Vytvoříme **shiftovací pravidla** následujícím způsobem: Pro každý terminální symbol t gramatiky G vytvoříme pravidlo tvaru $st \to ts$. Pro danou gramatiku G tedy vytvoříme pravidla:
   $sa \to as$, $sb \to bs$, $sc \to cs$

2. Vytvoříme **redukční pravidla** následujícím způsobem: Pro každé pravidlo gramatiky tvaru $A \to x$ vytvoříme odpovídající pravidlo $xs \to As$ pro rozšířený zásobníkový automat. Pro danou gramatiku G tedy vytvoříme pravidla:
   $aSs \to Ss$, $aAs \to Ss$, $bAcs \to As$, $bcs \to As$

3. Vytvoříme speciální pravidlo, které uvede automat do koncového stavu:
   $\#Ss \to f$

Formální popis tohoto rozšířeného zásobníkového automatu:
$M = (Q, \Sigma, \Gamma, R, s, S, F)$, kde

- $Q = \{s, f\}$,
- $\Sigma = T = \{a, b, c\}$,
- $\Gamma = N \cup T \cup \{\#\} = \{\#, S, A, a, b\}$,
- $R = \{sa \to as, sb \to bs, sc \to cs, aSs \to Ss, aAs \to Ss, bAcs \to As, bcs \to As, \#Ss \to f\}$
- $F = \{f\}$

Simulace přijetí řetězce *abbcc* tímto rozšířeným zásobníkovým automatem:
$\#sabbcc \vdash \#asbbcc \vdash \#absbcc \vdash \#abbscc \vdash \#abbcsc \vdash \#abAsc \vdash \#abAcs \vdash \#aAs \vdash \#Ss \vdash f$