Formální jazyky II

Bezkontextové jazyky a jejich reprezentace. Varianty zásobníkových automatů (metody akceptování, determinismus a nedeterminismus, rozšířené zásobníkové automaty). Nedeterministická syntaktická analýza. Uzávěrové vlastnosti bezkontextových jazyků.

IB102/IB005

Uvod

Abeceda a jazyk

Abecedou rozumieme lubovolnu konecnu mnozinu Σ. Jej prvky nazyvame znaky (pismena alebo symboly).
Priklad: {a,b}, {0,1,…,9}, $\varnothing$

Slovo (retazec) nad abecedou Σ je lubovolna konecna postupnost znakov z Σ. Dlzka slova v je pocet znakov v slove (znacime #(v)), pocet vyskytu znaku a v slove znacime $\#_a(v)$. Specialny pripad je prazdne slovo s nulovou delkou, znacime ho ε.
Priklad: pre Σ = {a,b} je slovo aabb, aa, b. Pre v = aabb, #(v) = 4 a $\#_a(v)$ = 2

Mnozina vsetkych slov nad Σ znacime Σ^*, mnozinu vsetkych neprazdnych slov znacime Σ^+.
Priklad: $\{a\}^* = \{\varepsilon,a,aa,aaa,aaaa,...\}$, $\{a\}^+ = \{a,aa,aaa,aaaa,...\}$
Specialne: $\varnothing^* = \{\varepsilon\}$ a $\varnothing^+ = \varnothing$.

Jazyk nad abecedou Σ je libovolna mnozina slov nad Σ (i.e. podmnoziny Σ^*). Mozu byt konecne aj nekonecne.
Priklad: $\Sigma = \{0, 1\}$, potom L nad Σ je {10, 1, 011101}. L nad {a,b} definovany ako $\{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$ je zas nekonecny.
Specialne: $\varnothing$ je L nad libovolnou Σ.

Operace nad jazykmi

Kedze jazyku su len mnoziny, mozeme aplikovat mnozinove operacie: zjednotenie, prienik a rozdiel.
Priklad: Ak L je nad Σ, K je nad Δ , tak $L \cup K$ je nad $\Sigma \cup \Delta$.

Dalej definujeme (pre jazyky L je nad Σ, K je nad Δ)
- Zretazenie: $K.L = \{uv \mid u \in K, v \in L\}$ nad $\Sigma \cup \Delta$
- i-ta mocnina jazyka L: $L^0 = \{\varepsilon\}$ a $L^{i+1} = L.L^i$
- Iterace jazyka L: $L^* = \bigcup_{i=0}^{\infty} L^i$.
- Doplnok: co–L = $\Sigma^* \setminus L$.
- Substituce, homomorfizmus (prip. inverzni), zrkadlovy obraz…

Specialne: $\varnothing.L = L.\varnothing = \varnothing$ a $\{\varepsilon\}.L = L.\{\varepsilon\} = L$, $\varnothing^0 = \{\varepsilon\}$, $\varnothing^i = \varnothing$ $(i \in N)$, $\{\varepsilon\}^j = \{\varepsilon\}$ $(j \in N_0)$.

Definice 1.2. *Gramatika* $\mathcal{G}$ je čtveřice (N, Σ, P, S), kde

- N je neprázdná konečná množina *neterminálních symbolů* (stručněji: *neterminálů*).
- Σ je konečná množina *terminálních symbolů (terminálů)* taková, že $N \cap \Sigma = \emptyset$. Sjednocením N a Σ obdržíme množinu *všech symbolů* gramatiky, kterou obvykle označujeme symbolem V.
- $P \subseteq V^*NV^* \times V^*$ je konečná množina *pravidel*. Pravidlo (α, β) obvykle zapisujeme ve tvaru $\alpha \to \beta$ (a čteme jako "α přepiš na β").
- $S \in N$ je speciální *počáteční* neterminál (nazývaný také *kořen gramatiky*).

Poziadavky na pravidla (α,β) : α musi obsahovat aspon jeden neterminal.
β moze byt aj prazna (ε).
Vetna forma gramatiky G: prvky mnoziny (N a Σ) * , ktore ide odvodit z pociatocneho neterminalu za pomoci pravidiel gramatiky.
Formalne: α ∈ (N a Σ) * je vetna forma $\Longleftrightarrow$ S $\Rightarrow^*$ α.
Vetna forma bez neterminalu je veta. Mnozina vsetkcyh viet gramatiky je jazyk generovany gramatikou: L(G) = {w ∈ Σ^* | S $\Rightarrow^*$ w}

Bezkontextové jazyky a jejich reprezentace (derivacne stromy, Chomskeho normalni forma, Greibachove normalni forma).

CFG = context free grammar

Derivacni stromy

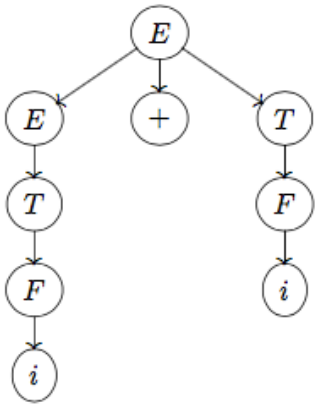
Nech G je CFG. Strom T je derivacny strom G ak:

- kazdy uzol ma navesti - symbol z mnoziny $N \cup \Sigma \cup \{\varepsilon\}$
- koren ma navesti S
- ak ma vnutorny uzol navesti A, tak $A \in N$
- ak ma uzol navesti A a ma k synov $(X_1,\ldots,X_k)$, tak existuje pravidlo $A \rightarrow X_1 \ldots X_k \in P$
- ak ma uzol navesti ε, tak je list, a jeho otec nema inych potomkov

Příklad 3.2. *Nechť $\mathcal{G}_0$ je gramatika s pravidly*

$$\begin{array}{lcll} E & \rightarrow & E+T & \mid\ T \\ T & \rightarrow & T*F & \mid\ F \\ F & \rightarrow & (E) & \mid\ i \end{array}$$

pak derivační strom



Pozor: jeden derivacny strom moze reprezentovat vecsie mnoztvo derivacii (ktore su ale ekvivalentne), kedze strom nedefinuje poradie vyhodnocovania (kedy sa ktory neterminal prepise).

Definice 3.5. CFG $\mathcal{G}$ se nazývá *víceznačná (nejednoznačná)* právě když existuje $w \in L(\mathcal{G})$ mající alespoň dva různé derivační stromy. V opačném případě říkáme, že $\mathcal{G}$ je jednoznačná. Jazyk L se nazývá *vnitřně (inherentně) víceznačný* právě když každá gramatika, která jej generuje, je víceznačná.

Priklad:

Příklad 3.6. *Gramatika $\mathcal{G}_1$ s pravidly $E \to E + E \mid E * E \mid (E) \mid i$, která je ekvivaltní s gramatikou $\mathcal{G}_0$ z příkladu 3.2, je víceznačná; například proto, že věta $i + i + i$ má dvě různé levé derivace a jim odpovídající dva různé derivační stromy:*

Redukovana gramatika:
Gramatika G je redukovana, ak neobsahuje ziadne nepouzitelne symboly. To su tie, z ktorych nevieme nic vyderivovat (neexistuje derivace $S \Rightarrow^* wXy \Rightarrow^* wxy$)

Necyklicka gramatika:
Gramatika G je necyklicka, ak neobsahuje ziadne odvodenie tvaru $A \Rightarrow^+ A$.

Gramatika bez jednoduchych pravidiel:
Jednoduche pravidla su pravidla v tvare $A \to B$, $(A,B \in N)$,

Vlastni gramatika:
Gramatika G je vlastni, ak je bez nepouzitelnych symbolov, bez ε-pravidel a je necyklicka.

Chomskeho normalni forma(CNF):

Definice 3.19. Řekneme, že CFG $\mathcal{G} = (N, \Sigma, P, S)$ je v *Chomského normální formě* (CNF) $\overset{def}{\iff}$ $\mathcal{G}$ je bez ε-pravidel (viz def. 3.13) a každé pravidlo z P (s eventuelní výjimkou $S \to \varepsilon$) má jeden z těchto tvarů:

1. $A \to BC$, kde $B, C \in N$ nebo
2. $A \to a$, kde $a \in \Sigma$.

Na prevod gramatiky do Chomskeho normalni formy existuje algoritmus. Podmienkou je, ze gramatika neobsahuje jednoduche pravidla.

Priklad: $S \rightarrow ASA \mid aB, A \rightarrow B \mid S, B \rightarrow b \mid \varepsilon$

1. koren sa moze redukovat na ε:

$S_0 \rightarrow S$, $S \rightarrow ASA \mid aB$, $A \rightarrow B \mid S$, $B \rightarrow b \mid \varepsilon$

2. odstranenie ε pravidiel:

$S_0 \rightarrow S$, $S \rightarrow ASA \mid aB \mid a \mid AS \mid SA \mid S$, $A \rightarrow B \mid S$, $B \rightarrow b$

3. odstranenie jednoduchych pravidiel ($S \rightarrow S$, $S_0 \rightarrow S$, $A \rightarrow B$, $A \rightarrow S$):

$S_0 \rightarrow ASA \mid aB \mid a \mid AS \mid SA$

$S \rightarrow ASA \mid aB \mid a \mid AS \mid SA$

$A \rightarrow b \mid ASA \mid aB \mid a \mid AS \mid SA$

$B \rightarrow b$

4. odstranenie pravidiel s pravou stranou dlhsou ako 2 ($S_0 \rightarrow ASA$, $S \rightarrow ASA$, $A \rightarrow ASA$):

$S_0 \rightarrow AX \mid aB \mid a \mid AS \mid SA$

$S \rightarrow AX \mid aB \mid a \mid AS \mid SA$

$A \rightarrow b \mid AX \mid aB \mid a \mid AS \mid SA$

$B \rightarrow b$

$X \rightarrow SA$

5. odstranenie pravidiel v tvare aB ($S_0 \rightarrow aB$, $S \rightarrow aB$, $A \rightarrow aB$):

$S_0 \rightarrow AX \mid YB \mid a \mid AS \mid SA$

$S \rightarrow AX \mid YB \mid a \mid AS \mid SA$

$A \rightarrow b\ A \rightarrow b \mid AX \mid YB \mid a \mid AS \mid SA$

$B \rightarrow b$

$X \rightarrow SA$

$Y \rightarrow a$

Na gramatiku v CNF mozme aplikovat pumping lemmu pre CFG:

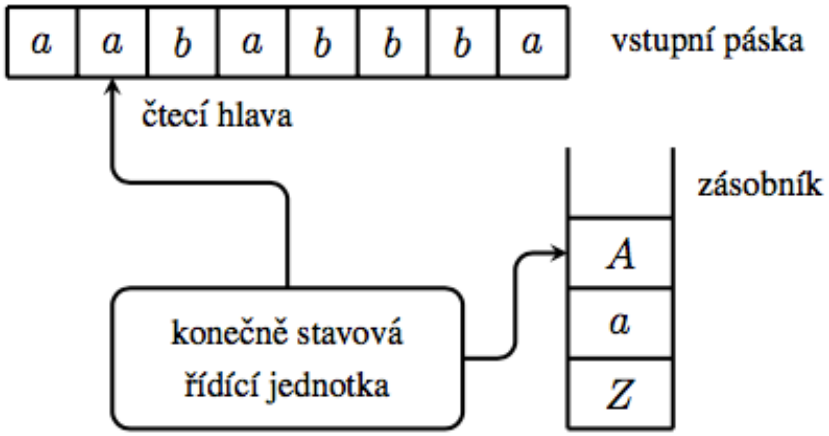
Věta 3.24. ***(Lemma o vkládání, pumping lemma pro CFL)*** *Nechť L je CFL. Pak existují přirozená čísla p, q (závisející na L) taková, že každé slovo $z \in L$, $|z| > p$ lze psát ve tvaru $z = uvwxy$, kde*

- *alespoň jedno ze slov v, x je neprázdné (tj. $vx \neq \varepsilon$),*
- *$|vwx| \leq q$ a*
- *$uv^i wx^i y \in L$ pro všechna $i \geq 0$.*

Greibachove normalni forma:

Gramatika je v Greibachove normalni forme ak kazda prava strana pravidla zacina terminalnim symbolem (za ktorym mozu pripadne nasledovat neterminaly)

Varianty zásobníkových automatů (metody akceptování, determinismus a nedeterminismus, rozšířené zásobníkové automaty).



Obrázek 3.2: Zásobníkový automat

Definice 3.36. *Nedeterministický zásobníkový automat* (PDA) je sedmice

$$\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F), \text{ kde}$$

- Q je konečná množina, jejíž prvky nazýváme *stavy*,
- Σ je konečná množina, tzv. *vstupní abeceda*,
- Γ je konečná množina, tzv. *zásobníková abeceda*,
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma) \to \mathcal{P}_{Fin}(Q \times \Gamma^*)$ je (parciální) *přechodová funkce*[2] ,
- $q_0 \in Q$ je *počáteční stav*,
- $Z_0 \in \Gamma$ je *počáteční symbol v zásobníku*,
- $F \subseteq Q$ je množina *koncových stavů*.

Krok vypoctu:

$$(p, aw, Z\alpha) \underset{\mathcal{M}}{\vdash} (q, w, \gamma\alpha) \overset{def}{\iff} \exists (q, \gamma) \in \delta(p, a, Z) \text{ pro } a \in \Sigma \cup \{\varepsilon\}$$

kde (q, γ) ∈ Q × Γ* je vnnutorna konfiguracia autoamtu, ktora hovori o aktualnom stave, a co je na zasobniku (celom, nie len vrchole!), a (p, w, α) z Q × Σ* × Γ* je konfiguracia (w predstavuje doteraz neprecitane slovo).

Metody akcaptovania (->* je tranzitivny uzaver nad krokom vypoctu):

1. konecnym stavom:
   $L(M) = \{ w \in \Sigma^* \mid (q_0,w,Z_0) \to^* (q_f, \varepsilon, \alpha), q_f \in F, \alpha \in \Gamma^* \}$
2. prazdnym zasobnikom
   $L_e(M) = \{ w \in \Sigma^* \mid (q_0,w,Z_0) \to^* (q, \varepsilon, \varepsilon), q \in Q \}$
3. koncovym stavom a prazdnym zasobnikom
   prvky z $F \times \{\varepsilon\}$
4. vrcholovymi symbolmi na zasobniku
   prvky $Q \times \Gamma'\Gamma^*$ pro ne˘jakou $\Gamma' \subset \Gamma$

Definice 3.44. *Rozšířeným* PDA nazveme $\mathcal{R} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde všechny symboly mají tentýž význam jako v definici PDA s výjimkou δ, která je zobrazením z *konečné podmnožiny* množiny $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^*$ do konečných podmnožin množiny $Q \times \Gamma^*$. Pojmy konfigurace, kroku výpočtu, výpočtu a akceptovaného jazyka (koncovým stavem, prázdným zásobníkem) zůstávají rovněž beze změny.

Deterministicke zasobnikove automaty DPDA

Definice 3.72. Řekneme, že PDA $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je *deterministický* (DPDA), jestliže jsou splněny tyto podmínky:

1. pro všechna $q \in Q$ a $Z \in \Gamma$ platí: kdykoliv $\delta(q, \varepsilon, Z) \neq \emptyset$, pak $\delta(q, a, Z) = \emptyset$ pro všechna $a \in \Sigma$;
2. pro žádné $q \in Q, Z \in \Gamma$ a $a \in \Sigma \cup \{\varepsilon\}$ neobsahuje $\delta(q, a, Z)$ více než jeden prvek.

Řekneme, že L je *deterministický bezkontextový jazyk* (DCFL, stručněji též deterministický jazyk), právě když existuje DPDA $\mathcal{M}$ takový, že $L = L(\mathcal{M})$.

Podmienka 1 vylucuje, ze by sme sa mohli rozhodnut bez citania vstupenho sybbolu (i.e. ε krok len na zaklade zasobnika), a normalnom kroku na zaklade zasobnika + vstupu.
Podmienka 2 vylucuje viacero moznozsti pre ε krok alebo vstupni symbol.

Nedeterministická syntaktická analýza.

Vyuzitie: na ukazanie ze trieda jazykov rozpoznavana zasobnikovymi automatmi tvori prave triedu bezkontextovych jazykov.

1. zhora dolu - simulacia lavej derivacie v gramatike

Příklad 3.49. *Mějme gramatiku* $\mathcal{G}_0 = (\{E, T, F\}, \{+, *, (,), i\}, P, E)$ *s pravidly P danými takto:*

$$E \to E{+}T \mid T$$
$$T \to T{*}F \mid F$$
$$F \to (E) \mid i$$

krok		odpovídající pravidlo z $\mathcal{G}_0$
$(i+i*i,\ E)$	$\vdash^{\varepsilon} (i+i*i,\ E+T)$	$E \to E+T$
	$\vdash^{\varepsilon} (i+i*i,\ T+T)$	$E \to T$
	$\vdash^{\varepsilon} (i+i*i,\ F+T)$	$T \to F$
	$\vdash^{\varepsilon} (i+i*i,\ i+T)$	$F \to i$
	$\vdash^{i} (\ \ +i*i,\ \ +T)$	
	$\vdash^{+} (\ \ i*i,\ \ T)$	
	$\vdash^{\varepsilon} (\ \ i*i,\ T*F)$	$T \to T*F$
	$\vdash^{\varepsilon} (\ \ i*i,\ F*F)$	$T \to F$
	$\vdash^{\varepsilon} (\ \ i*i,\ i*F)$	$F \to i$
	$\vdash^{i} (\ \ *i,\ \ *F)$	
	$\vdash^{*} (\ \ i,\ \ F)$	
	$\vdash^{\varepsilon} (\ \ i,\ \ i)$	$F \to i$
	$\vdash^{i} (\ \ ,\ \ )$	

Vysledok: Ke každé CFG G lze sestrojit PDA M takový, že $L(G) = L_e(M)$.

poznamky: Plati, ze tato analyza je nedeterministicka: a teda casova zlozitost je exponencialna. Vzhladom na siroke vyuzitie tejto analyzy (napr. syntakticke analyzatory, kompilatory), bezne sa k tymto gramatikam naratavaju informacie (napr. tabulk), ktore nasledne zabezpecuju syntakticku analyzu v linearnom case.
Tento postup vsak nie je mozny pre vsetky typy CFG, ale len pre niektore. Rozlysuju sa LR, SLR (prip. LALR), LL, SLL gramatiky, na zaklade “narocnosti” predpocitavania tabulky.

2. zdola nahoru

Příklad 3.49. *Mějme gramatiku* $\mathcal{G}_0 = (\{E, T, F\}, \{+, *, (,), i\}, P, E)$ *s pravidly P danými takto:*

$$E \to E{+}T \mid T$$
$$T \to T{*}F \mid F$$
$$F \to (E) \mid i$$

krok výpočtu		odpovídající pravidlo z $\mathcal{G}_0$ pro následující krok
$(q,\ \bot,\ i+i*i)$	$\vdash^{i} (q,\ \bot i,\ +i*i)$	$F \to i$
	$\vdash^{\varepsilon} (q,\ \bot F,\ +i*i)$	$T \to F$
	$\vdash^{\varepsilon} (q,\ \bot T,\ +i*i)$	$E \to T$
	$\vdash^{\varepsilon} (q,\ \bot E,\ +i*i)$	
	$\vdash^{+} (q,\ \bot E+,\ i*i)$	
	$\vdash^{i} (q,\ \bot E+i,\ *i)$	$F \to i$
	$\vdash^{\varepsilon} (q,\ \bot E+F,\ *i)$	$T \to F$
	$\vdash^{\varepsilon} (q,\ \bot E+T,\ *i)$	
	$\vdash^{*} (q,\ \bot E+T*,\ i)$	
	$\vdash^{i} (q,\ \bot E+T*i,\ \varepsilon)$	$F \to i$
	$\vdash^{\varepsilon} (q,\ \bot E+T*F,\ \varepsilon)$	$T \to T*F$
	$\vdash^{\varepsilon} (q,\ \bot E+T,\ \varepsilon)$	$E \to E+T$
	$\vdash^{\varepsilon} (q,\ \bot E,\ \varepsilon)$	
	$\vdash^{\varepsilon} (r,\ \varepsilon\ \ \varepsilon)$	

vysledok: Nechť G je libovolná CFG, pak lze zkonstruovat rozšířený PDA R takový, že L(G) = L(R).

Uzávěrové vlastnosti bezkontextových jazyků.

1. Trieda bezkontextových jazykov je uzavrena na zjednotenie

$L_1 \cup L_2$: Majme G_1, G_2, plati $L_1 \cup L_2 = G_1 \cup G_2$,ktory je definovany tak, ze gramatike pridame stav S a pravidla $S \rightarrow S_1 \mid S_2$, kde S_1 je inicialny stav 1. gramatiky a S_2 je inicialny stav 2. gramatiky. Gramatika sa tak na zaciatku rozhodne.

2. Trieda bezkontextových jazykov nie je uzavrena na komplement a prienik.

prienik: Majme jazyky $L_1 = \{a^n b^n c^m \mid m,n \geq 1\}$ a $L_2 = \{a^m b^n c^m \mid m,n \geq 1\}$. Oba tyto jazyky jsou CFL , ale ich prienik $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 1\}$ uz nie je CFL.
komplement: $L_1 \cap L_2 = \neg(\neg L_1 \cup \neg L_2)$,

3. Trieda bezkontextových jazykov je uzavrena na zretazenie.

Podobne ako zjednotenie, ale $S \rightarrow S_1 S_2$

4. Trieda bezkontextových jazykov je uzavrena na iteraciu.

Ako zjednotenie, ale $S \rightarrow SS_1 \mid \varepsilon$

5. Trieda bezkontextových jazykov je uzavrena na prienik s regularnym jazykom.

Dokaz je pomocou konstrukcie PDA. Stavy su opet kartezksky sucin povodnych dvoch, a ostane nam potreba mat len jeden zasobnik.

6. Trieda regularnych jazykov je uzavrena na substituciu, homomorfizmus… (uz prilis specificke, no time for that)

7. DCFL je uzavreny na prienik s regularnym jazykom.

Dokaz je pomocou konstrukcie PDA. Stavy su opet kartezksky sucin povodnych dvoch, a ostane nam potreba mat len jeden zasobnik. Zaroven, pre kazdy regularny jazyk existuje deterministicky konecny automat.

8. DCFL nie je uzavreny na prienik.

Idea pre prienik pri CFL je zalozena na nedeterminizme.

9. DCFL je uzavreny na komplement.

stavy budu dvojice: $Q \times \{p, n, f\}$. Druhy prvok zaznamenava, ci automat presiel konecnym stavom vramci postupnosti ε-krokov (p-prosel, n-neprosel) Ak po ε-krokoch presiel do $[q, n]$, zmeni sa na $[q,f]$ a vykona jeden krok. Automat akceptuje, len ak sa dostane do stavu $[q,f] \in F'$

Priklady:

8.2 Je daný ZA $A = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b, c, d\}, \{X, Y, Z\}, \delta, q_0, Z, \{q_2, q_4\})$, kde

$$\begin{aligned}
\delta(q_0, a, Z) &= \{(q_0, X)\} & \delta(q_0, a, X) &= \{(q_0, XX), (q_1, YX)\} \\
\delta(q_1, a, Y) &= \{(q_1, YY)\} & \delta(q_1, b, Y) &= \{(q_2, \varepsilon)\} \\
\delta(q_2, b, Y) &= \{(q_2, \varepsilon)\} & \delta(q_2, c, X) &= \{(q_3, \varepsilon)\} \\
\delta(q_3, c, X) &= \{(q_3, \varepsilon)\} & \delta(q_3, d, X) &= \{(q_4, \varepsilon)\}
\end{aligned}$$

a) Popište jazyk akceptovaný automatem, pokud $F = \{q_2\}$.

8.2 a) $\{a^i b^j \mid i > j > 0\}$

9.1 O každé z následujících implikací rozhodněte, zda je pravdivá

a) L_1, L_2 bezkontextové $\Rightarrow$ $L_1 \cup L_2$ je kontextový

b) L_1 bezkontextový $\wedge$ $L_1 \cap L_2$ není bezkontextový $\Rightarrow$ L_2 není bezkontextový

c) L_1 regulární $\wedge$ L_2 bezkontextový $\Rightarrow$ $co{-}(L_1 \cap L_2)$ bezkontextový

d) L_1 konečný $\wedge$ L_2 bezkontextový $\Rightarrow$ $co{-}(L_1 \cap L_2)$ bezkontextový

a) ano. CFL su uzavrete na zjednotenie, a teda L ich zjednotenia je opet CFL. Z hierarchie plati : kazdy CFL jazyk je zaroven aj kontextovym.

b) ano.bezkontextove jazyky su uzavrete na prienik a teda ak po prieniku jazyk nie je bezkontextovy, tak aj L2 nie je bezkontextovy.

c) nie. regularni jazyk je Σ^* , bezkontextovy je co-($a^n b^n c^n$). komplement ich prieniku je ($a^n b^n c^n$), ktory nie je CFL.

d) ano. prienik z konecnym jazykom je konecny. kazdy konceny jazyk je regularny, a teda je jeho komplement tiez regularny. regularne jazyky su podmnozinou bezkontextovych.