Učební texty k státní bakalářské zkoušce
Obecná informatika
Automaty a jazyky

8. června 2011

2 Automaty a jazyky

Požadavky

- Chomského hierarchie, třídy automatů a gramatik, determinismus a nedeterminismus.
- Uzávěrové vlastnosti tříd jazyků.

2.1 Automaty – Chomského hierarchie, třídy automatů a gramatik, determinismus a nedeterminismus.

- Popište jednotlivé třídy jazyků a jejich vztahy; definujte třídy pomocí odpovídajících gramatik. Napište priklady gramatik pro jednotlivé třídy.
- Popište automaty, ktere tyto tridv jazyku rozpoznávají i s ohledem na jejich (ne)deterministicnost.

Třídy automatů a gramatik

Definice *(Konečný automat)*
Konečný automat je pětice $A = (Q, X, \delta, q_0, F)$, kde Q je stavový prostor (množina všech možných stavů), X je *abeceda* (množina symbolů), δ je přechodová funkce $\delta : Q \times X \to Q$, $q_0 \in Q$ je poč. stav a $F \subseteq Q$ množina koncových stavů.

Definice
Slovo w je posloupnost symbolů v abecedě X. *Jazyk* L je množina slov, tedy $L \subseteq X^*$, kde X^* je množina všech posloupností symbolů abecedy X. λ je prázdná posloupnost symbolů. *Rozšířená přechodová funkce* je $\delta^* : Q \times X^* \to Q$ - tranzitivní uzávěr δ. Jazyk rozpoznávaný konečným automatem – *regularní jazyk* je $L(A) = \{w | w \in X^*, \delta^*(q_0, w) \in F\}$. *Pravá kongruence* je taková relace ekvivalence na X^*, že $\forall u, v, w \in X^* : u \sim v \Rightarrow uw \sim vw$.[1] Je *konečného indexu*, jestliže $X^*/\sim$ (rozklad na třídy ekvivalence) má konečný počet tříd.

Věta *(Nerodova)*
Jazyk L nad konečnou abecedou X je rozpoznatelný kon. automatem $\Leftrightarrow$ existuje pravá kongruence konečného indexu $\sim$ na X^* tak, že L je sjednocením jistých tříd rozkladu $X^*/\sim$.[2]

Věta *(Pumping (iterační) lemma)*
Pro jazyk rozpoznatelný kon. automatem (tzn. regulární) L existuje $n \in \mathbb{N}$ tak, že libovolné slovo $z \in L, |z| \geq n$ lze psát jako uvw, kde $|uv| \leq n$, $|v| \geq 1$ a $\forall i \geq 0 : uv^i w \in L$.[3]

Definice
Dva automaty jsou *ekvivalentní*, jestliže přijímají stejný jazyk. *Homomorfismus (isomorfismus)* automatů je zobrazení, zachovávající poč. stav, přech. funkci i konc. stavy (+ prosté a na). Pokud existuje homomorfismus automatů $A \to B$, pak jsou tyto dva ekvivalentní (jen 1 implikace!). *Dosažitelný stav* q - $\exists w \in X^* : \delta^*(q_0, w) = q$. Relace ekvivalence je *automatovou kongruencí*, pokud zachovává konc. stavy a přech. funkci. Ke každému automatu existuje *redukt* - ekvivaletní automat bez nedosažitelných a navzájem ekvivalentních stavů. Ten je určen jednoznačně pro daný jazyk (až na isomorfismus), proto lze zavést normovaný tvar.

[1] Pokud dvě různá slova u, v převedou automat do stejného stavu (=jsou navzájem ekvivalentní ($u \sim v$)), pak musí patřit do stejné třídy rozkladu. Pokud k těmto dvěma slovům přidáme stejné slovo zprava, pak tato zřetězená slova budou opět patřit do stejné třídy rozkladu (=musí být navzájem ekvivalentní ($uw \sim vw$)). A toto je právě ta vlastnost definující pravou kongruenci.

[2] Důležité tedy je, že pokud je jazyk regulární, pak pro něj musí existovat pravá kongruence, která (což je nejdůležitější) rozkládá všechna slova jazyka do konečně mnoha tříd.

[3] Platí i pro konečné jazyky: když je jazyk konečný, tak si za n stačí vzít délku nejdelšího slova a pak to pro všechny slova delší než n (tj. žádná) platí taky.

Poznámka *(Operace s jazyky)*

S jazyky lze provádět množinové operace ($\cup, \cap$), rozdíl ($\{w|w \in L_1 \& w \notin L_2\}$), doplněk ($\{w|w \notin L\}$), dále zřetězení ($L_1 \cdot L_2 = \{uv|u \in L_1, v \in L_2\}$), mocniny ($L^0 = \lambda, L^{i+1} = L^i \cdot L$), iterace ($L^* = L^0 \cup L^1 \cup L^2 \cup ...$), otočení ($L^R = \{u^R|u \in L\}$), levý ($L_2 \backslash L_1 = \{v|uv \in L_1, u \in L_2\}$) i pravý ($L_1/L_2 = \{u|uv \in L_1, v \in L_2\}$) kvocient L_1 podle L_2 a derivace (kvocienty podle jednoslovného jazyka). Třída jazyků rozpoznatelných konečnými automaty je na tyto operace uzavřená.

Definice *(Regulární jazyky)*

Třída reguláních jazyků nad abecedou X je nejmenší třída, která obsahuje $\emptyset$, $\forall x \in X$ obsahuje x a je uzavřená na sjednocení, iteraci a zřetězení.

Věta *(Kleenova)*

Jazyk je regulární $\Leftrightarrow$ je rozpoznatelný konečným automatem.[4]

Definice *(Regulární výrazy)*

Regulární výrazy nad abecedou $X = x_1, ..., x_n$ jsou nejmenší množina slov v abecedě $x_1, ..., x_n, \emptyset, \lambda, ^+, \cdot, ^*, (,)$, která obsahuje výrazy $\emptyset$ a λ a $\forall i$ obsahuje x_i a je uzavřená na sjednocení ($+$), zřetězení ($\cdot$) a iterace (*). *Hodnota reg. výrazu* a je reg. jazyk $[a]$, lze takto reprezentovat každý reg. jazyk.

Definice *(Dvoucestné konečné automaty)*

Dvoucestný konečný automat je pětice (Q, X, δ, q_0, F), kde oproti kon. automatu je $\delta : Q \times X \to Q \times \{-1, 0, 1\}$ (tj. pohyb čtecí hlavy). Přijímá slovo, pokud výpočet začal vlevo v poč. stavu a čtecí hlava opustila slovo w vpravo v konc. stavu (mimo slovo končí výpočet okamžitě).

Poznámka

Jazyky přijímané dvoucestnými automaty jsou regulární - každý dvoucestný automat lze převést na (nedeterministický) konečný automat.

Definice *(Zásobníkové automaty)*

Zásobníkový automat je sedmice $M = (Q, X, Y, \delta, q_0, Z_0, F)$, kde proti konečným automatům je Y abeceda pro symboly na zásobníku, Z_0 počáteční symbol na zásobníku a funkce instrukcí $\delta : Q \times (X \cup \{\lambda\}) \times Y \ \to \ \mathcal{P}(Q \times Y^*)$. Je z principu nedeterministický; vždy se nahrazuje vrchol zásobníku, nečte ale pokaždé vstupní symboly. Instrukci $(p, a, Z) \to (q, w)$ lze vykonat, pokud je automat ve stavu p, na zásobníku je Z a na vstupu a. Vykonání instrukce znamená změnu stavu, pokud $a \neq \lambda$, tak i posun čtecí hlavy a odebrání Z ze zásobníku, kam se vloží w (prvním písmenem nahoru). Výpočet končí buď přečtením slova, nebo v případě, že pro danou situaci není definována instrukce (*Situace* zás. automatu je trojice (p, u, v), kde $p \in Q$, u je nepřečtený zbytek slova a v celý zásobník).

Přijímat slovo je možné buď koncovým stavem (slovo je přečteno a automat v konc. stavu), nebo zásobníkem (slovo je přečteno a zásobník prázdný – konc. stavy jsou v takovém případě nezajímavé - $F = \emptyset$).

Poznámka

Pro zás. automat přijímající konc. stavem vždy existuje ekvivalentní automat ($L(A_1) = L(A_2)$) přijímající zásobníkem a naopak.

Definice *(Přepisovací systém)*

Přepisovací (produkční) systém je dvojice $R = (V, P)$, kde V je konečná abeceda a P množina přepisovacích pravidel (uspořádaných dvojic prvků z V^*). Slovo w se *přímo přepíše* na z ($w \Rightarrow z$), pokud $\exists u, v, x, y \in V^* : w = xuy, z = xvy, (u, v) \in P$. Derivace (*odvození*) je zřetězení několika přímých přepsání.

[4] Důkaz se dá indukcí podle počtu hran v nedeterministickém automatu.

Definice *(Formální (generativní) gramatika)*

Formální gramatika je čtveřice $G = (V_N, V_T, S, P)$, kde V_N je množina neterminálních symbolů (ostatní znaky např. S), V_T množina terminálních symbolů ("znaky z abecedy"), S startovací symbol ($S \in V_N$) a P množina pravidel. *Jazyk generovaný gramatikou* je $L(G) = \{w | w \in V_T^*, S \Rightarrow^* w\}$.

Věta

Každý bezkontextový jazyk je rozpoznáván zásobníkovým automatem, přijímajícím prázdným zásobníkem. Stejně pro každý zásobníkový automat existuje bezkontextová gramatika, která generuje jazyk jím přijímaný.

Poznámka *(Vlastnosti bezkontextových gramatik)*

Bezkontextová gramatika je *redukovaná*, pokud $\forall X \in V_N$ existuje terminální slovo $w \in V_T^*$ tak, že $X \Rightarrow^* w$ a navíc $\forall X \in V_N, X \neq S$ existují slova u, v tak, že $S \Rightarrow^* uXv$. Ke každé bezkontextové gramatice lze sestrojit ekvivalentní redukovanou.

Pro každé terminální slovo v bezkontextové gramatice existují derivace, které se liší jen pořadím použití pravidel (a prohozením některých pravidel dostanu stejné terminální slovo), proto lze zavést *levé (pravé) derivace* - tj. *kanonické* derivace. Pokud $X \Rightarrow^* w$, pak existuje i levá (pravá) derivace. Znázornění průběhu derivací je možné určit *derivačním stromem* – určuje jednoznačně pravou/levou derivaci.

Bezkontextová gramatika je *víceznačná* (nejednoznačná), pokud v ní existuje slovo, které má dvě různé levé derivace; jinak je *jednoznačná*. Jazyk je jednoznačný, pokud k němu existuje generující jednoznačná gramatika. Pokud je každá gramatika nějakého jazyka nejednoznačná, je tento *podstatně nejednoznačný*.

Definice *(Greibachové normální forma)*

Gramatika je v *Greibachové normální formě*, jsou-li všechna její pravidla ve tvaru $A \to au$, kde $a \in V_T$ a $u \in V_N^*$. Ke každému bezkontextovému jazyku existuje gramatika v G. normální formě tak, že $L(G) = L \setminus \{\lambda\}$. Každou bezkontextovou gramatiku lze převést do G. normální formy.

Poznámka *(Úpravy bezkontextových gramatik)*

Spojením více pravidel ($A \to uBv, B \to w_1, ...B \to w_k$ se převede na $A \to uw_1v|...|uw_kv$) dostanu ekvivalentní gramatiku. Stejně tak odstraněním levé rekurze (převod přes nový neterminál).

Definice *(Chomského normální forma)*

Pro gramatiku v *Chomského normální formě* jsou všechna pravidla tvaru $X \to YZ$ nebo $X \to a$, kde $X, Y, Z \in V_N$, $a \in V_T$. Ke každému bezkontextovému jazyku L existuje gramatika G v Chomského normální formě tak, že $L(G) = L \setminus \{\lambda\}$

Poznámka *(Vlastnosti třídy bezkontextových jazyků)*

Třída bezkontextových jazyků je uzavřená na sjednocení, zrcadlení, řetězení, iteraci a pozitivní iteraci, substituci a homomorfismus, inverzní homomorfismus a kvocient s regulárním jazykem. Není uzavřená na průnik a doplněk.

Definice *(Dyckův jazyk)*

Dyckův jazyk je definován nad abecedou $a_1, a_1', ...a_n, a_n'$ gramatikou

$$S \to \lambda | SS | a_1 S a_1' | ... | a_n S a_n'$$

Je bezkontextový, popisuje správná uzávorkování a lze jím popisovat výpočty zásobníkových automatů, tedy i bezkontextové jazyky.

Definice *(Turingův stroj)*

Turingův stroj je pětice $T = (Q, X, \delta, q_0, F)$, kde X je abeceda, obsahující symbol ε pro prázdné políčko, přechodová funkce $\delta : (Q \setminus F) \times X \to Q \times X \times \{-1, 0, 1\}$ popisuje změnu stavu, zápis na pásku a posun hlavy. Výpočet končí, není-li definována žádná instrukce (spec. platí pro $q \in F$). *Konfigurace* Turingova stroje jsou údaje, popisující stav výpočtu – nejmenší souvislá část pásky, obsahující všechny neprázdné buňky a čtenou buňku, vnitřní stav a poloha hlavy. *Krok výpočtu* je $uqv \vdash wpz$ pro u část slova vlevo od akt. pozice na pásce, v od čteného písmena dál a q stav stroje. *Výpočet* je posloupnost kroků, slovo w je přijímáno, pokud $q_0 w \vdash^* upv$, $p \in F$. Jazyky (množiny slov bez ε) přijímané Turingovými stroji jsou *rekurzivně spočetné.*

Věta

Každý jazyk typu 0 (s gramatikou s obecnými pravidly) je rekurzivně spočetný.

Chomského hierarchie

Definice *(Chomského hierarchie)*

Chomského hierarchie je rozdělení gramatik do 4 tříd podle omezení na pravidla:

Zařazení do Chomskeho hierarchie	Gramatiky	Jazyky	Automaty	Pravidla
Typu 0	Gramatiky typu 0	Rekurzivně spočetné jazyky	Turingův stroj	Pravidla v obecné formě (tj. $u \to v$, kde $u,v \in (V_N \cup V_T)^*$ a u obsahuje alespoň 1 neternimální symbol)
není	(není společný název)	Rekurzivní jazyky	Vždy zastavující Turingův stroj	
Typu 1	Kontextové gramatiky	Kontextové jazyky	Lineárně omezené automaty	Pouze pravidla ve tvaru $\alpha X\beta \to \alpha w\beta$, $X \in V_N$; $\alpha,\beta \in (V_N \cup V_T)^*$; $w \in (V_N \cup V_T)^+$ Jedinou výjimkou je pravidlo $S \to \lambda$, potom se ale S nevyskytuje na pravé straně žádného pravidla.
Typu 2	Bezkontextové gramatiky	Bezkontextové jazyky	(Nedeterministický) Zásobníkový automat	Pouze pravidla ve tvaru $X \to w$, $X \in V_N$; $w \in (V_N \cup V_T)^*$
není	Deterministické bezkontextové gramatiky	Deterministické bezkontextové jazyky	Deterministický zásobníkový automat	
Typu 3	Regulární gramatiky	Regulární (pravé lineární) jazyky	Konečný automat	Pouze pravidla ve tvaru $X \to wY$, $X \to w$; $X,Y \in V_N$; $w \in V_T^*$

Každá kategorie jazyků nebo gramatik je podmnožinou jazyků nebo gramatik kategorie přímo nad ní, a jakýkoli automat v každé kategorii má ekvivaletní automat v kategorii přímo nad ním.

"není" znamená že nepatří do Chomskeho hierarchie.

Z originálu: http://en.wikipedia.org/wiki/Template:Formal_languages_and_grammars

Bezprefixový jazyk

L je bezprefixový pokud, neexistuje slovo $u \in L$ takové, že rovněž $uw \in L$, $w \in X^+$

Poznámka

S $\mathcal{L}1 \supset \mathcal{L}2$ nastává problém, protože bezkontextové gramatiky umožňují pravidla tvaru $X \to \lambda$. Řešením je převod na *nevypouštějící bezkontextové gramatiky* - takové bezkontextové gramatiky, které nemají pravidla typu $X \to \lambda$.

Věta *(o nevypouštějících bezkontextových gramatikách)*

Ke každé bezkontextové G existuje nevypouštějící bezkontextová G_0 tak, že

$$L(G_0) = L(G) \setminus \{\lambda\}$$

Je-li $\lambda \in L(G)$, pak $\exists G_1$, t.ž. $L(G_1) = L(G)$ a jediné pravidlo v G_1 s λ na pravé straně je $S \to \lambda$ a S není v G_1 na pravé straně žádného pravidla.

Poznámka *(Lineární gramatiky)*

Pro každou gramatiku typu G3 lze sestrojit konečný automat, který přijímá právě jazyk jí generovaný, stejně tak pro každý konečný automat lze sestrojit gramatiku G3. Levé lineární gramatiky také generují regulární jazyky, díky uzavřenosti na reverzi. *Lineární gramatiky*, s pravidly typu $X \to uYv, X \to w$, kde $X, Y \in V_N, u, v, w \in V_T^*$, generují *lineární jazyky* - silnější než regulární jazyky.

Definice *(Separovaná a nevypouštějící gramatika)*

Separovaná gramatika je gramatika (obecně libovolné třídy), obsahující pouze pravidla tvaru $\alpha \to \beta$, kde buď $\alpha, \beta \in V_N^+$, nebo $\alpha \in V_N$ a $\beta \in V_T \cup \{\lambda\}$. *Nevypouštějící (monotónní) gramatika* (také se neomezuje na konkrétní třídu) je taková, že pro každé pravidlo $u \to v$ platí $|u| \leq |v|$.

Poznámka *(Kontextové gramatiky)*

Ke každé kontextové gramatice lze sestrojit ekvivalentní separovanou. Ke každé monotónní gramatice lze nalézt ekvivalentní kontextovou.

Determinismus a nedeterminismus

Definice *(Nedeterministický konečný automat)*

Nedeterministický konečný automat je pětice (Q, X, δ, S, F), kde Q je mn. stavů, X abeceda, F mn. konc. stavů, S množina počátečních stavů a $\delta : Q \times X \to \mathcal{P}(Q)$ je přechodová funkce. Slovo w je takovým automatem přijímáno, pokud existuje posloupnost stavů $\{q_i\}_{i=1}^n$ tak, že $q_1 \in S$, $q_{i+1} \in \delta(q_i, x_i)$, $q_{n+1} \in F$.

Poznámka

Pro každý nedeterministický konečný automat A lze sestrojit deterministický kon. automat B tak, že jimi přijímané jazyky jsou ekvivalentní (může to znamenat exponenciální nárůst počtu stavů).

Definice *(Deterministický zásobníkový automat)*

Deterministický zásobníkový automat je $M = (Q, X, Y, \delta, q_0, Z_0, F)$ takové, že $\forall p \in Q, \forall a \in (X \cup \{\lambda\}), \forall Z \in Y$ platí $|\delta(p, a, Z)| \leq 1$[5] a navíc pokud pro nějaké p, Z je $\delta(p, \lambda, Z) \neq \emptyset$, pak $\forall a \in X$ je $\delta(p, a, Z) = \emptyset$[6].

Poznámka

Deterministický zásobníkový automat je slabší než nedeterministický, rozpoznává *deterministické bezkontextové jazyky* koncovým stavem a *bezprefixové bezkontextové jazyky* prázdným zásobníkem (takové jazyky, kde $u \in L(M) \Rightarrow \forall w \in X^* : uw \notin L(M)$) - když se poprvé zásobník automatu vyprázdní, výpočet určitě končí.

Bezprefixové bezkontextové jazyky jsou vždy deterministické, opačně to neplatí. Deterministický bezkontextový jazyk lze na bezprefixový převést zřetězením s dalším symbolem, který není v původní abecedě.

Regulární jazyky a bezprefixové bezkontextové jazyky jsou neporovnatelné množiny.

Definice *(Nedeterministický Turingův stroj)*

Nedet. Turingův stroj je pětice $T = (Q, X, \delta, q_0, F)$, kde oproti deterministickým je $\delta : (Q \setminus F) \times X \to \mathcal{P}(Q \times X \times \{-1, 0, 1\})$. Přijímá slovo w, pokud existuje nějaký výpočet $q_0 w \vdash^* upv$ tak, že $p \in F$.

---

[5] definuje ze v kazdem kroku si nemuzeme vybirat

[6] definuje ukončení vypoctu

Poznámka

Nedeterministické Turingovy stroje přijímají právě rekurzivně spočetné jazyky, tj. nejsou silnější než deterministické. Výpočty nedet. stroje lze totiž díky nekonečnosti pásky simulovat deterministickým (např. prohledáváním do šířky).

Definice *(Lineárně omezený automat)*

Lineárně omezený automat je nedeterministický Turingův stroj s omezenou páskou (např. symboly l a r, které nelze přepsat ani se dostat mimo jejich rozmezí). Slovo je přijímáno, pokud $q_0lwr \vdash^* upv$, kde $p \in F$. Prostor výpočtu je omezen délkou vstupního slova. Lineárně omezené automaty přijímají právě kontextové jazyky.

Poznámka *(Rozhodnutelnost)*

Turingův stroj může nepřijmout slovo buď skončením výpočtu v nekoncovém stavu, nebo pokud výpočet nikdy neskončí. Turingův stroj *rozhoduje jazyk L*, když přijímá právě slova tohoto jazyka a pro libovolné slovo je jeho výpočet konečný. Takové jazyky se nazývají *rekurzivní.*

Problém zastavení výpočtu Turingova stroje je algoritmicky nerozhodnutelný (kvůli možnosti jeho simulace jiným Turingovým strojem). Pro bezkontextové jazyky je algoritmicky rozhodnutelné, zda dané slovo patří do jazyka. Pro bezkontextovou gramatiku nelze algoritmicky rozhodnout, zda $L(G) = X^*$. Pro dvě kontextové gramatiky je nerozhodnutelné, zda jejich jazyky mají neprázdný průnik.

Report *(Hnetynka)*

napisal som hierachiu, pravidla gramatik, ake automaty rozpoznavaju jednotlive gramatiky, pre reg gram veticky o KA a reg jazykoch. Pytal sa ma ako jednotlive automaty pracuju, zadal par jednoduchych prikladov a chcel zdovodnenie do akych tried patri -nakreslit/opisat slovami KA, gramatiky, ZA + dokaz pomocou pumping lemma. Pytal sa, do akej triedy by som zaradil rozpoznavanie prg jazykov, napr Java. Povedal som kontextove, ale zdovodnit som to velmi nevedel. Potvrdil ze su to kontextove + ze prave kvoli dobrym znalostam sa pouzivaju bezkontextove v kombinacii s analyzou kontextu (kedze na poradi jednotlivych riadkov zalezi) (znamka 1-2, ze zalezi na druhej inf otazke)

Report *(Bulej)*

napisal som gramatiky, odpovedajuce automaty, vysvetlil inkluzie a to bohato stacilo

Report *(Bednarek)*

Měl jsem definice hierarchie a srovnání s příslušnými automaty, nástin (opravdu lehce) inkluzí mezi třídami jazyků, Greibachové a Chomského n.f. plus nějaké příklady jazyků, které dokazují ostrou inkluzi, ale bez přesnějších důkazů (kouzelná věta: "to se ukáže přes pumping lemma" ;-)) Ptal se mě na srovnání deterministických a nedeterministických verzí jednotlivých druhů automatů, což jsem věděl. Informatika za jedna.

Report *(Kucera)*

Toto se obeslo bez problemu, stacilo to definovat, a rict ktere automaty prijimaij ktere jazyky, umet je definovat. Vety kolem moc zajem nevzbudily (Nerudovka, pump. lemma):-)

Report *(Fiala)*

Napsal sem třídy automatů, gramatik, determinismus/nedeterminismus a pak ještě rozhodnutelnost. Při procházení sem dostával doplňující otázky typu : "Jak nějak líp omezit odhad o nárustu počtu stavu při převodu NKA do KA" (záleží na počtu počátečních stavů a na počtu "stejnejch" přechodů z jednotlivých stavů.), U rozhodnutelnosti náznak důkazu halting problem a další.

Report *(Bednárek)*
Tak jsem napsal automat a gramatiku ke každé třídě jazyků, determ./nedeterm. verze a jak je to kde s jejich silou. Drobné chyby v definicích nevadily, když byly po upozornění opraveny. U RJ se zeptal ještě na reg. výrazy a pak taky proč že se rekurzivně spočetné jazyky jmenují jak se jmenují (kde je ta rekurze), což jsem nevěděl a byl poučen.

2.2 Uzávěrové vlastnosti tříd jazyků

TODO: nejaka zduvodneni

Uzávěrové vlastnosti v kostce

	RJ	BKJ	DBKJ
Sjednocení	✓	✓	✗
Průnik	✓	✗	✗
Průnik s RJ	✓	✓	✓
Doplněk	✓	✗	✓
Substituce/ homomorfismus	✓	✓	✗
Inverzní homomorfismus	✓	✓	✓

Automaty a gramatiky, Roman Barták

Poznámka *(DBKJ nejsou uzavřené na homomorfismus (BKJ ano)!)*
$L_1 = \{a^i b^j c^k | i = j\}$ je DBKJ
$L_2 = \{a^i b^j c^k | j = k\}$ je DBKJ
$0L_1 \cup 1L_2$ je DBKJ, $1L_1 \cup 1L_2$ není DBKJ
položme $h(0) = 1$ a $h(x) = x$ pro ostatní symboly
$h(0L_1 \cup 1L_2) = 1L_1 \cup 1L_2$

Poznámka *(BKJ nejsou uzavřené na průnik (DBKJ ano))*
• $L_1 = \{a^i b^i c^j | i, j \geq 0\}$
• $L_2 = \{a^i b^j c^j | i, j \geq 0\}$
TODO

Poznámka *(Doplněk deterministického BKJ je opět deterministický BKJ!)*
prohodíme koncové a nekoncové stavy
potíže:
• nemusí přečíst celé vstupní slovo
– krok není definován (např. vyprázdnění zásobníku)
snadno ošetříme „podložkou“ na zásobníku
– cyklus (zásobník roste, zásobník pulsuje)
odhalíme pomocí čítače
• po přečtení slova prochází koncové a nekoncové stavy
stačí si pamatovat, zda prošel koncovým stavem