Učební texty k státní bakalářske zkoušce
Programováni
Základy teoretické informatiky

studenti MFF

18. augusta 2007

Požadavky

- Logika - jazyk, formule, sémantika, tautologie
- Rozhodnutelnost, splnitelnost, pravdivost, dokazatelnost
- Normální tvary výrokových formulí, prenexní tvary formulí predikátové logiky
- Automaty - Chomského hierarchie, třídy automatů a gramatik, determinismus a nedeterminismus.

0.1 Logika – jazyk, formule, sémantika, tautologie

Jazyk

Logika prvního řádu

Formální systém logiky prvního řádu obsahuje jazyk, axiomy, odvozovací pravidla, věty a důkazy. Jazyk prvního řádu zahrnuje:

- neomezeně mnoho proměnných $x_1, x_2, \ldots$
- funkční symboly $f_1, f_2 \ldots$, každý má aritu $n \geq 0$
- predikátové symboly $p_1, p_2 \ldots$, každý má aritu
- symboly pro logické spojky $(\neg, \vee, \&, \rightarrow, \leftrightarrow)$
- symboly pro kvantifikátory $(\forall, \exists)$
- může (ale nemusí) obsahovat binární predikát " = ", který pak musí mít vlastnosti ekvivalence.

Proměnné, logické spojky a kvantifikátory jsou *logické symboly*, ostatní symboly se nazývají *speciální*.

Definice *(Jazyk výrokové logiky)*
Jazyk výrokové logiky je jazyk prvního řádu, obsahující *výrokové proměnné* ("prvotní formule"), *logické spojky* $\neg, \vee, \&, \rightarrow, \leftrightarrow$ a *pomocné symboly* (závorky).

Definice *(Jazyk predikátové logiky)*
Jazyk predikátové logiky je jazyk prvního řádu, obsahující proměnné, *predikátové symboly* (s nenulovou aritou), *funkční symboly* (mohou mít nulovou aritu), symboly pro logické spojky a symboly pro *kvantifikátory*.

Formule

Definice *(Formule výrokové logiky)*
Pro jazyk výrokové logiky jsou následující výrazy formule:

1. každá výroková proměnná
2. pro formule A, B i výrazy $\neg A$, $(A \vee B)$, $(A \& B)$, $(A \rightarrow B)$, $A \leftrightarrow B$
3. každý výraz vzniknuvší konečným užitím pravidel 1. a 2.

Množina formulí se nazývá *teorie*.

Definice *(Term)*
V predikátové logice je *term*:

1. každá proměnná
2. výraz $f(t_1, \ldots, t_1)$ pro f n-ární funkční symbol a $t_1, \ldots, t_n$ termy
3. každý výraz vzniknuvší konečným užitím pravidel 1. a 2.

Podslovo termu, které je samo o sobě term, se nazývá *podterm*.

Definice *(Formule predikátové logiky)*
V predikátové logice je formule každý výraz tvaru $p(t_1, \ldots, t_n)$ pro p predikátový symbol a $t_1, \ldots, t_n)$ termy. Stejně jako ve výrokové logice je formule i (konečné) spojení jednodušších formulí log. spojkami. Formule jsou navíc i výrazy $(\exists x)A$ a $(\forall x)A$ pro formuli A.

Podslovo formule, které je samo o sobě formule, se nazývá *podformule*.

Definice *(Volné a vázané proměnné)*
Výskyt proměnné x ve formuli je *vázaný*, je-li tato součástí nějaké podformule tvaru $(\exists x)A$ nebo $(\forall x)A$. V opačném případě je *volný*. Formule je *otevřená*, pokud neobsahuje vázanou proměnnou, je *uzavřená*, když neobsahuje volnou proměnnou. Proměnná může být v téže formuli volná i vázaná (např. $(x = z) \rightarrow (\exists x)(x = z)$).

Sémantika

Definice *(Pravdivostní ohodnocení ve výrokové logice)*
Výrokové proměnné samotné neanalyzujeme – jejich hodnoty máme dány už zvnějšku, máme pro ně *množinu pravdivostních hodnot* ($\{0, 1\}$).

Pravdivostní ohodnocení v je zobrazení, které každé výrokové proměnné přiřadí právě jednu hodnotu z množiny pravdivostních hodnot. Je-li známo ohodnocení proměnných, lze určit *pravdivostní hodnotu* $\overline{v}$ pro každou formuli (při daném ohodnocení) – indukcí podle její složitosti, podle tabulek pro logické spojky.

Definice *(Interpretace jazyka predikátové logiky)*
Interpretace jazyka je definována množinovou strukturou $\mathcal{M}$, která ke každému symbolu jazyka a množině proměnných přiřadí nějakou množinu individuí. $\mathcal{M}$ obsahuje:

- neprázdnou množinu individuí M.
- zobrazení $f_M : M^n \rightarrow M$ pro každý $n-$ární funkční symbol f
- relaci $p_M \subset M^n$ pro každý $n-$ární predikát p

Interpretace termů se uvažuje pro daný jazyk L a jeho interpretaci $\mathcal{M}$. *Ohodnocení proměnných* je zobrazení $e : X \rightarrow M$ (kde X je množina proměnných). *Interpretace termu* t při ohodnocení e - $t[e]$ se definuje následovně:

- $t[e] = e(x)$ je-li t proměnná x
- $t[e] = f_M(t_1[e], \ldots, t_n[e])$ pro term tvaru $f(t_1, \ldots, t_n)$.

Ohodnocení závisí na zvoleném $\mathcal{M}$, interpretace termů při daném ohodnocení pak jen na konečně mnoha hodnotách z něj. Pokud jsou $x_1, \ldots, x_n$ všechny proměnné termu t a e, e' dvě ohodnocení tak, že $e(x_i) = e'(x_i) \forall i \in \{1, \ldots, n\}$, pak $t[e] = t[e']$.

Pozměněné ohodnocení y pro $x = m \in M$ je definováno:

$$e(x/m)(y) = \begin{cases} m(\text{pro } y \equiv x) \\ e(y)(\text{jinak}) \end{cases}$$

Definice *(Substituce)*
Substituce proměnné za podterm v termu ($t_{x_1,\ldots,x_n}[t_1, \ldots, t_n]$) je současné nahrazení všech výskytů proměnných x_i termy t_i. Je to term. *Instance* formule je současné nahrazení všech volných výskytů nějakých proměnných za termy. Je to taky formule, vyjadřuje speciálnější tvrzení – ne vždy ale lze provést substituci bez změny významu formule. Term je *substituovatelný* do formule A za proměnnou x, pokud pro $\forall y$ vyskytující se v t žádná podformule formule A tvaru $(\exists y)B$ ani $(\forall y)B$ neobsahuje volný výskyt x.

Definice *(Uzávěr formule)*
Jsou-li $x_1, \ldots, x_n$ všechny proměnné s volným výskytem ve formuli A, potom $(\forall x_1) \ldots (\forall x_n)A$ je *uzávěr* formule A.

Tautologie

Definice *(Tautologie)*
Formule je *tautologie*, jestliže je pravdivá při libovolném ohodnocení proměnných ($\models A$).

Definice *(Tautologický důsledek)*
Teorie U je *tautologický důsledek* teorie T, jestliže každý model T je také modelem U ($T \models U$).

0.2 Rozhodnutelnost, splnitelnost, pravdivost a dokazatelnost

Rozhodnutelnost

Definice *(Rekurzivní funkce a množina)*
Rekurzivní funkce jsou všechny funkce popsatelné jako $f : \mathbb{N}^k \to \mathbb{N}$, kde $k \geq 1$, tedy všechny "algoritmicky vyčíslitelné" funkce. Množina přirozených čísel je *rekurzivní množina (rozhodnutelná množina)*, pokud je rekurzivní její charakteristická funkce.

Definice *(Spočetný jazyk, kód formule)*
Spočetný jazyk je jazyk, který má nejvýš spočetně mnoho speciálních symbolů. Pro spočetný jazyk, kde lze efektivně (rekurzivní funkcí) očíslovat jeho speciální symboly, lze každé jeho formuli A přiřadit její *kód formule* - přir. číslo $\#A$.

Věta *(Churchova o nerozhodnutelnosti predikátové logiky)*
Pokud spočetný jazyk L prvního řádu obsahuje alespoň jednu konstantu, alespoň jeden funkční symbol arity $k > 0$ a pro každé přirozené číslo spočetně mnoho predikátových symbolů, potom množina $\{\#A | A$ je uzavřená formule a $L \models A\}$ není rozhodnutelná.

Věta *(o nerozhodnosti predikátové logiky)*
Nechť L je jazyk prvního řádu bez rovnosti a obsahuje alespoň 2 binární predikáty. Potom je predikátová logika (jako teorie) s jazykem L nerozhodnutelná.

Definice *(Tři popisy aritmetiky)*
Je dán jazyk $L = \{0, S, +, \cdot\}$.

- *Robinsonova aritmetika* - "Q" s jazykem L má 8 násl. axiomů:
  1. $S(x) \neq 0$
  2. $S(x) = S(y) \rightarrow x = y$
  3. $x \neq 0 \rightarrow (\exists y)(x = S(y))$
  4. $x + 0 = x$
  5. $x + S(y) = S(x + y)$
  6. $x \cdot 0 = 0$
  7. $x \cdot S(y) = (x \cdot y) + x$
  8. $x \leq y \leftrightarrow (\exists z)(z + x = y)$
- *Peanova aritmetika* - "P" má všechny axiomy Robinsonovy kromě třetího, navíc má *Schéma(axiomů) indukce* - pro formuli A a proměnnou x platí: $A_x[0] \rightarrow \{(\forall x)(A \rightarrow A_x[S(x)]) \rightarrow (\forall x)A\}$.
- *Úplná aritmetika* má za axiomy všechny uzavřené formule pravdivé v $\mathbb{N}$, je-li $\mathbb{N}$ standardní model aritmetiky - "pravdivá aritmetika". *Teorie modelu* $\mathbb{N}$ je množina $Th(\mathbb{N}) = \{A | A$ je uzavřená formule a $N \models A\}$.

Platí: $Q \subseteq P \subseteq Th(\mathbb{N})$. Q má konečně mnoho axiomů, je tedy rekurzivně axiomatizovatelná. P má spočetně mnoho axiomů, kódy axiomů schématu indukce tvoří rekurzivní množinu. $Th(\mathbb{N})$ není rekurzivně axiomatizovatelná.

Definice *(Množina kódů vět teorie)*
Pro T teorii s jazykem aritmetiky definujeme *množinu kódů vět teorie* T jako $Thm(T) = \{\#A | A$ je uzavřená formule a $T \vdash A\}$.

Definice *(Rozhodnutelná teorie)*
Teorie T s jazykem aritmetiky je *rozhodnutelná*, pokud je množina $Thm(T)$ rekurzivní. V opačném případě je T *nerozhodnutelná*.

Věta *(Churchova o nerozhodnutelnosti aritmetiky)*
Každé bezesporné rozšíření Robinsonovy aritmetiky Q je nerozhodnutelná teorie.

Věta *(Gödel-Rosserova o neúplnosti aritmetiky)*
Žádné bezesporné a rekurzivně axiomatizovatelné rozšíření Robinsonovy aritmetiky Q není úplná teorie.

Splnitelnost

Definice *(Splnitelnost)*
Množina formulí ve výrokové logice je *splnitelná*, jestliže existuje ohodnocení v, tak. že $\forall A \in T$ je pravdivá při v. Potom se v nazývá *model teorie* T.

Pravdivost

Definice *(Pravdivá formule výrokové logiky)*
Formule výrokové logiky A je *pravdivá* při ohodnocení v, je-li $\overline{v}(A) = 1$, jinak je *nepravdivá*. Je-li formule A pravdivá při ohodnocení v, pak říkáme, že v *je model* A ($v \models A$).

Definice *(Tarského definice pravdy)*
Pro daný jazyk predikátové logiky L, $\mathcal{M}$ jeho interpretaci, ohodnocení e a A formuli tohoto jazyka platí:

1. A je *splněna v ohodnocení* e ($\mathcal{M} \models A[e]$), když:
   - A je atomická tvaru $p(t_1, \ldots, t_n)$, kde p není rovnost a $(t_1[e], \ldots, t_n[e]) \in p_M$.
   - A je atomická tvaru $t_1 = t_2$ a $t_1[e] = t_2[e]$
   - A je tvaru $\neg B$ a $\mathcal{M} \not\models B[e]$
   - A je tvaru $B \to C$ a $\mathcal{M} \not\models B[e]$ nebo $\mathcal{M} \models C[e]$
   - A je tvaru $(\forall x)B$ a $\mathcal{M} \models B[e(x/m)]$ pro každé $m \in M$
   - A je tvaru $(\exists x)B$ a $\mathcal{M} \models B[e(x/m)]$ pro nějaké $m \in M$
2. A je *pravdivá v interpretaci* $\mathcal{M}$ ($\mathcal{M} \models A$), jestliže je A splněna v M při každém ohodnocení proměnných (pro uzavřené formule stačí jedno ohodnocení, splnění je vždy stejné)

Definice *(Logicky pravdivá formule predikátové logiky)*
Formule A je *validní (logicky pravdivá)* ($\models A$), když je platná při každé interpretaci daného jazyka.

Dokazatelnost

Definice *(Axiomy výrokové logiky)*
Pro redukovaný jazyk výrokové logiky (po snížení počtu log. spojek na základní ($\to$, $\neg$)) jsou *axiomy výrokové logiky* (schémata axiomů) všechny formule následujících tvarů:

- $(A \to (B \to A))$ (A1 - "implikace sebe sama")
- $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$ (A2 - "roznásobení")
- $(\neg B \to \neg A) \to (A \to B)$ (A3 - "obrácená negace implikace")

Definice *(Odvozovací pravidlo výrokové logiky)*
Výroková logika má jediné odvozovací pravidlo – *modus ponens*:

$$\frac{A, A \to B}{B}$$

Definice *(Důkaz ve výrokové logice)*
Důkaz A je konečná posloupnost formulí $A_1, \ldots A_n$, jestliže $A_n = A$ a pro každé $i = 1, ..n$ je A_i buď axiom, nebo je odvozená z předchozích pravidlem modus ponens. Existuje-li důkaz formule A, pak je tato *dokazatelná* ve výrokové logice (je větou výrokové logiky - $\vdash A$).

Definice *(Důkaz z předpokladů)*
Důkaz formule A z předpokladů je posloupnost formulí $A_1, \ldots A_n$ taková, že $A_n = A$ a $\forall i \in \{1, ..n\}$ je A_i axiom, nebo prvek množiny předpokladů T, nebo je odvozena z přechozích pravidlem modus ponens. Existuje-li důkaz A z T, pak A *je dokazatelná* z T - $T \vdash A$.

Věta *(o dedukci)*
Pro T množinu formulí a formule A, B platí:

$$T \vdash A \to B \text{ právě když } T, A \vdash B$$

Definice
Množina formulí T je *sporná*, pokud je z předpokladů T dokazatelná libovolná formule, jinak je T *bezesporná*. T je *maximální bezesporná* množina, pokud je T bezesporná a navíc jediná její bezesporná nadmnožina je T samo. Množina všech formulí dokazatelných z T se značí $Con(T)$.

Věta *(Lindenbaumova)*
Každou bezespornou množinu formulí výrokové logiky T lze rozšířit na maximální bezespornou S, $T \subset S$.

Věta *(o bezespornosti a splnitelnosti)*
Množina formulí výrokové logiky je bezesporná, právě když je splnitelná.

Věta *(Věta o úplnosti výrokové logiky)*
Je-li T množina formulí a A formule, pak platí:

1. $T \vdash A$ právě když $T \models A$
2. $\vdash A$ právě když $\models A$ (A je větou výrokové logiky, právě když je tautologie)

Tedy výroková logika je bezesporná a jsou v ní dokazatelné právě tautologie.

Věta *(o kompaktnosti)*
Množina formulí výrokové logiky je splnitelná, právě když je splnitelná každá její konečná podmnožina.

Definice *(Formální systém predikátové logiky)*
Pracujeme s redukovaným jazykem (jen s log. spojkami $\neg, \rightarrow$ a jen s kvantifikátorem $\forall$). *Schémata Axiomů predikátové logiky* vzniknou z těch ve výrokové logice prostým dosazením libovolných formulí predikátové logiky za výrokové proměnné. *Modus ponens* platí i v pred. logice. Další axiomy a pravidla:

- *schéma(axiom) specifikace*: $(\forall x)A \rightarrow A_x[t]$
- *schéma přeskoku*: $(\forall x)(A \rightarrow B) \rightarrow (A \rightarrow (\forall x)B)$, pokud proměnná x nemá volný výskyt v A.
- *pravidlo generalizace*: $\frac{A}{(\forall x)A}$

Toto je formální systém pred. logiky *bez rovnosti*. S rovností přibývá symbol $=$ a další tři axiomy.

Poznámka *(Vlastnosti formulí predikátové logiky)*

1. Je-li A' instance formule A, pak jestliže platí $\vdash A$, platí i $\vdash A'$. (*Věta o instancích*)
2. Je-li A' uzávěr formule A, pak $\vdash A$ platí právě když $\vdash A'$. (*Věta o uzávěru*)

Věta *(o dedukci v predikátové logice)*
Nechť T je množina formulí pred. logiky, A je uzavřená formule a B lib. formule, potom $T \vdash A \rightarrow B$ právě když $T, A \vdash B$.

Definice *(Teorie, model)*
Pro nějaký jazyk L prvního řádu je množina T formulí tohoto jazyka *teorie prvního řádu*. Formule z T jsou *speciální axiomy* teorie T. Pro interpretaci $\mathcal{M}$ jazyka L je $\mathcal{M}$ *model teorie* T ($\mathcal{M} \models T$), pokud jsou všechny speciální axiomy T pravdivé v $\mathcal{M}$. Formule A je *sémantickým důsledkem* T: $T \models A$, jestliže je pravdivá v každém modelu teorie T.

Věta *(o korektnosti)*
Je-li T teorie prvního řádu a A formule, potom platí:

1. Jestliže $T \vdash A$, potom $T \models A$.
2. Speciálně jestliže $\vdash A$, potom $\models A$.

Věta *(o úplnosti v predikátové logice)*
Nechť T je teorie s jazykem prvního řádu L. Je-li A lib. formule jazyka L, pak platí:

1. $T \vdash A$ právě když $T \models A$
2. T je bezesporná, právě když má model.

Definice *(Úplná teorie)*
Teorie T s jazykem L prvního řádu je *úplná*, je-li bezesporná a pro libovolnou uzavřenou formuli A je jedna z formulí $A, \neg A$ dokazatelná v T.

Věta *(o kompaktnosti)*
Teorie T s jazykem L prvního řádu má model, právě když každý její konečný fragment $T' \subset T$ má model.Tj. pro libovolnou formuli A jazyka L platí: $T \models A$ právě když $T' \models A$ pro nějaký konečný fragment $T' \subset T$.

0.3 Normální tvary výrokových formulí, prenexní tvary formulí predikátové logiky

Poznámka *(Vlastnosti log. spojek)*
Platí:

1. $A \wedge B \vdash A$, $A, B \vdash A \wedge B$
2. $A \leftrightarrow B \vdash A \rightarrow B$, $A \rightarrow B, B \rightarrow A \vdash A \leftrightarrow B$
3. $\wedge$ je idempotentní, komutativní a asociativní.
4. $\vdash (A_1 \rightarrow \dots (A_n \rightarrow B) \dots) \leftrightarrow ((A_1 \wedge \dots \wedge A_n) \rightarrow B)$
5. DeMorganovy zákony: $\vdash \neg(A \wedge B) \leftrightarrow (\neg A \vee \neg B)$, $\vdash \neg(A \vee B) \leftrightarrow (\neg A \wedge \neg B)$
6. $\vee$ je monotonní ($\vdash A \rightarrow A \vee B$), idempotentní, komutativní a asociativní.
7. $\vee$ a $\wedge$ jsou navzájem distributivní.

Věta *(o ekvivalenci ve výrokové logice)*
Jestliže jsou podformule $A_1 \dots A_n$ formule A ekvivalentní s $A'_1 \dots A'_n$ ($\vdash A'_i \leftrightarrow A_i$) a A' vytvořím nahrazením A'_i místo A_i, je i A ekvivalentní s A'. (Důkaz indukcí podle složitosti formule, rozborem případů A_i tvaru $\neg B$, $B \rightarrow C$)

Lemma *(o důkazu rozborem případů)*
Je-li T množina formulí a A, B, C formule, pak $T, (A \vee B) \vdash C$ platí právě když $T, A \vdash C$ a $T, B \vdash C$.

Definice *(Normální tvary)*
Výrokovou proměnnou nebo její negaci nazveme *literál*. *Klauzule* budiž disjunkce několika literálů. *Formule v normálním konjunktivním tvaru (CNF)* je konjunkce klauzulí. *Formule v disjunktivním tvaru (DNF)* je disjunkce konjunkcí literálů.

Věta *(o normálních tvarech)*
Pro každou formuli A lze sestrojit formule A_k, A_d v konjunktivním, resp. disjunktivním tvaru tak, že $\vdash A \leftrightarrow A_d$, $\vdash A \leftrightarrow A_k$. (Důkaz z DeMorganových formulí a distributivity, indukcí podle složitosti formule)

Prenexní tvary formulí predikátové logiky

Věta *(o ekvivalenci v predikátové logice)*
Nechť formule A' vznikne z A nahrazením některých výskytů podformulí $B_1, \dots, B_n$ po řadě formulemi $B'_1, \dots, B'_n$. Je-li $\vdash B_1 \leftrightarrow B'_1, \dots, \vdash B_n \leftrightarrow B'_n$, potom platí i $\vdash A \leftrightarrow A'$.

Definice *(Prenexní tvar)*
Formule predikátové logiky A je v *prenexním tvaru*, je-li $A \equiv (Q_1x_1)(Q_2x_2)\dots(Q_nx_n)B$, kde $n \geq 0$ a $\forall i \in \{1,\dots,n\}$ je $Q_1 \equiv \forall$ nebo $\exists$, B je otevřená formule a kvantifikátory Q_i jsou navzájem různé. B je *otevřené jádro* A, část s kvantifikátory je *prefix* A.

Definice *(Varianta formule predikátové logiky)*
Formule A' je *varianta* A, jestliže vznikla z A postupným nahrazením podformulí $(Qx)B$ (kde Q je $\forall$ nebo $\exists$) formulemi $(Qy)B_x[y]$ a y není volná v B. Podle *věty o variantách* je varianta s původní formulí ekvivalentní.

Lemma *(o prenexních operacích)*
Pro převod formulí do prenexního tvaru se používají tyto operace (výsledná formule je s původní ekvivalentní).Pro podformule B, C, kvantifikátor Q a proměnnou x:

1. podformuli lze nahradit nějakou její variantou
2. $\vdash \neg(Qx)B \leftrightarrow (\overline{Q}x)\neg B$
3. $\vdash (B \to (Qx)C) \leftrightarrow (Qx)(B \to C)$, pokud x není volná v B
4. $\vdash ((Qx)B \to C) \leftrightarrow (\overline{Q}x)(B \to C)$, pokud x není volná v C
5. $\vdash ((Qx)B\&C) \leftrightarrow (Qx)(B\&C)$, pokud x není volná v C
6. $\vdash ((Qx)B \vee C) \leftrightarrow (Qx)(B \vee C)$, pokud x není volná v C

Věta *(o prenexních tvarech)*
Ke každé formuli A predikátové logiky lze sestrojit ekvivalentní formuli A', která je v prenexním tvaru. (Důkaz: indukcí podle složitosti formule a z prenexních operací, někdy je nutné přejmenovat volné proměnné)

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

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
Jazyk L je posloupnost symbolů, tedy $L \subseteq X^*$, kde $\mathbf{X}^*$ je množina všech posloupností symbolů abecedy X. λ je prázdná posloupnost symbolů. *Rozšířená přechodová funkce* je $\delta^* : Q \to X^*$ - 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 $\forall u, v, w \in X^* : u \sim v \Rightarrow uw \sim vw$. Je *konečného indexu*, jestliže $X^*/\sim$ má konečný počet tříd.

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

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

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 ismorfismus 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.

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, levý (i pravý) kvocient ($K \setminus L = \{v|uv \in L, u \in K\}$) 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, právě když je rozpoznatelný konečným automatem.

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 $M = (Q, X, \delta, q_0, Z_0, F)$, kde proti regulárním automatům je Y abeceda pro symboly na zásobníku, Z_0 počáteční symbol na zásobníku a funkce instrukcí $\delta : Q \times (X \cup \{\lambda\}) \times Y \rightarrow \mathcal{P}(Q \times Y^*)$. Je z principu nedeterministický; vždy se nahrazuje vrchol zásobníku, nečte ale pokaždé vstupní symboly. Instrukci $(p, a, Z) \rightarrow (q, w)$ 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.

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.

Derivace, kterými lze v bezkontext. gramatice dostat nějaké terminální slovo, se liší jen pořadím použití pravidel, 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*, 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 $A \rightarrow au$, 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 $S \to \lambda|a_1Sa'_1|...|a_nSa'_n$. 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 $uqv| - wpz$ pro u část slova vlevo od akt. pozice na pásce, v od čteného písmena dál a q stav stroje. *Výpočet* je posloupnost kroků, slovo je w přijímáno, pokud $q_0w| -^* upv$, $p \in F$. 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 *(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.

Definice *(Generativní gramatika)*
Generativní gramatika je čtveřice $G = (V_N, V_T, S, P)$, kde V_N je množina neterminálních symbolů, V_T množina terminálních symbolů, S startovací symbol ($S \in V_N$) a P množina pravidel. *Jazyk generovaný gramatikou* je $L(G) = \{w \in V_T^*, S \Rightarrow^* V\}$.

Definice *(Chomského hierarchie)*
Chomského hierarchie je rozdělení gramatik do 4 tříd podle omezení na pravidla:

- *G0 (Rekurzivně spočetné jazyky)* mohou mít obecná pravidla.
- *G1 (Kontextové jazyky/gramatiky)* - jen pravidla tvaru $\alpha X\beta \to \alpha\omega\beta$, kde $X \in V_N$ a $\alpha, \beta, \omega \in (V_N \cup V_T)^*$, navíc $|\omega| > 0$. Může obsahovat i pravidlo $S \to \lambda$, ale pak se S nesmí vyskytovat na pravé straně žádného pravidla.
- *G2 (Bezkontextové jazyky/gramatiky)* - jen pravidla tvaru $X \to \omega$, kde $X \in V_N$, $\omega \in (V_N \cup V_T)^*$.
- *G3 (Regulární jazyky/pravé lineární gramatiky)* - jen pravidla typu $X \to \omega Y$ a $X \to \omega$, kde $\omega \in V_T^*$ a $X, Y \in V_N$.

Definuje uspořádání tříd jazyků podle inkluze, tedy $\mathcal{L}0 \supset \mathcal{L}1 \supset \mathcal{L}2 \supset \mathcal{L}3$.

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.

Poznámka *(Kontextové gramatiky)*
Separovaná gramatika je gramatika, obsahující pouze pravidla tvaru $\alpha \to \beta$, kde buď $\alpha, \beta \in V_N^*$, nebo $\alpha \in V_N$ a $\beta \in V_T \cup \{\lambda\}$. Ke každé kontextové gramatice lze sestrojit ekvivalentní separovanou. *Nevypouštějící (monotónní) gramatika* je taková, že pro každé pravidlo $u \to v$ platí $|u| \leq |v|$. 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 $|\delta(p, a, Z)| \leq 1$ a navíc pokud pro nějaké p, Z je $\delta(p, \lambda, Z) \neq \emptyset$, pak $\forall a \in X$ je $\delta(p, a, Z) = \emptyset$.

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 $q_0 w| -^* upv$ tak, že $p \in F$.

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 $q_0 lwr| -^* upv$, 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.