Matematická logika

přednáška pátá

Miroslav Kolařík

Zpracováno dle textu R. Bělohlávka:
Matematická logika – poznámky k přednáškám, 2004.

a dle učebního textu R. Bělohlávka a V. Vychodila:
Diskrétní matematika pro informatiky II, Olomouc 2006.

Obsah

Formulemi VL jsme formalizovali intuitivní pojem výrok a dovedli jsme jimi popsat skládání složitějších výroků z jednodušších pomocí logických spojek. Výroky, které byly dále nedělitelné, jsme označovali výrokovými symboly a vnitřní strukturou těchto výroků jsme se nezabývali. Naproti tomu predikátová logika (PL) formalizuje vztahy mezi individui neboli objekty, například jejich funkční závislosti, vlastnosti a vzájemné vztahy. Oproti VL se tedy na tvrzení díváme daleko jemnějším pohledem a formule PL to musí pochopitelně zohledňovat. Nyní si na příkladu ukážeme, co máme konkrétně na mysli pod pojmem "vztahy mezi individui".

Například tvrzení

"Pokud je x sudé číslo, pak je $x+1$ liché."

je z pohledu VL ve tvaru imlikace dvou výroků a je tudíž formalizovatelné výrokovou formulí $p \Rightarrow q$. Z pohledu PL se ale ve tvrzení vyskytují individua (čísla), jejich vlastnosti (být sudé, být liché) a funkční závislosti ($x+1$ je následníkem x, nebo podrobněji 1 označuje individuum a "$+$" označuje funkční závislost dvou individuí, v našem případě individuí označených x a 1).

Dalším typickým rysem je vytváření výroků kvantifikací, kterou ve slovním popisu vyjadřujeme obraty "každý", "nějaký", "právě jeden" a podobně. Například ve tvrzení

"Každý člověk má otce."

se vyskytuje kvantifikátor "každý". Kdybychom si toto tvrzení reformulovali poněkud kostrbatěji, mohlo by znít: "Pro každého člověka platí, že má otce." Vazbu "mít otce" bychom mohli chápat hned několika způsoby, například jako vlastnost (člověk *A* má otce), nebo třeba jako vztah dvou individuí (člověk *B* je otcem člověka *A*). Ve druhém případě je navíc ve tvrzení skryt další kvantifikátor: "ke každému člověku *A* existuje jeho otec *B*".

Přirozený jazyk je tedy základním prostředkem, pomocí kterého formulujeme a zaznamenáváme své usuzování. Již základní rozbor vět přirozeného jazyka odhalí některé jeho významné jednotky/součásti. Pro nás to budou: proměnné, relační symboly (symboly pro označování relací), funkční symboly (symboly pro označování funkcí) včetně symbolů pro označování konstant, symboly pro logické spojky, symboly pro kvantifikátory a pomocné symboly.

Ve větách "Každý slon je savec.", "Pro každé dva body existuje bod, který mezi nimi neleží", "Petr je mladší než Pavel nebo věk Jiřího je větší než součet věků Petra a Pavla.", "Součet druhých mocnin nenulových čísel je větší než nula." se mluví o jednomístných vztazích "být slonem", "být savcem", trojmístném vztahu "neležet mezi dvěma body", dvojmístném vztahu "být větší", o funkci přiřazující člověku jeho věk, o funkci sčítání, o funkci mocnění, o (konstantních) objektech Petr, Pavel, Jiří, nula, implicitně se zde objevují proměnné (např. první tvrzení, formulováno přesneji, říká "pro každé x platí, že je-li x slonem, je x savcem"), logické spojky ("nebo") a kvantifikátory ("pro každé", "existuje").

Stejně jako ve VL se budeme v PL soustředit na formu usuzovaného a budeme abstrahovat od obsahu sdělení.

Definice

Jazyk $\mathscr{J}$ **PL** obsahuje (a je tím určen)

- (**předmětové**) **proměnné** $x, y, z, \ldots, x_1, x_2, \ldots$
- **relační symboly** $p, q, r, \ldots, p_1, p_2 \ldots$, ke každému relačnímu symbolu r je dáno nezáporné celé číslo $\sigma(r)$ nazývané arita symbolu r; musí existovat alespoň jeden relační symbol
- **funkční symboly** $f, g, h, \ldots, f_1, f_2, \ldots$ ke každému funkčnímu symbolu f je dáno nezáporné celé číslo $\sigma(f)$ nazývané arita symbolu f
- **symboly pro logické spojky** $\neg$ (negace) a $\Rightarrow$ (implikace)
- **symbol pro univerzální kvantifikátor** $\forall$
- **pomocné symboly** – různé typy závorek a čárka.

Místo předmětové proměnné často říkáme jen proměnné. Předpokládáme, že proměnných je spočetně mnoho.

Množina všech relačních (někdy se říká predikátových) symbolů jazyka $\mathscr{J}$ se značí R; množina všech funkčních (někdy se říká operačních) symbolů jazyka $\mathscr{J}$ se značí F. Je-li $r \in R$ a $\sigma(r) = n$, pak říkáme, že r je **n-ární**. Podobně pro $f \in F$. Je-li $f \in F$ 0-ární, nazývá se f **symbol konstanty** (neboť funkce, která má 0 argumentů, musí přiřazovat vždy stejnou hodnotu, tj. je konstantní).

Je zřejmé, že jazyk je jednoznačně určen svými relačními symboly, funkčními symboly a jejich aritami (vše ostatní mají všechny jazyky stejné). Trojici $\langle R, F, \sigma \rangle$ proto nazýváme **typ jazyka**. (Pochopitelně předpokládáme, že $R \cap F = \emptyset$.)

Je-li mezi relačními symboly symbol $\approx$, nazýváme ho **symbol pro rovnost** a celý jazyk pak **jazyk s rovností**. (Symbol pro rovnost má specifické postavení, jak uvidíme dále.)

Termy PL

Základní syntaktické jednotky vybudované ze symbolů jazyka PL jsou termy a formule. Termy jsou výrazy reprezentující funkci aplikovanou na své operandy; formule reprezentují tvrzení o prvcích univerza.

Definice

Term jazyka typu $\langle R, F, \sigma \rangle$ je induktivně definován takto:

(i) každá proměnná x je term

(ii) je-li $f \in F$ n-ární a jsou-li $t_1, \ldots, t_n$ termy, pak $f(t_1, \ldots, t_n)$ je term.

Termy jsou tedy jisté konečné posloupnosti prvků daného jazyka. Je-li $f \in F$ binární, používáme také tzv. infixovou notaci a píšeme $x\, f\, y$ nebo $(x\, f\, y)$ místo $f(x, y)$, např. $2+3$ místo $+(2,3)$; ve složených termech používáme závorky, např. $(2+3)\cdot 5$.

Definice

Formule jazyka typu $\langle R, F, \sigma \rangle$ je induktivně definována takto:

(i) je-li $r \in R$ n-ární a jsou-li $t_1, \ldots, t_n$ termy, pak $r(t_1, \ldots, t_n)$ je formule

(ii) jsou-li φ a ψ formule, pak $\neg\varphi$, $(\varphi \Rightarrow \psi)$ jsou také formule

(iii) je-li φ formule a x proměnná, pak $(\forall x)\varphi$ je formule.

Formule vytvořené dle (i) se nazývají **atomické**. Je-li $r \in R$ binární, píšeme také $t_1\, r\, t_2$ nebo $(t_1\, r\, t_2)$ místo $r(t_1, t_2)$, tedy např. $x \leq y$ místo $\leq (x, y)$. Obzvláště píšeme $t_1 \approx t_2$ místo $\approx (t_1, t_2)$.

Budeme používat obvyklé konvence, zjednodušující čitelnost formulí: budeme vynechávat vnější závorky a budeme psát $(\forall x_1, \ldots, x_n)$ místo $(\forall x_1) \ldots (\forall x_n)$.

Stejně jako ve VL, symboly pro ostatní logické spojky ($\vee, \wedge, \Leftrightarrow$) nepatří do jazyka PL. Podobně existenční kvantifikátor ($\exists$) nepatří do jazyka PL. Abychom tyto symboly měli k dispozici, chápeme

posloupnost $\varphi \wedge \psi$ jako zkratku za $\neg(\varphi \Rightarrow \neg\psi)$,

posloupnost $\varphi \vee \psi$ jako zkratku za $\neg\varphi \Rightarrow \psi$,

posloupnost $\varphi \Leftrightarrow \psi$ jako zkratku za $(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$,

posloupnost $(\exists x)\varphi$ jako zkratku za $\neg(\forall x)\neg\varphi$.

Posloupnosti obsahující symboly $\wedge, \vee, \Leftrightarrow$ a $\exists$ tedy nejsou formulemi. Pro jednoduchost však vědomi si toho, že a jakým způsobem se dopouštíme nepřesnosti, budeme těmto posloupnostem také říkat formule.

Příklad

Uvažujme jazyk $\mathscr{J}$ typu $\langle R, F, \sigma \rangle$, kde $R = \{p, d, b, m\}$, $F = \emptyset$, relační symboly p, d, b jsou unární, m binární. Je to jazyk bez funkčních symbolů, a tedy jedinými termy jsou proměnné. Atomickými formulemi jsou $p(x), p(y), d(y), b(x)$ atd. Dalšími formulemi (ne atomickými) jsou např. výrazy $p(x) \vee p(x)$, $p(x) \vee \neg p(x)$, $(\forall x)((p(x) \wedge \neg d(x)) \Rightarrow b(x))$, $(\exists x) b(x) \wedge \neg p(x)$, $(\forall x, y)(b(x) \Rightarrow m(x, y)) \Rightarrow b(y)$. Výrazy $((x \Rightarrow$, $p(x, x)$, $p(d(x))$, $(\forall x)(m(x, y) \Rightarrow\Rightarrow p(x))$ formulemi nejsou.

Příklad

Uvažujme jazyk $\mathscr{J}$ typu $\langle R, F, \sigma \rangle$, kde $R = \{p, \leq\}$, $F = \{c, \circ\}$, c je nulární (tj. symbol konstanty), p je unární, $\leq, \circ$ jsou binární. Pak termy jsou např. výrazy $c, x, c \circ c, c \circ (x \circ y)$. Výrazy $cc, c \circ p(x), p(x), c \circ \circ x$ termy nejsou. Formulemi jsou např. $c \leq x$, $p(x) \Rightarrow (c \leq x)$, $(\forall x)(x \leq x \circ x)$, $(\forall x, y)(x \circ y \leq y \circ x)$.

Poznámka: Jazyk PL obsahuje symboly různých typů (relační symboly, funkční symboly, symboly spojek). Z nich se dají vytvářet termy a formule, které jsou v určitém smyslu "rozumnými řetězci". Rozumné proto, že dáme-li relačním a funkčním symbolům smysl, dají se "rozumně" číst. (Přesný smysl dostanou relační a funkční symboly, a také termy a formule, až vybudujeme sémantiku.) Tak například uvažujme jazyk z 1. příkladu předchozího slidu. Nechť p, d, b, m označují po řadě "mít dostatečný příjem", "mít velké dluhy", "být bonitní", "být manželé", tj. $p(x)$ znamená "objekt označený x má dostatečný příjem" atd. Formule $(\forall x)(p(x) \wedge \neg d(x) \Rightarrow b(x))$ pak "říká": "pro každé x platí, že má-li x dostatečný příjem a nemá-li velké dluhy, pak je x bonitní". Formule $(\forall x, y)(b(x) \Rightarrow (m(x, y) \Rightarrow b(y)))$ "říká": "pro každé x a y platí, že je-li x bonitní a jsou-li x a y manželé, je i y bonitní".

Můžeme také postupovat obráceně: K dané větě přirozeného jazyka navrhneme jazyk PL a napíšeme formuli, která odpovídá danému tvrzení. Viz přednášky a cvičení.

Podobně jako ve VL můžeme i v PL provádět důkazy strukturální indukcí.

Věta – důkaz strukturální indukcí pro termy

Nechť $\mathscr{V}$ je vlastnost termů. Nechť platí, že

- každá proměnná má vlastnost $\mathscr{V}$
- mají-li termy $t_1, \ldots, t_n$ vlastnost $\mathscr{V}$ a je-li $f \in F$ n-ární, pak vlastnost $\mathscr{V}$ má i term $f(t_1, \ldots, t_n)$.

Pak vlastnost $\mathscr{V}$ má každý term.

Věta – důkaz strukturální indukcí pro formule

Nechť $\mathscr{V}$ je vlastnost formulí. Nechť platí, že

- každá atomická formule má vlastnost $\mathscr{V}$
- mají-li formule φ a ψ vlastnost $\mathscr{V}$, pak vlastnost $\mathscr{V}$ mají i formule $\neg\varphi$ a $(\varphi \Rightarrow \psi)$
- má-li formule φ vlastnost $\mathscr{V}$, pak vlastnost $\mathscr{V}$ má i formule $(\forall x)\varphi$ s proměnnou x.

Pak vlastnost $\mathscr{V}$ má každá formule PL.

Termy a formule jsou definovány induktivně. Každý term použitý při konstrukci termu t se nazývá **podterm** termu t. Každá formule použitá v konstrukci formule φ se nazývá **podformule** formule φ. Přesněji:

Množina $\mathrm{sub}(t)$ všech podtermů termu t je definována následovně: pro proměnnou x je $\mathrm{sub}(x) = \{x\}$;
$\mathrm{sub}(f(t_1, \ldots, t_n)) = \{f(t_1, \ldots, t_n)\} \cup \mathrm{sub}(t_1) \cup \cdots \cup \mathrm{sub}(t_n)$.

Množina $\mathrm{sub}(\varphi)$ všech podformulí formule φ je definována následovně: je-li φ atomická, pak $\mathrm{sub}(\varphi) = \{\varphi\}$;
$\mathrm{sub}(\neg\varphi) = \{\neg\varphi\} \cup \mathrm{sub}(\varphi)$;
$\mathrm{sub}(\varphi \Rightarrow \psi) = \{\varphi \Rightarrow \psi\} \cup \mathrm{sub}(\varphi) \cup \mathrm{sub}(\psi)$;
$\mathrm{sub}((\forall x)\varphi) = \{(\forall x)\varphi\} \cup \mathrm{sub}(\varphi)$.

Viz přednášky a cvičení.

Říkáme, že proměnná **se vyskytuje** v termu nebo ve formuli, jestliže je některým symbolem termu nebo formule jakožto řetězce symbolů s tou výjimkou, že výskyty za $\forall$ se nepočítají (tj. proměnná x nemá výskyt ve formuli $(\forall x)r(y,z)$). Množinu všech proměnných, které se vyskytují v termu t označujeme $\mathrm{var}(t)$; množinu všech proměnných, které se vyskytují ve formuli φ označujeme $\mathrm{var}(\varphi)$.

Je-li t term, píšeme $t(x_1,\ldots,x_n)$, abychom zdůraznili, že všechny proměnné, které se v t vyskytují, se nacházejí mezi $x_1,\ldots,x_n$.

Proměnné z $\mathrm{var}(\varphi)$ mohou mít ve formuli φ výskyt dvojího druhu – volný a vázaný. Viz přednášky a cvičení.

Množina $\mathrm{free}(\varphi)$ proměnných, které mají ve φ **volný výskyt** je definována následovně: je-li φ atomická, pak $\mathrm{free}(\varphi) = \mathrm{var}(\varphi)$; dále $\mathrm{free}(\neg\varphi) = \mathrm{free}(\varphi)$; $\mathrm{free}(\varphi \Rightarrow \psi) = \mathrm{free}(\varphi) \cup \mathrm{free}(\psi)$; a $\mathrm{free}((\forall x)\varphi) = \mathrm{free}(\varphi) - \{x\}$.

Množina $\mathrm{bound}(\varphi)$ proměnných, které mají ve φ **vázaný výskyt** je definována následovně: je-li φ atomická, pak $\mathrm{bound}(\varphi) = \emptyset$; dále $\mathrm{bound}(\neg\varphi) = \mathrm{bound}(\varphi)$;
$\mathrm{bound}(\varphi \Rightarrow \psi) = \mathrm{bound}(\varphi) \cup \mathrm{bound}(\psi)$; a
$\mathrm{bound}((\forall x)\varphi) = \mathrm{bound}(\varphi) \cup \{x\}$.

Píšeme $\varphi(x_1, \ldots, x_n)$, abychom zdůraznili, že všechny proměnné, které mají ve φ volný výskyt, se nacházejí mezi $x_1, \ldots, x_n$, tj. $\mathrm{free}(\varphi) \subseteq \{x_1, \ldots, x_n\}$.

Poznámka: Chceme-li napsat formuli nebo term, musíme nejdříve popsat jazyk, jehož formuli nebo term chceme napsat. To je však často pracné a zbytečné. Proto zavedeme pojem indukovaného jazyka. **Jazyk indukovaný** množinami $\mathscr{F}$ a $\mathscr{T}$ řetězců je nejmenší jazyk $\mathscr{J}$ typu $\langle R, F, \sigma \rangle$, v němž řetězce z $\mathscr{F}$ jsou formulemi a řetězce z $\mathscr{T}$ jsou termy (tj. je-li $\mathscr{J}'$ jiný takový jazyk typu $\langle R', F', \sigma' \rangle$, pak $R \subseteq R'$, $F \subseteq F'$). Místo jazyk indukovaný množinami $\mathscr{F}$ a $\mathscr{T}$ řetězců také říkáme jazyk určený formulemi z $\mathscr{F}$ a termy z $\mathscr{T}$.

Příklad

Uvažujme jazyk určený termy $t_1 = x_1 + ((c + x_1) + x_2)$, $t_2 = (c + c) + c$ a formulemi $\varphi_1 = (\forall x, y)((r(x) \wedge (x \leq y)) \Rightarrow r(y))$, $\varphi_2 = (c \leq x) \wedge (\forall x)(\exists y)(x + x \leq x + y)$. Zřejmě $R = \{r, \leq\}$, $F = \{c, +\}$, kde $\sigma(r) = 1$, $\sigma(\leq) = 2$, $\sigma(c) = 0$ a $\sigma(+) = 2$. Dále pak $\mathrm{var}(\varphi_1) = \{x, y\}$, $\mathrm{var}(\varphi_2) = \{x, y\}$, $\mathrm{free}(\varphi_1) = \emptyset$, $\mathrm{bound}(\varphi_1) = \{x, y\}$, $\mathrm{free}(\varphi_2) = \{x\}$, $\mathrm{bound}(\varphi_2) = \{x, y\}$ (proměnná x má ve formuli φ_2 volný i vázaný výskyt).

Užitečným pojmem je pojem substituce termu za proměnnou.

Definice

Nechť t a s jsou termy, x proměnná. **Výsledek substituce termu s za x v t** je term $t(x/s)$ definovaný následovně:

(i) je-li t proměnná, pak

$$t(x/s) = \begin{cases} \text{s} & \text{pro } t = x \\ \text{t} & \text{pro } t \neq x \end{cases}$$

(ii) pro $t = f(t_1, \ldots, t_n)$, kde $f \in F$ je n-ární a $t_1, \ldots, t_n$ jsou termy, je $t(x/s) = f(t_1(x/s), \ldots, t_n(x/s))$.

Definice

Pro formuli φ, term s a proměnnou x je **výsledkem substituce termu s za x ve** φ formule $\varphi(x/s)$ definovaná následovně:

(i) pro $\varphi = r(t_1, \ldots, t_n)$, kde $r \in R$ je n-ární a $t_1, \ldots, t_n$ jsou termy, je $\varphi(x/s) = r(t_1(x/s), \ldots, t_n(x/s))$

(ii) $(\neg\varphi)(x/s) = \neg(\varphi(x/s))$, $(\varphi \Rightarrow \psi)(x/s) = \varphi(x/s) \Rightarrow \psi(x/s)$

(iii) $((\forall y)\varphi)(x/s) = (\forall y)\varphi$ pro $y = x$,
$((\forall y)\varphi)(x/s) = (\forall y)(\varphi(x/s))$ pro $y \neq x$.

Předchozí definice lze snadno rozšířit na definici substituce termu za term.

Substituce termu za proměnnou může vést k nechtěným situacím. Uvažujme např. formuli $\varphi = (\forall y)(y \leq x)$, která vyjadřuje, že x je větší než jakékoliv y. Substituce $y + y$ za x vede k $\varphi(x/y+y) = (\forall y)(y \leq y + y)$, což je jistě formule, která vyjadřuje něco jiného. Abychom zabránili takovým případům, definujeme tzv. korektní substituce.

Definice

Substituce termu t za proměnnou x ve formuli φ je **korektní** (t se nazývá **substituovatelný** za x ve φ), jestliže pro každou $y \in \mathrm{var}(t)$ platí, že žádná podformule formule φ, která je ve tvaru $(\forall y)\psi$, neobsahuje výskyt x, který je volným výskytem x ve φ.

Viz cvičení.