Kapitola 1

Množiny

1.1 Základní množinové pojmy

Pojem množiny nedefinujeme, pouze připomínáme, že množina je „souhrn, nebo soubor“ navzájem rozlišitelných objektů, kterým říkáme prvky.

1.1.1 Princip rovnosti. Dvě množiny S a T jsou si rovny (píšeme $S = T$) právě tehdy, když každý prvek množiny S je prvkem množiny T a naopak každý prvek T je také prvkem S.

1.1.2 Zadání množiny. Množinu můžeme zadat buď výčtem, tj. vypíšeme všechny její prvky, nebo naznačíme, které prvky obsahuje. Příkladem je např. množina sudých přirozených čísel $S = \{0, 2, 4, 6, 8, \ldots\}$. Množinu zadáváme vlastností, která charakterizuje její prvky. Je-li $p(x)$ vlastnost, kterou prvek má nebo nemá, pak množinu C všech prvků x s vlastností $p(x)$ a žádných jiných zapisujeme

$$C = \{x \mid p(x)\}.$$

Množina všech sudých přirozených čísel je množina

$$\{m \mid m = 2k, k \text{ je přirozené číslo}\}.$$

Množina všech lichých přirozených čísel je množina

$$\{m \mid m = 2k + 1, k \text{ je přirozené číslo}\}.$$

1.1.3 Podmnožiny. Mějme dvě množiny S a T. Jestliže každý prvek množiny S je také prvkem množiny T, říkáme, že S je *podmnožina* T a píšeme $S \subseteq T$.

Jestliže platí $S \subseteq T$ a S a T jsou různé množiny, říkáme též, že S je *vlastní podmnožina* množiny T.

1.1.4 Tvrzení. $S = T$ právě tehdy, když $S \subseteq T$ a současně $T \subseteq S$.

1.1.5 Prázdná množina. *Prázdná množina* je množina, která nemá žádný prvek; značíme ji $\emptyset$.

$\emptyset \subseteq A$ pro každou množinu A.

1.1.6 Operace s množinami. Mějme dvě množiny A a B. Jejich *sjednocením* je množina

$$A \cup B = \{x \mid x \in A \text{ nebo } x \in B\};$$

průnikem těchto dvou množin je množina

$$A \cap B = \{x \mid x \in A \text{ a } x \in B\}.$$

Rozdílem množin A a B (v tomto pořadí) je množina

$$A \setminus B = \{x \mid x \in A \text{ a } x \notin B\}.$$

Je-li $A \subseteq U$, potom *doplňkem množiny A v množině U* je množina $U \setminus A$.

1.1.7 Disjunktní množiny. Je-li $A \cap B = \emptyset$, říkáme, že množiny A a B jsou *disjunktní*.

1.1.8 Kartézský součin. *Kartézský součin* množin A, B (značíme $A \times B$) je definován

$$A \times B = \{(a, b) \mid a \in A, b \in B\}.$$

Jestliže se jedná o kartézský součin stejných množin, mluvíme o *kartézských mocninách* množiny A a píšeme A^2 místo $A \times A$, A^3 místo $A \times (A \times A)$, atd.

1.1.9 Potenční množina. *Potenční množina* $P(A)$ množiny A je rovna množině všech podmnožin množiny A; formálně

$$P(A) = \{B \mid B \subseteq A\}.$$

Potenční množina je vždy neprázdná; obsahuje totiž prázdnou množinu.

1.1.10 Charakteristická funkce podmnožiny. *Charakteristická funkce* χ_A podmožiny $A \subseteq U$ je zobrazení $\chi_A : U \to \{0, 1\}$ definované

$$\chi_A(x) = \begin{cases} 1, & x \in A; \\ 0, & x \in U \setminus A. \end{cases}$$

1.2 Russellův paradox

Russellův paradox. Uvažujme vlastnost „nebýti prvkem sebe sama“. Tuto vlastnost má řada množin: Např. množina $A = \{2\}$ není prvkem sebe sama, protože množina A má jediný prvek a to 2. Jistě byste našli řadu jiných množin, které nejsou prvkem sebe sama. Utvořme tedy tuto množinu:

$$R = \{x \mid x \notin x\}.$$

Podle principu abstrakce se jedná o dobře utvořenou množinu (nebýti prvkem sebe sama jako množiny je vlastnost $\mathcal{K}$). Nyní se můžeme ptát, zda množina R je prvkem sebe sama nebo ne. Jsou pouze dvě možnosti:

- $R \in R$; ale v tomto případě R musí splňovat vlastnost $\mathcal{K}$, a tedy $R \notin R$. Tedy má současně platit $R \in R$ a $R \notin R$ a to nastat nemůže.
- $R \notin R$; v tomto případě R nesplňuje vlastnost $\mathcal{K}$, a tedy není pravda, že $R \notin R$, tj. $R \in R$. Opět má současně platit $R \in R$ a $R \notin R$, takže ani tato situace nastat nemůže.

Chyba spočívá v tom, že jsme předpokládali, že R je množina.

1.2.1 Princip vydělení. Mějme vlastnost $\mathcal{K}$, kterou každý prvek má nebo nemá. Pak pro každou množinu U existuje množina skládající se právě ze všech prvků množiny U splňujících vlastnost $\mathcal{K}$. Tuto množinu zapisujeme $\{x \mid x \in \in U, x \text{ má vlastnost } \mathcal{K}\}$, nebo kratčeji $\{x \in U \mid \mathcal{K}\}$.

1.2.2 Uvědomte si, že existence množiny vytvořené podle tohoto principu

$$R = \{x \mid x \in U, x \notin x\}$$

již nevede ke sporu. Na otázku, zda R je prvkem sebe sama můžeme dát tuto odpověď:

- $R \notin R$; to znamená, že není pravda tvrzení $R \in U$ a současně $R \notin R$. Tedy buď není pravda $R \in U$ nebo není pravda $R \notin R$. Protože $R \notin R$, dostáváme, že $R \notin U$. Tato situace nastat může.

1.3 Mohutnost množin

1.3.1 Vzájemně jednoznačné zobrazení. Zobrazení f množiny A do množiny B je *vzájemně jednoznačné* právě tehdy, když je prosté a na.

Prosté zobrazení je takové, které dvěma různým prvkům x, y množiny A přiřazuje různé prvky $f(x)$, $f(y)$ množiny B.

Zobrazení je na B, jestliže pro každý prvek $y \in B$ existuje prvek $x \in A$ takový, že $f(x) = y$.

1.3.2 Mohutnost množin. Řekneme, že dvě množiny A, B mají *stejnou mohutnost*, jestliže existuje vzájemně jednoznačné zobrazení množiny A na množinu B. Tento fakt značíme $|A| = |B|$.

1.3.3 Poznámka. Poznamenejme, že existuje-li vzájemně jednoznačné zobrazení f množiny A na množinu B, pak také existuje vzájemně jednoznačné zobrazení množiny B na množinu A — inversní zobrazení k zobrazení f.

1.3.4 Příklad. Množina všech sudých čísel $S = \{0, 2, 4, \ldots\} = \{2n \mid n \in \mathbb{N}\}$ a množina všech lichých čísel $L = \{1, 3, 5, \ldots\} = \{2n + 1 \mid n \in \mathbb{N}\}$ mají stejnou mohutnost.

Definujme zobrazení $f\colon S \to L$ předpisem:

$$f(2n) = 2n + 1 \text{ pro všechna } n \in \mathbb{N}.$$

Toto zobrazení je prosté, protože z rovnosti $f(2n) = f(2m)$ pro dvě přirozená čísla n, m vyplývá $2n+1 = 2m+1$ a tedy $n = m$. Z popisu množiny L vyplývá, že zobrazení f je také na: Ano, každé liché přirozené číslo je tvaru $2m + 1$ a je tedy obrazem sudého čísla $2m$.

1.3.5 Spočetné a nespočetné množiny. Řekneme, že množina A je *spočetná*, má-li stejnou mohutnost jako množina všech přirozených čísel $\mathbb{N}$. Jestliže množina A je nekonečná a není spočetná, řekneme, že je *nespočetná*.

Pro množinu, která je buď spočetná nebo konečná, se často používá termín *nejvýše spočetná množina*.

1.3.6 Tvrzení. Množina A je spočetná právě tehdy, když ji lze uspořádat do prosté nekonečné posloupnosti (tj. neopakují se v ní prvky).

1.3.7 Příklad. Množina všech celých čísel je spočetná.

Množinu celých čísel uspořádáme do nekonečné prosté posloupnosti takto:

$$0, 1, -1, 2, -2, 3, -3, 4, -4, \ldots, n, -n, \ldots$$

Přesněji číslo 0 je na 0-tém místě (tj. je to prvek a_0), číslo 1 je na prvním místě (prvek a_1), číslo -1 je na druhém místě (prvek a_2), číslo 2 je na místě $2 \cdot 2 - 1 = 3$ (prvek a_3), číslo -2 je na místě $2 \cdot 2 = 4$ (prvek a_4), atd. Obecně: celé kladné číslo n je na místě $2n - 1$ (prvek a_{2n-1}) a číslo $-n$ je na místě $2n$ (prvek a_{2n}).

Takto jsme uspořádali množinu $\mathbb{Z}$ do prosté nekonečné posloupnosti, a tedy je spočetná.

1.3.8 Tvrzení. Nekonečná podmnožina spočetné množiny je opět spočetná množina.

Tedy např. množina všech kladných přirozených čísel je spočetná.

1.3.9 Tvrzení. Sjednocení dvou nejvýše spočetných množin je nejvýše spočetná množina. Sjednocení dvou množin z nichž jedna je spočetná a druhá je nejvýše spočetná je spočetná množina.

1.3.10 Tvrzení. Kartézský součin dvou nejvýše spočetných množin je nejvýše spočetná množina.

1.3.11 Ukážeme, že množinu $C = A \times B$, kde A a B jsou spočetné množiny ($A = \{a_0, a_1, a_2, \ldots\}$ a $B = \{b_0, b_1, b_2, \ldots\}$), lze uspořádat do prosté posloupnosti podle následujícího schematu:

$$\begin{array}{ccccc}
(a_0,b_0) & & (a_0,b_1) & & (a_0,b_2) \quad \ldots \\
& \swarrow & & \swarrow & \\
(a_1,b_0) & & (a_1,b_1) & & (a_1,b_2) \quad \ldots \\
& \swarrow & & \swarrow & \\
(a_2,b_0) & & (a_2,b_1) & & (a_2,b_2) \quad \ldots \\
& \swarrow & & \swarrow & \\
\vdots & & \vdots & & \vdots
\end{array}$$

Na schematu je naznačeno, jak množinu C uspořádat. Máme

$$C = \{(a_0,b_0), (a_0,b_1), (a_1,b_0), (a_0,b_2), (a_1,b_1), (a_2,b_0), (a_0,b_3), \ldots\}.$$

Přesný popis je tento: dvojice (a_i, b_j) bude v posloupnosti na místě $k = i + \\ + \frac{(i+j)(i+j+1)}{2}$.

1.3.12 Příklad. Množina $\mathbb{Q}$ všech racionálních čísel je spočetná.

Každé racionální číslo lze reprezentovat jako zlomek $\frac{p}{q}$, kde q je nenulové přirozené číslo a p je celé číslo. Tedy zlomky můžeme chápat jako uspořádané dvojice (p, q), kde p je čitatel a q jmenovatel racionálního čísla $\frac{p}{q}$. Navíc množina celých čísel je spočetná, stejně jako množina všech nenulových přirozených čísel. Proto množina M všech dvojic (p, q) je spočetná. Množina racionálních čísel $\mathbb{Q}$ je nyní nekonečná podmnožina množiny M která obsahuje pouze ty dvojice (p, q), kde p a q jsou nesoudělné. Proto je množina $\mathbb{Q}$ spočetná.

1.3.13 Tvrzení. Sjednocení spočetně mnoha nejvýše spočetných množin je nejvýše spočetná množina.

Jinak řečeno: Jsou-li $A_0, A_1, \dots A_n, \dots$ nejvýše spočetné množiny, pak jejich sjednocení $A_0 \cup A_1 \cup \dots = \bigcup_{i\in\mathbb{N}} A_i$ je nejvýše spočetná množina.

Pro zdůvodnění je možné použít stejného schematu jako pro kartézský součin.

1.3.14 Příklad. Mějme neprázdnou množinu A, která je nejvýše spočetná. Množina $A^\star$ všech konečných posloupností prvků z A, je spočetná.

Množinu všech konečných posloupností rozdělíme do množin A_i, $i \in \mathbb{N}$, tak, že množina A_i obsahuje přesně všechny konečné posloupnosti délky i. Pak platí:

$$A^\star = \bigcup_{i\in\mathbb{N}} A_i.$$

Přitom každá z množin A_i je nejvýše spočetná. Je tedy $A^\star$ spočetné sjednocení neprázdných nejvýše spočetných množin A_i splňujících

$$A_i \cap A_j = \emptyset \quad \text{pro různá} \quad i, j \in \mathbb{N}.$$

Proto je $A^\star$ spočetná množina.

1.3.15 Věta. Množina všech podmnožin množiny přirozených čísel $\mathbb{N}$ není spočetná.

1.3.16 Cantorova diagonální metoda. Každou podmnožinu množiny přirozených čísel $\mathbb{N}$ si můžeme představit jako její charakteristickou funkci, tj. jako nekonečnou posloupnost nul a jedniček. Cantorova diagonální metoda ukazuje (sporem), že množina všech nekonečných posloupností nul a jedniček je nespočetná.

Předpokládejme, že množina všech nekonečných posloupností nul a jedniček je spočetná, tudíž ji můžeme uspořádat do prosté nekonečné posloupnosti. Znázorněme si to schematicky — v prvním řádku máme posloupnost s_0, v druhém řádku posloupnost s_1, ve třetím řádku posloupnost s_2, atd.

$$\begin{array}{lcccccccc}
s_0 & = & \boxed{s_0(0),} & s_0(1), & s_0(2), & s_0(3), & s_0(4), & s_0(5), & \dots \\
s_1 & = & s_1(0), & \boxed{s_1(1),} & s_1(2), & s_1(3), & s_1(4), & s_1(5), & \dots \\
s_2 & = & s_2(0), & s_2(1), & \boxed{s_2(2),} & s_2(3), & s_2(4), & s_2(5), & \dots \\
s_3 & = & s_3(0), & s_3(1), & s_3(2), & \boxed{s_3(3),} & s_3(4), & s_3(5), & \dots \\
s_4 & = & s_4(0), & s_4(1), & s_4(2), & s_4(3), & \boxed{s_4(4),} & s_4(5), & \dots \\
s_5 & = & s_5(0), & s_5(1), & s_5(2), & s_5(3), & s_5(4), & \boxed{s_5(5),} & \dots \\
\vdots & & & & & & & &
\end{array}$$

Vytvoříme novou posloupnost nul a jedniček a ukážeme o ní, že nebyla vypsána. Je to posloupnost $\overline{s}$ definovaná takto: Bylo-li v prvním rámečku schematu číslo 1, začíná $\overline{s}$ číslem 0; bylo-li v prvním rámečku číslo 0, začíná posloupnost číslem 1. Jinými slovy, posloupnost $\overline{s}$ má na nultém místě to druhé z čísel 0 a 1, než které má posloupnost s_0. Dále postupujeme obdobně: Jestliže v druhém rámečku má posloupnost s_1 číslo 0, položíme $\overline{s}(1)$ rovno 1; je-li $s_1(1) = = 1$, položíme $\overline{s}(1) = 0$. Hodnota $\overline{s}(2)$ bude číslo $1 - s_2(2)$, atd.

Formálně zápis vypadá takto: $\overline{s} = \{\overline{s}(0), \overline{s}(1), \overline{s}(2), \ldots, \overline{s}(n), \ldots\}$, kde $\overline{s}(n) = 1 - s_n(n)$ pro všechna $n \in \mathbb{N}$.

Posloupnost $\overline{s}$ mezi posloupnostmi $s_0, s_1, s_2, \ldots, s_n, \ldots$ není, neboť od posloupnosti s_0 se liší na nultém místě, od posloupnosti s_1 se liší na prvním místě, od posloupnosti s_2 se liší na druhém místě, ... od n-té posloupnosti s_n se liší na n-tém místě. Dospěli jsme ke sporu s předpokladem, že jsme na začátku vypsali všechny posloupnosti. Tedy množina všech nekonečných posloupností nul a jedniček není spočetná.

1.3.17 Poznámka. Obdobně jako jsme ukázali, že množina všech podmnožin množiny přirozených čísel je nespočetna, je možné ukázat, že množina všech reálných čísel v otevřeném intervalu $(0, 1)$ je také nespočetná.

1.3.18 Věta. Pro žádnou množinu S neexistuje zobrazení $f: S \to P(S)$, které je na $P(S)$.

1.3.19 Zdůvodnění. Předpokládejme, že f je zobrazení množiny S do $P(S)$. Definujme

$$C = \{x \in S \mid x \notin f(x)\}.$$

Uvědomte si, že $f(x) \in P(S)$, tj. $f(x) \subseteq S$.

Tím jsme definovali podmnožinu C množiny S. Ukážeme, že C není f-obrazem žádného prvku $y \in S$ [tj. $C \neq f(y)$ pro žádné $y \in S$]. Tím bude ukázáno, že zobrazení f není na.

Pro každý prvek $a \in S$ platí buď $a \in C$ nebo $a \notin C$. Jestliže $a \in C$, pak podle definice množiny C platí $a \notin f(a)$. Tudíž $C \neq f(a)$. Jestliže $a \notin C$, pak podle definice množiny C platí $a \in f(a)$. Tudíž i v tomto případě $C \neq f(a)$. Tedy množina C není rovna žádnému obrazu $f(a)$. Ukázali jsme, že zobrazení f není na celé $P(S)$.

Protože f bylo libovolné zobrazení S do $P(S)$, ukázali jsme, že neexistuje zobrazení množiny S na $P(S)$.

1.3.20 Poznámka. Ukázali jsme, že pro žádnou množinu S neexistuje zobrazení množiny S *na* množinu všech jejích podmnožin $P(S)$. To intuitivně říká, že S „má méně prvků“ než $P(S)$. Povšimněte si přitom podobnosti předchozí úvahy s Russellovým paradoxem. (Cantor ovšem takto uvažoval více než 10 let před Russellem!)

Kapitola 2

Relace

2.1 Binární relace z množiny do množiny

2.1.1 Definice. *Relace* (přesněji *binární relace*) *z množiny A do množiny B* je libovolná množina uspořádaných dvojic $R \subseteq A \times B$. Jestliže $A = B$, mluvíme o *relaci na množině A*.

2.1.2 Příklady.

1. Být dědečkem. Jedná se o relaci R na množině A všech lidí. Dvojice (a, b) patří do R právě tehdy, když osoba a je dědečkem osoby b.

2. Být stejně dlouhé. Jedná se o relaci na množině všech objektů (tady se musíte rozhodnout, které objekty chcete uvažovat); pro dva objekty a, b platí $(a, b) \in R$ právě tehdy, když oba objekty jsou stejně dlouhé.

3. Být podmnožinou. Jedná o relaci R na množině všech podmnožin množiny U. Pro dvě množiny X, Y, $X \subseteq U$, $Y \subseteq U$ platí: (X, Y) je v relaci R právě tehdy, když množina X je podmnožinou množiny Y.

4. Být větší nebo rovno. Jedná se např. o relaci R na množině všech přirozených čísel $\mathbb{N}$, kde $(m, n) \in R$ právě tehdy, když $m \leq n$.

5. Být studentem studijní skupiny. Jedná se o relaci R z množiny A všech studentů prvého ročníku (např. FEL) do množiny B všech studijních skupin. Dvojice (a, K), kde $a \in A$ a K je studijní skupina, patří do relace R právě tehdy, když je student a zapsán do skupiny K.

6. Funkce sinus. Jedná o relaci R na množině reálných čísel $\mathbb{R}$ definovanou: $(x, y) \in R$ právě tehdy, když $y = \sin x$.

2.1.3 Konvence. Zápis $(a, b) \in R$ je často nepříliš šťastný; nikoho by nenapadlo psát $(X, Y) \in \subseteq$, či dokonce $(2, 3) \in \leq$. Proto v dalším budeme místo zápisu $(a, b) \in R$ psát $a \; R \; b$.

2.1.4 Každé zobrazení $f : A \to B$ je relace (nebo přesněji definuje relaci); a to relace f z A do B definovaná $x\ f\ y$ právě tehdy, když $y = f(x)$. Ne každá relace z A do B je zobrazením množiny A do množiny B; k tomu, aby relace R byla zobrazením je třeba (a stačí), aby pro každé $a \in A$ existovalo právě jedno $b \in B$ takové, že $a\ R\ b$.

Pro relace přejímáme také termíny, které jsou běžné, když mluvíme o zobrazení. *Definičním oborem* relace R je množina všech $a \in A$, pro něž existuje $b \in B$ takové, že $a\ R\ b$; *oborem hodnot* relace R je množina všech $b \in B$, pro něž existuje $a \in A$ takové, že $a\ R\ b$.

2.1.5 Poznámka. Jestliže obě množiny A a B jsou konečné, pak relaci $R \subseteq$ $\subseteq A \times B$ můžeme representovat maticí takto:

Označme $A = \{a_1, a_2, \ldots, a_n\}$ a $B = \{b_1, b_2, \ldots, b_k\}$. Položme

$$M_R = (m_R(i,j))_{i=1,\ldots,n}^{j=1,\ldots,k},$$

kde $m_R(i,j) = 1$ pro $(a_i, b_j) \in R$ a $m_R(i,j) = 0$ pro $(a_i, b_j) \notin R$.

Uvědomte si, že se vlastně jedná o charakteristickou funkci relace R jakožto podmnožiny množiny všech uspořádaných dvojic $A \times B$.

2.2 Operace s relacemi

2.2.1 Podrelace. Řekneme, že relace R je *podrelací* relace S, jestliže $R \subseteq S$; tj. je-li $a\ R\ b$, pak také platí $a\ S\ b$.

Např. „býti menší než“ je podrelací relace „býti menší nebo rovno“.

2.2.2 Množinové operace s relacemi. Mějme dvě relace R a S z množiny A do množiny B. Pak *průnikem* relací R a S je relace $R \cap S$; *sjednocením* těchto relací je relace $R \cup S$; *doplňkem* relace R je relace $\overline{R} = (A \times B) \setminus R$.

Např. označíme-li T relaci rovnosti na množině reálných čísel $\mathbb{R}$ a S relaci býti ostře menší také na množině $\mathbb{R}$, pak $T \cap S = \emptyset$ a $T \cup S$ je relace býti menší nebo roven. Doplňkem relace T je nerovnost, tj. relace $\overline{T} = \{(a,b) \mid a, b \in \mathbb{R}, a \neq$ $\neq b\}$.

2.2.3 Inversní relace. Mějme relaci R z množiny A do množiny B. Pak *inversní relací* k relaci R je relace R^{-1} z množiny B do množiny A definovaná takto:

$$x\ R^{-1}\ y \quad \text{právě tehdy, když} \quad y\ R\ x.$$

2.2.4 Poznámka. Uvědomte si, že inversní relace k relaci z příkladu 6 na straně 8 existuje: Je to relace R^{-1}, kde $x \in \mathbb{R}$, $y \in \mathbb{R}$ a platí $x\ R^{-1}\ y$ právě tehdy, když $y\ R\ x$, tj. právě tehdy, když $x = \sin y$. Přesto z matematiky víte, že inversní funkce k funkci $y = \sin x$ existuje pouze tehdy, když se omezíme na funkci $f(x) = \sin x$ definovanou na intervalu $\langle -\pi/2, \pi/2 \rangle$. Je to proto, že inversní relace sice existuje, ale není již zobrazením; není totiž pravda, že pro každé $-1 \leq x \leq 1$ existuje právě jedno $y \in \mathbb{R}$ tak, že $x = \sin y$.

2.2.5 Poznámka. Jsou-li množiny A, B konečné a representujeme-li relaci R maticí M_R, pak matice inversní relace R^{-1} je matice transponovaná k matici M_R. Tj.

$$m_{R^{-1}}(j,i) = m_R(i,j),\ i = 1,\dots,n,\ j = 1,\dots,k.$$

2.2.6 Skládání relací. Mějme relaci R z množiny A do množiny B a S relaci z množiny B do množiny C. Pak *složená relace* $R \circ S$ je relace z množiny A do množiny C definovaná předpisem:

$a\ R \circ S\ c$ právě tehdy, když existuje $b \in B$ takové, že $a\ R\ b$ a $b\ S\ c$.

2.2.7 Poznámka. Je-li M_R matice relace R a M_S matice relace S, pak matice relace $R \circ S$ je rovna součinu

$$M_{R\circ S} = M_R \cdot M_S,$$

s tím, že „počítáme“ $1 + 1 = 1$.

2.2.8 Tvrzení. Skládání relací je asociativní. Přesněji, je-li R relace z množiny A do množiny B, relace S z množiny B do množiny C a relace T z množiny C do množiny D, pak platí

$$R \circ (S \circ T) = (R \circ S) \circ T.$$

2.2.9 Poznámka. Skládání relací **není** komutativní, tj. neplatí $R \circ S = S \circ R$. To je vidět z následujícího příkladu:

Uvažujme množinu A všech lidí v České republice a dvě relace R, S definované na A:

$a\ R\ b$ právě tehdy, když a je sourozenec b a $a \neq b$

$c\ S\ d$ právě tehdy, když c je dítětem d.

Ukažme, že $R \circ S \neq S \circ R$.

Abychom ukázali, že $R \circ S \neq S \circ R$, stačí najít dva lidi x, y tak, že platí $x\ R \circ S\ y$ a neplatí $x\ S \circ R\ y$. Uvažujme dvojici synovec a a strýc b. Platí $a\ S \circ R\ b$, protože jeden z rodičů a je sourozencem strýce b. Neplatí ale, že $a\ R \circ S\ b$ protože to by znamenalo, že některý ze sourozenců a by byl rodičem strýce b.

2.3 Relace na množině

Relace $R \subseteq A \times A$ se nazývá relace na množině A.

2.3.1 Vlastnosti relací na množině Řekneme, že relace R na množině A je

1. *reflexivní*, jestliže pro všechna $a \in A$ platí $a \; R \; a$;
2. *symetrická*, jestliže pro všechna $a, b \in A$ platí: je-li $a \; R \; b$, pak také $b \; R \; a$;
3. *antisymetrická*, jestliže pro všechna $a, b \in A$ platí: je-li $a \; R \; b$ a $b \; R \; a$, pak nutně $a = b$;
4. *tranzitivní*, jestliže pro všechna $a, b, c \in A$ platí: je-li $a \; R \; b$ a $b \; R \; c$, pak nutně $a \; R \; c$.

2.3.2 Uvažujme relaci nerovnosti R na množině přirozených čísel $\mathbb{N}$ (tj. $n \; R \; m$ právě tehdy, když n a m jsou různá přirozená čísla). Tato relace není reflexivní, protože pro žádné $n \in \mathbb{N}$ neplatí $n \neq n$. Zato je symetrická: Je-li $n \neq m$, pak také $m \neq n$. Relace R není antisymetrická, protože např. $2 \neq 3$, $3 \neq 2$ a 2 a 3 jsou různá čísla (tj. $2 \; R \; 3$ a $3 \; R \; 2$ a přesto $2 \neq 3$). Tato relace také není tranzitivní, protože např. $2 \neq 3$ a $3 \neq 2$ a přesto $2 = 2$ (tj. $2 \; R \; 3$ a $3 \; R \; 2$ a přesto není $2 \; R \; 2$).

Relace menší nebo rovno $\leq$ na množině $\mathbb{R}$ je reflexivní, neboť $a \leq a$ pro všechna reálná a. Je i antisymetrická, neboť jakmile pro dvě reálná čísla a, b platí $a \leq b$ a $b \leq a$, pak $a = b$. Je také tranzitivní, neboť je-li $a \leq b$ a $b \leq c$, pak i $a \leq c$.

2.3.3 Relace ekvivalence. Relace R na množině A se nazývá *ekvivalence*, jestliže je reflexivní, symetrická a tranzitivní.

2.3.4 Příklad. Relace R na množině všech celých čísel $\mathbb{Z}$ definovaná předpisem:

$$m \; R \; n \quad \text{právě tehdy, když} \quad m - n \text{ je sudé, } (m, n \in \mathbb{Z}),$$

je ekvivalence.

Relace R je reflexivní, protože pro každé $m \in \mathbb{Z}$ je $m - m = 0$ a nula je sudé číslo. Tedy $m \; R \; m$.

Relace R je symetrická protože, je-li $m \; R \; n$, tj. $m - n = 2k$ pro nějaké k, je i $n - m$ sudé ($n - m = -2k$) a proto $n \; R \; m$.

Navíc R je tranzitivní: Máme-li tři čísla $m, n, p \in \mathbb{Z}$ taková, že $m \; R \; n$ a $n \; R \; p$, tj. $m - n = 2k$ a $n - p = 2l$ pro nějaká k a l, potom $m - p = (m - n) + (n - p) = 2k + 2l = 2(k + l)$. Odtud plyne $m \; R \; p$.

2.3.5 Příklad. Na množině $A = \{(p, q) \mid q \neq 0, \; p, q \in \mathbb{Z}\}$ je relace S definována předpisem:

$$(p, q) \; S \; (m, n) \quad \text{právě tehdy, když} \quad pn = qm.$$

Ukažme, že S je ekvivalence.

S je reflexivní: Pro všechny $(p, q) \in A$ máme $(p, q) \; S \; (p, q)$, protože $pq = qp$.

S je symetrická: je-li $(p,q)\ S\ (m,n)$, pak $pn = qm$ a tedy i $mq = np$. To znamená, že $(m,n)\ S\ (p,q)$.

S je tranzitivní: Předpokládejme $(p,q)\ S\ (m,n)$ a $(m,n)\ S\ (r,s)$, tj. $pn = qm$ a $ms = nr$. K tomu, abychom ukázali, že S je tranzitivní relace, potřebujeme ověřit, že $(p,q)\ S\ (r,s)$, tj., že $ps = qr$. Protože $pn = qm$ a n je nenulové, máme $p = \frac{qm}{n}$. Odtud $ps = \frac{qm}{n}\, s = \frac{q}{n}\, ms = \frac{q}{n}\, nr = qr$. (Užili jsme rovnost $ms = nr$.) Tedy S je tranzitivní relace.

2.3.6 Třídy ekvivalence. Je dána relace ekvivalence R na množině A. *Třídou ekvivalence* R odpovídající prvku $a \in A$ nazýváme množinu $R[a] = \{b \in A \mid a\ R\ b\}$.

Množinu všech tříd dané ekvivalence, tj. množinu $\{R[a] \mid a \in A\}$ často nazýváme *faktorovou množinou podle ekvivalence* R a značíme A/R.

2.3.7 Příklady. Uvažujme relaci ekvivalence R z příkladu 2.3.4. Tato relace má dvě třídy ekvivalence, a to $R[a] = \{b \in A \mid a\ R\ b\}$. množinu všech sudých čísel a množinu všech lichých čísel.

Hledejme třídy ekvivalence S z příkladu 2.3.5 ke dvojicím $(p,q) \in A$. Např. s dvojicí $(1,2)$ jsou v relaci všechny dvojice (s,t) pro něž $1 \cdot t = s \cdot 2$, tj. $t = 2s$. Kdybychom si dvojice představili jako zlomky, byly by to všechny zlomky, které po zkrácení dávají racionální číslo $\frac{1}{2}$. Platí, že $(s,t)\ S\ (p,q)$ právě tehdy, když $\frac{s}{t} = \frac{p}{q}$. Tedy třídy ekvivalence odpovídají jednotlivým racionálním číslům.

2.3.8 Tvrzení. Nechť R je ekvivalence na množině A. Množina tříd ekvivalence $\{R[a] \mid a \in A\}$ má tyto vlastnosti:

1. Každý prvek $a \in A$ leží v $R[a]$ a platí rovnost $\bigcup\{R[a] \mid a \in A\} = A$.
2. Třídy ekvivalence $R[a]$ jsou po dvou disjunktní, tj. jestliže $R[a] \cap R[b] \neq \emptyset$, pak $R[a] = R[b]$.

2.3.9 Rozklad množiny. Mějme neprázdnou množinu A. Množina $\mathcal{S}$ neprázdných podmnožin množiny A se nazývá *rozklad množiny* A, jestliže jsou splněny následující podmínky:

1. Každý prvek $a \in A$ leží v některé podmnožině z $\mathcal{S}$, tj. $\bigcup \mathcal{S} = A$.
2. Prvky množiny $\mathcal{S}$ jsou po dvou disjunktní; tj. jestliže $X \cap Y \neq \emptyset$, pak $X = Y$ pro všechna $X, Y \in \mathcal{S}$.

2.3.10 Tvrzení. Nechť $\mathcal{S}$ je rozklad množiny A. Pak relace $R_{\mathcal{S}}$ definovaná:

$$a\ R_{\mathcal{S}}\ b \quad \text{právě tehdy, když} \quad a, b \in X \text{ pro nějaké } X \in \mathcal{S}$$

je ekvivalence na množině A.

2.3.11 Poznámka. Je dobré si uvědomit, že vyjdeme-li od ekvivalence R, utvoříme k ní rozklad na její třídy ekvivalence, načež k tomuto rozkladu utvoříme (podle předchozího tvrzení) opět ekvivalenci, pak dostaneme původní ekvivalenci R. Podobně, kdybychom vyšli od rozkladu, utvořili k němu ekvivalenci a k této ekvivalenci rozklad na její třídy, dostali bychom původní rozklad. Je proto jedno, zda se na ekvivalenci díváme jako na vztah (tj. relaci) nebo s ní

pracujeme jako s rozkladem na množiny těch prvků, které jsou pro nás „stejné“. Jedná se tedy o dva různé pohledy na tutéž matematickou realitu.

2.3.12 Uspořádání. Relaci R na množině A nazveme *uspořádání (částečné uspořádání)*, jestliže je reflexivní, antisymetrická a tranzitivní.

2.3.13 Příklady uspořádání.

1. Známé uspořádání reálných čísel je uspořádání ve smyslu předchozí definice, neboť pro všechna reálná čísla $a, b, c \in \mathbb{R}$ platí: $a \leq a$; jestliže $a \leq b$ a také $b \leq a$, pak nutně $a = b$; jestliže $a \leq b$ a $b \leq c$, pak také $a \leq c$.

2. Označme A množinu všech podmnožin množiny U. Pak relace $\subseteq$ „být podmnožinou“ je relace uspořádání na A. Ověření reflexivity, antisymetrie a tranzitivity přenecháme čtenáři.

3. Položme $A = \mathbb{N}$, kde $\mathbb{N}$ je množina přirozených čísel. Pak relace dělitelnosti definovaná $m \mid n$ právě tehdy, když m je dělitel čísla n (tj. když $n = k \cdot m$ pro nějaké $k \in \mathbb{N}$) je relace uspořádání. Ano, pro každá tři přirozená čísla m, n, k platí $m \mid m$; jestliže $m \mid n$ a $n \mid m$, pak $m = n$; jestliže $m \mid n$ a $n \mid k$, pak také $m \mid k$.

2.3.14 Největší a maximální prvek. Mějme množinu A a na ní relaci uspořádání $\preceq$. Řekneme, že prvek a množiny A je *největší prvek* množiny A, jestliže pro všechny prvky $x \in A$ platí $x \preceq a$. Řekneme, že prvek b množiny A je *maximální prvek* množiny A, jestliže neexistuje prvek $y \in A$, $y \neq b$, takový, že $b \preceq y$.

2.3.15 Tvrzení. Mějme množinu A a na ní uspořádání $\preceq$. Množina A má nejvýše jeden největší prvek; navíc, je-li a největší prvek množiny A, pak je jediným maximálním prvkem množiny A. Nemá-li množina A největší prvek, může mít několik maximálních prvků, anebo žádný.

2.3.16 Nejmenší a minimální prvek. Mějme množinu A a na ní relaci uspořádání $\preceq$. Řekneme, že prvek a množiny A je *nejmenší prvek* množiny A, jestliže pro všechny prvky $x \in A$ platí $a \preceq x$. Řekneme, že prvek b množiny A je *minimální prvek* množiny A jestliže neexistuje prvek $y \in A$, $y \neq b$, takový, že $y \preceq b$.

2.3.17 Tvrzení. Mějme množinu A a na ní uspořádání $\preceq$. Množina A má nejvýše jeden nejmenší prvek. Pokud nejmenší prvek existuje, je jediným minimálním prvkem. Nemá-li množina A nejmenší prvek, může mít několik minimálních prvků, anebo žádný.

2.3.18 Nesrovnatelné prvky. Mějme množinu A a na ní uspořádání $\preceq$. Řekneme, že prvky $a, b \in A$ jsou *nesrovnatelné*, jestliže neplatí ani $a \preceq b$, ani $b \preceq a$.

2.3.19 Poznámka. Má-li uspořádání několik maximálních prvků, pak jsou tyto prvky nesrovnatelné. Totéž platí o minimálních prvcích.

Kapitola 3

Výroková logika

3.1 Výroky

Matematická logika se zabývá studiem výroků, jejich vytvářením, jejich pravdivostí a jejich odvozováním. Nedefinujeme, co je to výrok (stejně jako jsme nedefinovali, co je to množina, co je to bod apod.); pouze zhruba popíšeme, co si pod pojmem výrok přestavujeme. Neformálně se dá říci, že výrok je každá věta, o které se dá rozhodnout, zda je pravdivá; nebo též: výrok je každá oznamovací věta. Tedy výrokem nejsou např. věty „Kéž by pršelo!“, „Prší?“ apod. Naopak výrokem jsou věty „Prší.“, „Svítí sluníčko.“, „Jedna a jedna jsou tři.“ apod. Výroková logika pracuje se základními výroky (o jejichž strukturu se dál již nestaráme) a zjišťuje pravdivost či nepravdivost složených výroků na základě pravdivosti základních výroků. K vytváření složitějších výroků používá tzv. logické spojky. Jsou to tyto logické spojky:

- není pravda, že; označujeme ji $\neg$ a nazýváme ji *negace*;
- a; označujeme ji $\wedge$ a nazýváme ji *konjunkce*;
- nebo; označujeme ji $\vee$ a nazýváme ji *disjunkce*;
- jestliže ..., pak; označujeme ji $\Rightarrow$ a nazýváme ji *implikace*;
- právě tehdy, když; označujeme ji $\Leftrightarrow$ a nazýváme ji *ekvivalence*.

3.1.1 Výrokové formule. Máme danou neprázdnou množinu A tzv. *elementárních výroků* (též jim říkáme *logické* nebo *výrokové proměnné*). Konečnou posloupnost prvků z množiny A, logických spojek a závorek nazýváme *výroková formule* (zkráceně jen *formule*), jestliže vznikla podle následujících pravidel:

1. Každá logická proměnná (elementární výrok) $a \in A$ je výroková formule.
2. Jsou-li α, β výrokové formule, pak $(\neg\alpha)$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \Rightarrow \beta)$ a $(\alpha \Leftrightarrow \beta)$ jsou také výrokové formule.
3. Nic jiného než to, co vzniklo pomocí konečně mnoha použití bodů 1 a 2, není výroková formule.

Všechny formule, které vznikly z logických proměnných množiny A, značíme $\mathcal{P}(\mathcal{A})$.

3.1.2 Poznámka. Spojka $\neg$ se nazývá *unární*, protože vytváří novou formuli z jedné formule. Ostatní zde zavedené spojky se nazývají *binární*, protože vytvářejí novou formuli ze dvou formulí.

V dalším textu výrokové (logické) proměnné označujeme malými písmeny např. $a, b, c, \ldots, x, y, z, \ldots$, výrokové formule označujeme malými řeckými písmeny např. $\alpha, \beta, \gamma, \ldots, \varphi, \psi, \ldots$.

3.1.3 Derivační strom formule. To, jak formule vznikla podle bodů 1 a 2, si můžeme znázornit na *derivačním stromu* dané formule. Jedná se o kořenový strom, kde každý vrchol, který není listem je ohodnocen logickou spojkou a jedná-li se o binární spojku, má vrchol dva následníky, jedná-li se o unární spojku, má vrchol pouze jednoho následníka. Přitom pro formule tvaru $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \Rightarrow \beta)$ odpovídá levý následník formuli α, pravý následník formuli β. Listy stromu jsou ohodnoceny logickými proměnnými.

3.1.4 Podformule. Z derivačního stromu formule α jednoduše poznáme všechny její podformule: *Podformule* formule α jsou všechny formule odpovídající podstromům derivačního stromu formule α.

3.1.5 Hloubka formule. *Hloubka formule* je definována jako výška stromu této formule. Například formule

$$(x \Rightarrow y) \Rightarrow ((\neg x \vee y) \wedge (y \Rightarrow \neg x))$$

má hloubku 4. Hloubka derivačního stromu logické proměnné je 0.

3.2 Pravdivostní ohodnocení

3.2.1 Pravdivostní ohodnocení je zobrazení $u\colon \mathcal{P}(A) \to \{0,1\}$, které splňuje pravidla

(1) $\neg\alpha$ je pravdivá právě tehdy, když α je nepravdivá;

(2) $\alpha \wedge \beta$ je pravdivá právě tehdy, když α a β jsou obě pravdivé;

(3) $\alpha \vee \beta$ je nepravdivá právě tehdy, když α a β jsou obě nepravdivé;

(4) $\alpha \Rightarrow \beta$ je nepravdivá právě tehdy, když α je pravdivá a β nepravdivá;

(5) $\alpha \Leftrightarrow \beta$ je pravdivá právě tehdy, když buď obě formule α a β jsou pravdivé nebo obě jsou nepravdivé.

3.2.2 Pravdivostní tabulky. Vlastnosti, které pravdivostní ohodnocení musí mít, znázorňujeme též pomocí tzv. pravdivostních tabulek logických spojek. Jsou to:

α	$\neg\alpha$
0	1
1	0

α	β	$\alpha \wedge \beta$	$\alpha \vee \beta$	$\alpha \Rightarrow \beta$	$\alpha \Leftrightarrow \beta$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

3.2.3 Věta. Každé zobrazení $u_0\colon A \to \{0,1\}$ jednoznačně určuje pravdivostní ohodnocení $u : \mathcal{P}(A) \to \{0,1\}$ takové, že $u_0(a) = u(a)$ pro všechna $a \in A$. Ohodnocení $u, v\colon \mathcal{P}(A) \to \{0,1\}$ jsou totožná právě tehdy, když pro všechny logické proměnné $x \in A$ platí $u(x) = v(x)$.

3.2.4 Pravdivostní tabulka formule. Víme, že pravdivostní hodnocení formule závisí pouze na ohodnocení těch logických proměnných, které obsahuje, a těch je konečně mnoho. Proto pravdivostní hodnotu dané formule dostáváme z pravdivostní tabulky takto:

Pravdivostní tabulka obsahuje sloupec pro každou logickou proměnnou a danou formuli. Pomáháme si i sloupci pro některé podformule dané formule. Tabulka obsahuje tolik řádků, kolik máme různých možností pravdivostních hodnot logických proměnných. Jestliže tedy formule obsahuje právě n proměnných, pak její tabulka má 2^n řádků. V každém řádku pak na základě vlastností a konstrukce formule dostáváme jednoznačně pravdivostní hodnotu (tj. 1 nebo 0) celé formule.

3.2.5 Příklad. Zde je pravdivostní tabulka formule $\varphi = (x \Rightarrow y) \Rightarrow ((\neg x \vee \vee y) \wedge (y \Rightarrow \neg x))$:

x	y	$x \Rightarrow y$	$\neg x \vee y$	$y \Rightarrow \neg x$	$(\neg x \vee y) \wedge (y \Rightarrow \neg x)$	φ
0	0	1	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	0	1
1	1	1	1	0	0	0

3.2.6 Tautologie, splnitelná formule, kontradikce. Formule se nazývá *tautologie*, jestliže je pravdivá ve všech pravdivostních ohodnoceních; nazývá se *kontradikce*, jestliže je nepravdivá ve všech pravdivostních ohodnoceních. Formule je *splnitelná*, jestliže existuje aspoň jedno pravdivostní ohodnocení, ve kterém je pravdivá.

3.2.7 Příklady. Formule $\alpha = (x \Rightarrow (y \wedge \neg y)) \Rightarrow \neg x$ je tautologie.

Formule $\beta = (\neg x \vee y) \Leftrightarrow (y \Rightarrow x)$ je splnitelná, ale není tautologie.

Formule $\gamma = \neg((x \Rightarrow \neg x) \Leftrightarrow \neg x)$ je kontradikce.

3.2.8 Souvislost s logickými obvody Každá výroková formule, která obsahuje pouze logické spojky $\neg$, $\wedge$ a $\vee$ se dá realizovat jako logický obvod s hradly odpovídajícími těmto spojkám, kde na vstupech jsou logické proměnné, vstupy se nevětví a obvody neobsahují cykly. Přidáme-li k použitelným prvkům i hradla realizující implikaci $\Rightarrow$ a ekvivalenci $\Leftrightarrow$, pak každé výrokové formuli odpovídá logický obvod. Máme-li danou formuli a odpovídající logický obvod, pak pravdivostní ohodnocení této formule je jednoznačně dáno hodnotami nula nebo jedna na vstupech a jeho hodnota je přesně hodnota na výstupu daného logického obvodu. Tautologii odpovídá logický obvod, který při všech kombinacích vstupů dává vždy na výstupu jedničku; kontradikci odpovídá takový logický obvod, který naopak vždy na výstupu dává nulu. Splnitelné formuli odpovídá logický obvod, který pro aspoň jednu kombinaci vstupů dává na výstupu jedničku.

3.3 Sémantický důsledek

V matematické logice nás nejvíce zajímá otázka: Jsou dány výroky (předpoklady našeho odvozování), co z těchto výroků vyplývá (na co z nich můžeme usoudit)?

3.3.1 Splnitelné množiny formulí. Řekneme, že množina formulí S je *pravdivá* v ohodnocení u, jestliže každá formule z S je pravdivá v u, tj. je-li $u(\varphi) = 1$ pro všechna $\varphi \in S$.

Řekneme, že množina formulí S je *splnitelná* jestliže existuje pravdivostní ohodnocení u, ve kterém je S je pravdivá.

3.3.2 Sémantický důsledek. Řekneme, že formule φ je *sémantickým*, též *konsekventem* nebo *tautologickým důsledkem* množiny formulí S, jestliže φ je pravdivá v každém ohodnocení u, v němž je pravdivá S.

Fakt, že formule φ je konsekventem množiny S, označujeme $S \models \varphi$.

3.3.3 Formule φ *není sémantický důsledek* množiny S, jestliže existuje pravdivostní ohodnocení u takové, že množina S je pravdivá v u a $u(\varphi) = 0$.

3.3.4 Příklady. Pro libovolné formule α, β platí:
1) $\{\alpha \Rightarrow \beta, \alpha\} \models \beta$;
2) $\{\alpha \wedge \beta\} \models \alpha$.

1) Má-li být množina formulí $\{\alpha \Rightarrow \beta, \alpha\}$ pravdivá v nějakém ohodnocení u, pak v u je pravdivá formule α. Navíc, je-li $u(\alpha) = 1$ a formule $\alpha \Rightarrow \beta$ je pravdivá v u, musí být v u pravdivá i formule β. Odtud plyne, že $\{\alpha \Rightarrow \beta, \alpha\} \models \beta$.

2) Je-li v nějakém pravdivostním ohodnocení u pravdivá formule $\alpha \wedge \beta$, pak jsou v u pravdivé obě formule α, β. Tedy platí $\{\alpha \wedge \beta\} \models \alpha$.

3.3.5 Konvence. Pro jednoduchost píšeme $\alpha \models \beta$ místo $\{\alpha\} \models \beta$ a $\models \varphi$ místo $\emptyset \models \varphi$.

3.3.6 Tvrzení.

1. Je-li S množina formulí a $\varphi \in S$, pak φ je konsekventem S, tj. $S \models \varphi$ pro každou $\varphi \in S$.

   Jestliže v nějakém pravdivostním ohodnocení u je pravdivá celá množina S, pak je v něm pravdivá i formule $\varphi \in S$.

2. Tautologie je konsekventem každé množiny formulí S.

   Tautologie je formule, která je pravdivá v každém pravdivostním ohodnocení. Nemůže se tedy stát, že bychom měli pravdivostní ohodnocení u, v němž by byla pravdivá množina S a nebyla pravdivá naše tautologie.

3. Formule φ je tautologie právě tehdy, když $\models \varphi$.

   Z vlastnosti 2 už víme, že pro tautologii φ platí $\models \varphi$. Předpokládejme, že formule φ není tautologie. Ukážeme, že v tomto případě φ není konsekventem prázdné množiny $\emptyset$: Protože φ není tautologie, existuje ohodnocení u takové, že $u(\varphi) = 0$. V ohodnocení u jsou všechny formule z prázdné množiny pravdivé (žádné formule totiž v $\emptyset$ nejsou) a přitom $u(\varphi) = 0$. Tedy neplatí $\models \varphi$.

4. Každá formule je konsekventem množiny formulí $\{\alpha, \neg\alpha\}$.

   Označme $S = \{\alpha, \neg\alpha\}$. Vezměme libovolnou formuli φ. Kdyby φ nebyla konsekventem množiny S, pak by existovalo pravdivostní ohodnocení u takové, že S je v u pravdivá a φ je v u nepravdivá. Ale množina S nemůže být v žádném pravdivostním ohodnocení pravdivá, vždyť α a $\neg\alpha$ nemohou být pravdivé současně! Proto platí $S \models \varphi$.

3.4 Operátor Con

3.4.1 Množina $Con(S)$ všech sémantických důsledků množiny formulí S je definována rovností

$$Con(S) = \{\alpha \mid \alpha \text{ je formule a } S \models \alpha\}.$$

3.4.2 Tvrzení. Pro každé dvě množiny formulí S, T platí:

a) $Con(S) \neq \emptyset$.

Ano, je-li formule α tautologie, pak je sémantickým důsledkem libovolné množiny formulí S. To znamená, že $Con(S)$ obsahuje vždy alespoň všechny tautologie.

b) $S \subseteq Con(S)$.

Z 3.3.6 víme, že každá formule z množiny S je sémantickým důsledkem množiny S. Proto je $S \subseteq Con(S)$.

c) Jestliže $S \subseteq T$, pak $Con(S) \subseteq Con(T)$.

Přeformulujme toto tvrzení: Jestliže $S \subseteq T$, pak pro každou formuli φ, pro kterou $S \models \varphi$, máme také $T \models \varphi$. (Méně formálně: vyplývá-li nějaká formule z menší množiny předpokladů, vyplývá i z větší množiny předpokladů.)

Ukážeme, že pro libovolné pravdivostní ohodnocení u, ve kterém je pravdivá množina formulí T, platí $u(\varphi) = 1$. Máme pravdivostní ohodnocení u takové, že množina formulí T je pravdivá. Protože $S \subseteq T$, je v tomto ohodnocení pravdivá i množina S. Formule φ je sémantickým důsledkem S, a proto je $u(\varphi) = 1$.

d) $Con(S) = Con(Con(S))$.

Tuto vlastnost můžeme přepsat následujícím způsobem: Máme dány formule α, $\varphi_1, \varphi_2, \ldots, \varphi_k$, pro které platí

$$\{\varphi_1, \varphi_2, \ldots, \varphi_k\} \models \alpha$$

a $S \models \varphi_1, S \models \varphi_2, \ldots, S \models \varphi_k$, pak

$$S \models \alpha.$$

Uvažujme libovolné pravdivostní ohodnocení u takové, že S je pravdivá v u. Pak platí $u(\varphi_1) = 1$, $u(\varphi_2) = 1$, ..., $u(\varphi_k) = 1$. To znamená, že v ohodnocení u je pravdivá celá množina $\{\varphi_1, \varphi_2, \ldots, \varphi_k\}$. Protože formule α je sémantický důsledek množiny $\{\varphi_1, \varphi_2, \ldots, \varphi_k\}$, máme $u(\alpha) = 1$.

3.4.3 Tvrzení. Pro množinu formulí S a formuli φ platí

$$S \models \varphi \quad \text{právě tehdy, když} \quad S \cup \{\neg\varphi\} \quad \text{je nesplnitelná.}$$

3.4.4 Poznámka. Je-li množina S nesplnitelná, pak $S \models \psi$ pro všechny formule ψ.

Ano, je-li množina S nesplnitelná, pak množina $S \cup \{\neg\psi\}$ je také nesplnitelná a proto podle tvrzení 3.4.3 platí $S \models \psi$.

Uvědomte si, že předcházející zdůvodnění je stejné jako bylo zdůvodnění vlastnosti 4 viz 4.

3.4.5 Věta. Pro formule φ a ψ platí

$$\varphi \models \psi \quad \text{právě tehdy, když} \quad \varphi \Rightarrow \psi \quad \text{je tautologie.}$$

3.4.6 Věta o dedukci. Pro množinu formulí S a formule φ a ψ platí

$$S \cup \{\varphi\} \models \psi \quad \text{právě tehdy, když} \quad S \models (\varphi \Rightarrow \psi).$$

3.5 Tautologická ekvivalence

Až dosud pro nás $p \wedge q$ a $q \wedge p$ byly dvě různé formule, byly to totiž různé posloupnosti znaků. Zajímá-li nás ovšem hlavně pravdivost formulí, pak se tyto dvě formule „od sebe neliší".

3.5.1 Tautologická ekvivalence formulí. Řekneme, že formule φ a ψ jsou *tautologicky ekvivalentní* (také *sémanticky ekvivalentní*), jestliže $\varphi \models \psi$ a také $\psi \models \varphi$. Fakt, že φ a ψ jsou tautologicky ekvivalentní, označujeme $\varphi \models\!\models \psi$.

3.5.2 Pozorování. Formule φ a ψ jsou tautologicky ekvivalentní právě tehdy, když pro každé pravdivostní ohodnocení u platí $u(\varphi) = u(\psi)$.

3.5.3 Věta. Relace $\models\!\models$ na množině všech formulí $\mathcal{P}(A)$ je ekvivalence. Navíc, jsou-li α, β, γ a δ formule splňující $\alpha \models\!\models \beta$ a $\gamma \models\!\models \delta$, pak platí

1. $\neg\alpha \models\!\models \neg\beta$;
2. $(\alpha \wedge \gamma) \models\!\models (\beta \wedge \delta)$, $(\alpha \vee \gamma) \models\!\models (\beta \vee \delta)$,
   $(\alpha \Rightarrow \gamma) \models\!\models (\beta \Rightarrow \delta)$, $(\alpha \Leftrightarrow \gamma) \models\!\models (\beta \Leftrightarrow \delta)$.

3.5.4 Pro každé formule α, β a γ platí

1. $\alpha \wedge \alpha \models\!\models \alpha$, $\alpha \vee \alpha \models\!\models \alpha$ (idempotence $\wedge$ a $\vee$);
2. $\alpha \wedge \beta \models\!\models \beta \wedge \alpha$, $\alpha \vee \beta \models\!\models \beta \vee \alpha$ (komutativita $\wedge$ a $\vee$);
3. $\alpha \wedge (\beta \wedge \gamma) \models\!\models (\alpha \wedge \beta) \wedge \gamma$, $\alpha \vee (\beta \vee \gamma) \models\!\models (\alpha \vee \beta) \vee \gamma$ (asociativita $\wedge$ a $\vee$);
4. $\alpha \wedge (\beta \vee \alpha) \models\!\models \alpha$, $\alpha \vee (\beta \wedge \alpha) \models\!\models \alpha$ (absorpce $\wedge$ a $\vee$);
5. $\neg\neg\alpha \models\!\models \alpha$;
6. $\neg(\alpha \wedge \beta) \models\!\models (\neg\alpha \vee \neg\beta)$, $\neg(\alpha \vee \beta) \models\!\models (\neg\alpha \wedge \neg\beta)$ (de Morganova pravidla);
7. $\alpha \wedge (\beta \vee \gamma) \models\!\models (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$, $\alpha \vee (\beta \wedge \gamma) \models\!\models (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ (distributivní zákony).

   Je-li navíc T libovolná tautologie a F libovolná kontradikce, pak

8. $T \wedge \alpha \models\!\models \alpha$, $T \vee \alpha \models\!\models T$, $F \wedge \alpha \models\!\models F$, $F \vee \alpha \models\!\models \alpha$;
9. $\alpha \wedge \neg\alpha \models\!\models F$, $\alpha \vee \neg\alpha \models\!\models T$.

3.5.5 Pozorování. Pro dvě formule φ a ψ platí $\varphi \models\!\models \psi$ právě tehdy, když formule $\varphi \Leftrightarrow \psi$ je tautologie.

Vrátíme-li se k souvislosti s logickými obvody, můžeme říci, že dvě formule jsou tautologicky ekvivalentní právě tehdy, když jejich logické obvody realizují tutéž booleovskou funkci, tj. dávají stejné výstupy při stejných vstupech.

Pro pojem sémantického důsledku není v teorii logických obvodů přímý ekvivalent.

3.5.6 Booleovské funkce. *Booleovskou funkcí* n proměnných rozumíme funkci, která každé n-tici nul a jedniček přiřazuje buď nulu nebo jedničku; jedná se tedy o zobrazeni $f\colon \{0,1\}^n \to \{0,1\}$. Místo booleovská funkce se často také říká *Booleova funkce*. Každé výrokové formuli odpovídá booleovská funkce — stačí se podívat na poslední sloupec pravdivostní tabulky formule. Dvě formule jsou tautologicky ekvivalentní právě tehdy, když oběma odpovídají stejné Booleovy funkce. V tomto smyslu najít „jednodušší“ formuli tautologicky ekvivalentní s danou formulí je jinak řečena úloha najít v nějakém smyslu jednodušší formuli (většinou kratší, nebo s menší hloubkou), která odpovídá stejné Booleově funkci. A to je úloha v teorii logických obvodů častá.

3.6 Další logické spojky

3.6.1 Unární logické spojky. Všechny unární Booleovy funkce (tj. Booleovy funkce jednoho argumentu) jsou dány v následující tabulce:

x	f_1	f_2	f_3	f_4
0	0	0	1	1
1	0	1	0	1

Funkce f_1 nemění svou hodnotu, je konstantní nepravda; pro nás to bude nová spojka F zvaná *kontradikce*. Obdobně funkce f_4 je konstatní pravda; pro nás to bude opět nová spojka T zvaná *tautologie*. Tyto dvě spojky, kontradikce a tautologie, mají tu vlastnost, že nezávisí na ohodnocení žádné logické proměnné (tj. závisí na „nula“ logických proměnných). Proto jim říkáme *nulární logické spojky*.

3.6.2 Binární logické spojky. Binárními booleovské funkce jsou v následující tabulce:

x	y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

(Jestliže někomu v předcházející tabulce sloupce funkcí f_i připadají jako binární zápisy čísel 0 až 15, pak to není náhoda. Toto jsou všechny uspořádané čtveřice nul a jedniček uspořádané lexikograficky.)

Funkce f_0 je konstantní nepravda, je to kontradikce F. Obdobně poslední funkce f_{15} je tautologie T.

Funkce f_1 odpovídá formuli $x \wedge y$. Obdobně funkce f_7 odpovídá formuli $x \vee y$, funkce f_9 formuli $x \Leftrightarrow y$ a funkce f_{13} formuli $x \Rightarrow y$.

Funkce f_3 kopíruje proměnnou x, stejně tak funkce f_5 kopíruje proměnnou y. Funkce f_{10} odpovídá formuli $\neg y$. Podobně funkce f_{12} odpovídá formuli $\neg x$. Tedy se nejedná o možné „nové spojky“.

Funkce f_{11} odpovídá formuli $y \Rightarrow x$, tedy opět spojce $\Rightarrow$. Funkce f_2 a f_4 odpovídají po řadě formulím $\neg(x \Rightarrow y)$ a $\neg(y \Rightarrow x)$.

Zvláštní význam mají zbylé tři funkce f_6, f_8 a f_{14}, jimž odpovídají nové logické spojky.

3.6.3 Vylučovací nebo — XOR. Logická spojka $\oplus$, nazývaná *vylučovací nebo* (též *XOR*), je definována

$$x \oplus y \models\!\!\!\models \neg(x \Leftrightarrow y).$$

Vylučovacímu nebo odpovídá funkce f_6.

3.6.4 Shefferův operátor — NAND. Logická spojka $|$, nazývaná *Shefferův operátor* (též *NAND*), je definována

$$x \mid y \models\!\!\!\models \neg(x \wedge y).$$

Shefferovu operátoru odpovídá funkce f_{14}.

3.6.5 Peirceova šipka — NOR. Logická spojka $\downarrow$, nazývaná *Peirceova šipka* (též *NOR*), je definována

$$x \downarrow y \models\!\!\!\models \neg(x \vee y).$$

Peirceově šipce odpovídá funkce f_8.

3.7 Úplné systémy logických spojek

3.7.1 Úplné systémy logických spojek. Řekneme, že množina logických spojek Δ tvoří *úplný systém logických spojek*, jestliže pro každou formuli α existuje formule β s ní tautologicky ekvivalentní, která používá pouze spojky z množiny Δ.

3.7.2 Tvrzení. Nechť Δ tvoří úplný systém logických spojek a nechť Π je množina spojek. Jestliže platí

1. pro každou binární spojku $\square \in \Delta$ existuje formule α obsahující pouze spojky z množiny Π a taková, že $\alpha \leftrightarrows x \square y$,
2. pro každou unární spojku $\diamond \in \Delta$ existuje formule β obsahující pouze spojky z množiny Π a taková, že $\beta \leftrightarrows \diamond x$,
3. pro každou nulární spojku $K \in \Delta$ existuje formule γ obsahující pouze spojky z množiny Π a taková, že $\gamma \leftrightarrows K$,

pak Π je také úplný systém logických spojek.

3.7.3 Úplné systémy logických spojek.

1. Množina $\Delta = \{\neg, \vee\}$ tvoří úplný systém logických spojek.

   Víme, že $\{\neg, \wedge, \vee, \Rightarrow\}$ tvoří úplný systém logických spojek. Proto budeme-li umět „vytvořit“ formule $a \wedge b$ a $a \Rightarrow b$ pomocí spojek $\neg$ a $\vee$, budeme vědět, že $\Delta = \{\neg, \vee\}$ tvoří úplný systém logických spojek. Platí:

   $$(a \Rightarrow b) \leftrightarrows (\neg a \vee b) \quad \text{a} \quad (a \wedge b) \leftrightarrows \neg(\neg a \vee \neg b).$$

2. Množina $\Delta = \{\neg, \wedge\}$ tvoří úplný systém logických spojek.

   Využijeme předchozího tvrzení: protože množina $\Delta = \{\neg, \vee\}$ tvoří úplný systém logických spojek, stačí ověřit, že $\vee$ můžeme „vytvořit“ pomocí logických spojek $\neg, \wedge$. Přitom

   $$(a \vee b) \leftrightarrows \neg(\neg a \wedge \neg b).$$

3. Množina $\Delta = \{\neg, \Rightarrow\}$ tvoří úplný systém logických spojek.

   Obdobně jako výše si stačí uvědomit, že

   $$(a \Rightarrow b) \leftrightarrows \neg(a \wedge \neg b) \quad \text{nebo} \quad (a \Rightarrow b) \leftrightarrows (\neg a \vee b).$$

4. Množina $\{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$ (a tudíž ani žádná její podmnožina) netvoří úplný systém logických spojek.

   Stačí si uvědomit, že každá formule obsahující pouze spojky $\wedge$, $\vee$, $\Rightarrow$ a $\Leftrightarrow$ je pravdivá v pravdivostním ohodnocení u_0, v němž jsou pravdivé všechny elementární výroky. Ovšem formule $\neg a$ je v tomto ohodnocení u_0 nepravdivá.

5. Každá z množin $\{|\}$ a $\{\downarrow\}$ je úplný systém logických spojek.

   Uvažujem nejprve spojku $\{|\}$. Využijeme fakt, že $\{\neg, \wedge\}$ tvoří úplný systém logických spojek. Stačí proto „vytvořit“ spojky $\neg$ a $\wedge$ pomocí spojky $|$. A to je možné, protože

   $$\neg a \models\!\!\!\models (a \mid a) \quad \text{a} \quad (a \wedge b) \models\!\!\!\models \neg(a \mid b) \models\!\!\!\models (a \mid b) \mid (a \mid b).$$

   Pro spojku $\downarrow$ postupujeme analogicky, pouze používáme $\{\neg, \vee\}$ jako úplný systém logických spojek místo množiny $\{\neg, \wedge\}$. Máme

   $$\neg a \models\!\!\!\models (a \downarrow a) \quad \text{a} \quad (a \vee b) \models\!\!\!\models \neg(a \downarrow b) \models\!\!\!\models (a \downarrow b) \downarrow (a \downarrow b).$$

3.8 CNF a DNF

Každé formuli o n logických proměnných odpovídá pravdivostní tabulka. Na tuto tabulku se můžeme dívat jako na zobrazení, které každé n-tici 0 a 1 přiřazuje 0 nebo 1. Ano, řádek pravdivostní tabulky je popsán n-ticí 0 a 1, hodnota je pak pravdivostní hodnota formule pro toto dosazení za logické proměnné. Zobrazení z množiny všech n-tic 0 a 1 do množiny $\{0, 1\}$ se nazývá *booleovská funkce*. Naopak platí, že pro každou booleovskou funkci existuje formule, která této funkci odpovídá. Ukážeme v dalším, že dokonce můžeme volit formu ve speciálním tvaru, v tzv. *konjunktivním normálním tvaru* a *disjunktivním normálním tvaru*.

3.8.1 Booleovská funkce. *Booleovskou funkcí n proměnných*, kde n je přirozené číslo, rozumíme každé zobrazení $f\colon \{0,1\}^n \to \{0,1\}$, tj. zobrazení, které každé n-tici $(x_1, x_2, \ldots, x_n)$ nul a jedniček přiřazuje nulu nebo jedničku (označenou $f(x_1, x_2, \ldots, x_n)$).

3.8.2 Disjunktivní normální tvar. *Literál* je logická proměnná nebo negace logické proměnné. Řekneme, že formule je v *disjunktivním normálním tvaru*, zkráceně v *DNF*, jestliže je disjunkcí jedné nebo několika formulí, z nichž každá je literálem nebo konjunkcí literálů.

Poznamenejme, že literálu nebo konjunkci literálů se také říká *minterm*. Jestliže každý minterm obsahuje všechny proměnné, říkáme, že se jedná o *úplnou DNF*.

3.8.3 Věta. Ke každé booleovské funkci f existuje formule v DNF odpovídající f.

Zdůvodnění: Předpokládejme, že funkce f nabývá v aspoň jednom řádku hodnotu 1. Pro každý řádek, ve kterém Booleova funkce f nabývá hodnotu 1, utvoříme jeden minterm o tolika literálech, kolik máme proměnných. Minterm tvoříme takto: Jestliže v daném řádku proměnná x měla hodnotu 1, do mintermu dáme literál x, jestliže měla hodnotu 0, dáme do mintermu literál $\neg x$. Výsledná formule je disjunkce mintermů odpovídajících jednotlivým řádkům, ve kterých má funkce f hodnotu 1. Není těžké se přesvědčit, že takto utvořená formule odpovídá dané Booleově funkci f.

3.8.4 Poznámka. Jestliže funkce f nenabývá jenom hodnotu nula, pak formuli utvořené v předchozím zdůvodnění se také říká *formule v úplném disjunktivním normálním tvaru*. Je to proto, že všechny její mintermy obsahují tolik literálů, kolik proměnných má funkce f. Často je možné tuto formuli zjednodušit.

3.8.5 Důsledek. Ke každé formuli α existuje formule β, která je v DNF a navíc $\alpha \models\!\models \beta$.

3.8.6 Konjunktivní normální tvar. Řekneme, že formule je v *konjunktivním normálním tvaru*, zkráceně v *CNF*, jestliže je konjunkcí jedné nebo několika formulí, z nichž každá je literálem nebo disjunkcí literálů.

Poznamenejme, že literálu nebo disjunkci literálů se také říká *maxterm* nebo *klausule*. Jestliže každá klausule obsahuje všechny proměnné, říkáme, že se jedná o *úplnou CNF*.

3.8.7 Věta. Ke každé booleovské funkci f existuje formule v CNF odpovídající f.

Zdůvodnění: Jestliže funkce f nabývá jenom hodnoty 1, pak odpovídá tautologii a tu lze vyjádřit formulí $x \vee \neg x$. Uvědomte si, že tato formule je v CNF, skládá se totiž z jedné klausule. Jestliže Booleova funkce f nabývá aspoň v jednom řádku hodnoty 0, popíšeme mintermy všechny řádky, v nichž je hodnota funkce f rovna 0 a utvoříme formuli, která je konjunkcí všech negací mintermů. Použitím de Morganova zákona dostaneme hledanou formuli v konjunktivním normálním tvaru.

3.8.8 Poznámka. Jestliže funkce f nenabývá jenom hodnotu jedna, pak formuli utvořené v předchozím zdůvodnění se také říká *formule v úplném konjunktivním normálním tvaru*. Je to proto, že všechny její maxtermy obsahují tolik literálů, kolik proměnných má funkce f.

3.8.9 Důsledek. Ke každé formuli α existuje formule β, která je v CNF a navíc $\alpha \models\!\models \beta$.

3.9 Booleovský kalkul

Víme, že pro pravdivostní ohodnocení formulí platí:

$$\begin{aligned} u(a \vee b) &= \max\{u(a), u(b)\} = \max\{x, y\}, \\ u(a \wedge b) &= \min\{u(a), u(b)\} = \min\{x, y\}, \\ u(\neg a) &= 1 - u(a) = 1 - x. \end{aligned}$$

kde $x = u(a)$, $y = u(b)$.

3.9.1 Booleovské operace. To motivuje zavedení booleovských operací (pro hodnoty 0, 1):

součin	$x \cdot y$	$=$	$\min\{x, y\}$,
logický součet	$x + y$	$=$	$\max\{x, y\}$,
doplněk	$\overline{x}$	$=$	$1 - x$.

Pro tyto operace platí řada rovností, tak, jak je známe z výrokové logiky:

3.9.2 Tvrzení. Pro všechna $x, y, z \in \{0, 1\}$ platí:

1. $x \cdot x = x,\ x + x = x$;
2. $x \cdot y = y \cdot x,\ x + y = y + x$;
3. $x \cdot (y \cdot z) = (x \cdot y) \cdot z,\ x + (y + z) = (x + y) + z$;
4. $x \cdot (y + x) = x,\ x + (y \cdot x) = x$;
5. $x \cdot (y + z) = (x \cdot y) + (x \cdot z),\ x + (y \cdot z) = (x + y) \cdot (x + z)$;
6. $\overline{\overline{x}} = x$;
7. $\overline{x + y} = \overline{x} \cdot \overline{y},\ \overline{x \cdot y} = \overline{x} + \overline{y}$;
8. $x + \overline{x} = 1,\ x \cdot \overline{x} = 0$;
9. $x \cdot 0 = 0,\ x \cdot 1 = x$;
10. $x + 1 = 1,\ x + 0 = x$.

3.9.3 Booleovské funkce v DNF a CNF. Nyní můžeme pro booleovskou funkci psát pomocí výše uvedených operaci, např.

$$f(x, y, z) = \overline{x}\,\overline{y}\,\overline{z} + \overline{x}\,\overline{y}\,z + \overline{x}\,y\,\overline{z} + \overline{x}\,y\,z + x\,\overline{y}\,z$$

a říkat, že jsme Booleovu funkci napsali v *disjunktivní normální formě*. Rovnost opravdu platí; dosadíme-li za logické proměnné jakékoli hodnoty, pak pravá strana rovnosti určuje hodnotu booleovské funkce f. Obdobně jako jsme booleovskou funkci f napsali v disjunktivní normální formě, můžeme ji také napsat v *konjunktivní normální formě* a to takto:

$$f(x, y, z) = (\overline{x} + y + z)\,(\overline{x} + \overline{y} + z)\,(\overline{x} + \overline{y} + \overline{z}).$$

3.9.4 Věta. Každou booleovskou funkci lze napsat v disjunktivní normální formě i v konjunktivní normální formě.

3.10 Rezoluční metoda ve výrokové logice

Rezoluční metoda rozhoduje, zda daná množina klausulí je splnitelná nebo je nesplnitelná. Tím je také "universální metodou" pro řešení problémů, neboť:

1. Daná formule φ je sémantickým důsledkem množiny formulí S právě tehdy, když množina $S \cup \{\neg\varphi\}$ je nesplnitelná.

2. Ke každé formuli α existuje množina klausulí S_α taková, že α je pravdivá v pravdivostním ohodnocení právě tehdy, když v tomto ohodnocení je pravdivá množina S_α.

3.10.1 Klausule. Množinu všech logických proměnných označíme A. Připomeňme, že *literál* je buď logická proměnná (tzv. *positivní literál*) nebo negace logické proměnné (tzv. *negativní literál*). *Komplementární literály* jsou literály p a $\neg p$. *Klausule* je literál nebo disjunkce konečně mnoha literálů.

Zvláštní místo mezi klausulemi zaujímá *prázdná klausule*, tj. klausule, která neobsahuje žádný literál a tudíž se jedná o kontradikci. Proto ji budeme označovat F.

Pro jednoduchost zavedeme následující konvenci: Máme dánu klausuli C a literál p, který se v C vyskytuje. Pak symbolem $C \setminus p$ označujeme klausuli, která obsahuje všechny literály jako C kromě p. Tedy např. je-li $C = \neg x \vee y \vee \neg z$, pak

$$C \setminus \neg z = \neg x \vee y.$$

3.10.2 Rezolventa. Řekneme, že klausule D je *rezolventou klausulí* C_1 a C_2 právě tehdy, když existuje literál p takový, že p se vyskytuje v klausuli C_1, $\neg p$ se vyskytuje v klausuli C_2 a

$$D = (C_1 \setminus p) \vee (C_2 \setminus \neg p).$$

Také říkáme, že klausule D je *rezolventou* C_1 *a* C_2 *podle literálu* p a značíme $D = res_p(C_1, C_2)$.

3.10.3 Tvrzení. Máme dány dvě klausule C_1, C_2 a označme D jejich rezolventu. Pak D je sémantický důsledek množiny $\{C_1, C_2\}$.

3.10.4 Tvrzení. Máme dánu množinu klausulí S a označme D rezolventu některých dvou klausulí z množiny S. Pak množiny S a $S \cup \{D\}$ jsou pravdivé ve stejných pravdivostních ohodnoceních.

3.10.5 Rezoluční princip. Označme

$$\begin{aligned} R(S) &= S \cup \{D \mid D \text{ je rezolventa některých klausulí z } S\} \\ R^0(S) &= S \\ R^{i+1}(S) &= R(R^i(S)) \quad \text{pro} \quad i \in \mathbb{N} \\ R^\star(S) &= \bigcup\{R^i(S) \mid i \geq 0\}. \end{aligned}$$

Protože pro konečnou množinu logických proměnných existuje jen konečně mnoho klausulí, musí existovat přirozené číslo n takové, že $R^n(S) = R^{n+1}(S)$. Pro toto n platí $R^n(S) = R^\star(S)$.

3.10.6 Věta (Rezoluční princip). Množina klausulí S je splnitelná právě tehdy, když $R^\star(S)$ neobsahuje prázdnou klausuli F.

3.10.7 Základní postup. Předchozí věta dává návod, jak zjistit, zda daná množina klausulí je spnitelná nebo je nesplnitelná:

1. Formule množiny M převedeme do CNF a množinu M pak nahradíme množinou S všech klausulí vyskytujících se v některé formuli v CNF. Klausule, které jsou tautologiemi, vynecháme. Jestliže nám nezbyde žádná klausule, množina M se skládala z tautologií a je pravdivá v každém pravdivostním ohodnocení.
2. Vytvoříme $R^\star(S)$.
3. Obsahuje-li $R^\star(S)$ prázdnou klausuli, je množina S (a tedy i množina M) nesplnitelná, v opačném případě je M splnitelná.

Je zřejmé, že konstrukce celé množiny $R^\star(S)$ může být zbytečná — stačí pouze zjistit, zda $R^\star(S)$ obsahuje F.

3.10.8 Výhodnější postup. Existuje ještě jeden postup, který usnadní práci s použitím rezoluční metody. Ten nejenom že nám odpoví na otázku, zda konečná množina klausulí S je splnitelná nebo nesplnitelná, ale dokonce nám umožní v případě splnitelnosti sestrojit aspoň jedno pravdivostní ohodnocení, v němž je množina S pravdivá.

Máme konečnou množinu klausulí S, kde žádná klausule není tautologií. Zvolíme jednu logickou proměnnou (označme ji x), která se v některé z klausulí z S vyskytuje. Najdeme množinu klausulí S_1 s těmito vlastnostmi:

1. Žádná klausule v S_1 neobsahuje logickou proměnnou x.
2. Množina S_1 je splnitelná právě tehdy, když je splnitelná původní množina S.

Množinu S_1 vytvoříme takto: Rozdělíme klausule množiny S do tří skupin:

M_0 se skládá ze všech klausulí množiny S, které neobsahují logickou proměnnou x.

M_x se skládá ze všech klausulí množiny S, které obsahují positivní literál x.

$M_{\neg x}$ se skládá ze všech klausulí množiny S, které obsahují negativní literál $\neg x$.

Označme N množinu všech rezolvent klausulí množiny S podle literálu x; tj. rezolvent vždy jedné klausule z množiny M_x s jednou klausulí z množiny $M_{\neg x}$. Všechny tautologie vyřadíme.

Položíme $S_1 = M_0 \cup N$.

3.10.9 Tvrzení. Množina klausulí S_1 zkonstruovaná výše je splnitelná právě tehdy, když je splnitelná množina S.

3.10.10 Dostali jsme tedy množinu klausulí S_1, která již neobsahuje logickou proměnnou x a je splnitelná právě tehdy, když je splnitelná množina S. Navíc, množina S_1 má o jednu logickou proměnnou méně než množina S.

Nyní opakujeme postup pro množinu S_1. Postup skončí jedním ze dvou možných způsobů:

1. Při vytváření rezolvent dostaneme prázdnou kalusuli F. Tedy S je nesplnitelná.
2. Dostaneme prázdnou množinu klausulí. V tomto případě je množina S splnitelná.

3.10.11 Je výhodné předchozí postup znázorňovat v tabulce. Na začátku práce utvoříme tabulku, která obsahuje pro každou klausuli množiny S jeden sloupec. V prvním řádku vybereme jednu proměnnou, řekněme x, a řádek označíme proměnnou x. Procházíme neoznačené sloupce tabulky, které odpovídají klausulím obsahujícím proměnnou x. Ve sloupci do řádku napíšeme 1, v případě, že klasusule obsahuje literál x, nebo 0, v případě, že klausule obsahuje literál $\neg x$.

Vybereme libovolnou klausuli C_1, která má v řádku 1, a libovolnou klausuli C_2, která má v řádku 0. Sloupec pro jejich rezolventu podle x přidáme v případě, že se jedná o novou klausuli, která není tautologií. Jestliže žádný sloupec není v řádku označen 1 ($M_x = \emptyset$) nebo žádný sloupec není v řádku označen 0 ($M_{\neg x} = \emptyset$), nepřidáváme nic.

Jestliže jsme přidali prázdnou klausuli, výpočet končí, množina S je nesplnitelná. Jestliže každý sloupec již má 1 nebo 0, výpočet ukončíme, množina S je splnitelná. Tím jsme ukončili první krok.

Ve druhém kroku se zajímáme jen o sloupce tabulky, které nemají ještě ani číslo 1 ani 0 (tyto sloupce tvoří množiny S_1). Opět vybereme proměnnou, která se v některé ze zbylých klausulí vyskytuje. Postupujeme dále jako v kroku 1.

Celý postup tedy končí buď přidáním prázdné klausule, v tom případě je množina S nesplnitelná, nebo vyčerpáním neoznačených sloupců, v tomto případě je množina S splnitelná.

3.10.12 Příklad. Rezoluční metodou rozhodněte, zda množina klausulí

$$S = \{x \vee y \vee \neg z, \neg x, x \vee y \vee z, x \vee \neg y, z \vee t \vee v, \neg t \vee w\}$$

je splnitelná. V kladném případě najděme pravdivostní ohodnocení, v němž je S pravdivá.

Vyjdeme z tabulky, která má jeden sloupec pro každou klausuli množiny S. (Sledujte tabulku 3.1.)

	$x \vee y \vee \neg z$	$\neg x$	$x \vee y \vee z$	$x \vee \neg y$	$z \vee t \vee v$	$\neg t \vee w$					
y:	1		1	0			$x \vee \neg z$	$x \vee z$			
x:		0					1	1	$\neg z$	z	
z:					1				0	1	F

Tabulka 3.1: Tabulka pro rezoluční metodu

Nejprve odstraňujeme logickou proměnnou y: První řádek označíme y. Do sloupce napíšeme 1 v případě, že jeho klausule obsahuje literál y (první a třetí sloupec), a napíšeme 0 v případě, že klausule sloupce obsahuje literál $\neg y$ (čtvrtý sloupec). Tím jsme označili všechny sloupce, jejichž klausule obsahují proměnnou y. K tabulce přidáme rezolventy klausulí podle literálu y: Jsou to klausule

$$x \vee \neg z = res_y(x \vee y \vee \neg z, x \vee \neg y) \quad \text{a} \quad x \vee z = res_y(x \vee y \vee z, x \vee \neg y).$$

Množina S_1 z našeho postupu je nyní tvořena všemi klausulemi, jejichž sloupce ještě nejsou označeny 0 nebo 1. Tedy

$$S_1 = \{\neg x, z \vee t \vee v, \neg t \vee w, x \vee \neg z, x \vee z\}.$$

V dalším kroku odstraníme logickou proměnnou x: V řádku odpovídajícím proměnné x napíšeme 0 do druhého sloupce (klausule $\neg x$) a napíšeme 1 do sedmého a osmého sloupce (klausule $x \vee \neg z$ a $x \vee z$). K tabulce přidáme sloupce pro rezolventy klausulí množiny S_1 podle literálu x. Jsou to $\neg z = res_x(\neg x, x \vee \vee \neg z)$ a $z = res_x(\neg x, x \vee z)$. Nyní stačí rozhodnout, zda je splnitelná množina klausulí

$$S_2 = \{z \vee t \vee v, \neg t \vee w, \neg z, z\}.$$

Dále vybereme logickou proměnnou z. Do pátého a desátého sloupce vepíšeme 1 (jejich klausule obsahují literál z) a do devátého sloupce napíšeme 0 (jeho klausule obsahuje literál $\neg z$). Jako rezolventu klausulí z devátého a desátého sloupce dostáváme prázdnou klausuli F. Proto je množina S_2 nesplnitelná. To znamená, že také množiny S_1 a S jsou nesplnitelné.

3.10.13 Příklad. Rezoluční metodou rozhodněme, zda množina klausulí

$$S = \{a \vee \neg d, \neg b \vee \neg c, b \vee d, \neg b \vee \neg e, a \vee c \vee d, \neg a \vee \neg d\}$$

je splnitelná. V kladném případě najděme pravdivostní ohodnocení, ve kterém je množina S pravdivá.

Postupujeme obdobně jako v předcházejícím příkladě. Sledujte tabulku 3.2. Postupujme trochu rychleji.

	$a \vee \neg d$	$\neg b \vee \neg c$	$b \vee d$	$\neg b \vee \neg e$	$a \vee c \vee d$	$\neg a \vee \neg d$			
e:				0					
c:		0			1		$a \vee \neg b \vee d$		
b:			1				0	$a \vee d$	
d:	0					0		1	a
a:									1

Tabulka 3.2: Konstrukce rezoluční tabulky shora dolů

První řádek odpovídá logické proměnné e. Protože tuto proměnnou obsahuje jen klausule $\neg b \vee \neg e$, řádek obsahuje pouze jedinou 0. Žádnou rezolventu podle proměnné e nelze utvořit (a nepřidáváme proto žádný sloupec).

Druhý řádek odpovídá logické proměnné c. V řádku máme jednu 0 (v druhém sloupci) a jednu 1 (v pátém sloupci). K tabulce přidáme rezolventu $a \vee \neg b \vee d$.

Třetí řádek odpovídá logické proměnné b. Máme dva sloupce, jejichž klausule obsahuje logickou proměnnou b a které ještě nebyly označeny. Jsou to třetí a sedmý sloupec. Do třetího sloupce píšeme 1, do sedmého 0. Přidáme k tabulce nový sloupec odpovídající rezolventě $a \vee d$.

Čtvrtý řádek odpovídá logické proměnné d. Všechny dosud nevyplněné sloupce odpovídají klausulím, které proměnnou d obsahují. Do prvního a šestého sloupce vepíšeme 0 a do osmého sloupce 1. Máme dvě rezolventy podle proměnné d a to klausuli a (rezolventa klausulí z prvního a osmého sloupce) a klausuli $a \vee \vee \neg a$ (z šestého a osmého sloupce). Druhá klausule je tautologie, takže ji dále

neuvažujeme a její sloupec k tabulce nepřidáváme. K tabulce přidáme pouze sloupec pro klausuli a.

Poslední logická proměnná je a. Zbývá pouze jediný nevyplněný sloupec, a to poslední; napíšeme do něj 1.

Tím jsme vyčerpali všechny logické proměnné a dostali jsme prázdnou množinu klausulí. Ta je jistě splnitelná a proto je splnitelná i celá množina S.

Zkonstruujeme zpětně pravdivostní ohodnocení u, v němž je množina S pravdivá. Sledujte tabulku 3.3.

	$a \vee \neg d$	$\neg b \vee \neg c$	$b \vee d$	$\neg b \vee \neg e$	$a \vee c \vee d$	$\neg a \vee \neg d$			
e:				0					
c:		0			1		$a \vee \neg b \vee d$		
b:			1				0	$a \vee d$	
d:	0					0		1	a
a:									1
	$\uparrow_1$	$\uparrow_4$	$\uparrow_3$	$\uparrow_5$	$\uparrow_1$	$\uparrow_2$	$\uparrow_1$	$\uparrow_1$	$\uparrow_1$

Tabulka 3.3: Konstrukce splňujícího ohodnocení — četba tabulky zdola nahoru

Začneme od posledního řádku naší tabulky. Protože tento řádek obsahuje 1, položíme $u(a) = 1$. Vyznačíme šipkou v tabulce všechny sloupce, které odpovídají klausuli s pozitivním literálem a. Jedná se o první, pátý, sedmý, osmý a devátý sloupec. Všechny jejich klausule jsou pravdivé v u nezávisle na tom, jaké hodnoty bude mít ohodnocení u pro ostatní logické proměnné. K šipkám označujícím sloupce připisujeme pro přehlednost i číslo, které říká, v kterém kroku jsme sloupec označili.

Přejdeme na předcházející řádek. Ve sloupcích, které ještě nebyly označeny šipkou, se vyskytuje pouze 0, a to v šestém sloupci. Položíme proto $u(d) = 0$ a opět označíme všechny dosud neoznačené sloupce, jejichž klausule se touto volbou stanou pravdivé. V našem případě je to pouze šestý sloupec.

V řádku odpovídajícím logické proměnné b máme 1 (sedmý sloupec, který obsahuje 0, již byl označem — jeho klausule je pravdivá, protože obsahuje positivní literál a). Položíme proto $u(b) = 1$ a označíme třetí sloupec.

V řádku odpovídajícím logické proměnné c máme neoznačený druhý sloupec, kde je 0. Položíme proto $u(c) = 0$. Touto volbou se stane pravdivou i klausule odpovídající tomuto sloupci.

Ze všech sloupců nyní pouze čtvrtý sloupec není označen. V řádku odpovídajícím logické proměnné e má 0, proto položíme $u(e) = 0$ a označíme i tento sloupec.

Není obtížné se přesvědčit, že v pravdivostním ohodnocení u definovaném: $u(a) = u(b) = 1$, $u(c) = u(d) = u(e) = 0$, je množina S pravdivá.

Kapitola 4

Predikátová logika

Výroková logika nezahrnuje všechny úsudky, které považujeme za správné. Uvažujme např. úsudek:

Petr hraje na housle.
Každý, kdo hraje na housle, má hudební sluch.
Petr má hudební sluch.

Jeho správnost žádným způsobem z výrokové logiky nevyplývá. První věta totiž ve výrokové logice představuje elementární výrok p. Druhá věta má tvar implikace dvou elementárních výroků: $q \Rightarrow t$, kde q je výrok „Každý kdo hraje na housle.“ a t je výrok „Má hudební sluch.“. Poslední věta je elementární výrok r, kde r je výrok „Petr má hudební sluch.“. Ve výrokové logice sémantický důsledek

$$\{p, q \Rightarrow t\} \models r$$

není správný. Správnost úsudku spočívá ve vnitřní struktuře jednotlivých elementárních výroků, a tuto strukturu výroková logika není schopna popsat. Ve výrokové logice jsou totiž elementární výroky dále neanalyzovatelné atomy formálního jazyka.

Abychom předchozí úsudek popsali, musíme uvážit vnitřní strukturu jednotlivých vět. Teprve na základě struktury těchto výroků budeme schopni rozhodnout, který úsudek je správný a který nikoli. A tím se zabývá predikátová logika.

4.1 Neformální zavedení predikátové logiky

Zamysleme se nad tím, co potřebujeme k popsání výroku „Petr hraje na housle.“. Výrok se týká „Petra“ jakožto „objektu“ a vlastnosti, kterou Petr má, tj. vlastnosti „hrát na housle“. Obdobnou strukturu má i třetí věta „Petr má hudební sluch.“. Vlastností tady je „mít hudební sluch“. Prostřední věta navíc mluví o „každém“ chápáno jako každém objektu (jednotlivci). Má opravdu alespoň zčásti strukturu implikace: Každý objekt, který má vlastnost „hrát na housle“, má i vlastnost „mít hudební sluch“.

Označíme H vlastnost „hrát na housle“ a S vlastnost „mít hudební sluch“. Nyní lze náš úsudek napsat ve zkrácenější podobě takto:

Petr má H.
Každý, kdo má H, má i S.

Petr má S.

I to je však zbytečně dlouhý zápis. Pišme místo „Petr má H." zkráceně $H(p)$, kde p označuje Petra. Obdobně zkrátíme zápis třetí věty na $S(p)$. Abychom obdobným způsobem zkrátili i prostřední větu, zavedeme zkratku za „každý", též „pro všechny". Tuto zkratku označíme symbolem $\forall$ a nazveme ji *obecným* (neboli *universálním*) *kvantifikátorem*. Kvantifikátor je vždy následován proměnnou. Formuli $\forall x$ čteme „pro každé x". Nejde nám totiž o to zachytit jen „každý", ale „každý objekt". Proměnné značíme $x, y, \ldots$, a zastupují nám objekty v případě, že objekty nemáme konkrétně zadané. Prostřední věta bude mít tvar: $\forall x\,(H(x) \Rightarrow S(x))$. Tedy celý úsudek zapíšeme následovně:

$H(p)$
$\forall x\,(H(x) \Rightarrow S(x))$

$S(p)$

Vlastnostem či vztahům budeme říkat *predikáty*, vyznačeným objektům *konstanty*.

4.1.1 Poznámka. Obecný kvantifikátor „$\forall x$" je vlastně zobecnění konjunkce, jakési její „nekonečné rozšíření".

4.1.2 Uvažujme ještě jeden úsudek:

Petr hraje na housle.
Petr má hudební sluch.

Někdo, kdo hraje na housle, má i hudební sluch.

Přitom poslední větu upřesníme: Někdo (ve smyslu existuje aspoň jeden objekt), kdo hraje na housle, má hudební sluch. Formalizujme tento úsudek. První dvě věty už máme zformalizovány: $H(p)$ a $S(p)$. K formalizaci třetí věty potřebujeme symbol pro „aspoň jeden", „někdo" atd. Tímto symbolem je *existenční kvantifikátor* $\exists$ opět následovaný proměnnou. Symbol $\exists x$ čteme „existuje x tak, že...". Poslední věta říká, že někdo (aspoň jeden) objekt má vlastnost „hrát na housle" a současně i vlastnost „mít hudební sluch". Proto její fomalizace je $\exists x\,(H(x) \wedge S(x))$. Celý úsudek má tvar:

$H(p)$
$S(p)$

$\exists x\,(H(x) \wedge S(x))$

4.1.3 Poznámka. Existenční kvantifikátor „$\exists x$" je vlastně zobecnění disjunkce, jakési její „nekonečné rozšíření".

4.1.4 Uvedeme ještě jeden úsudek, který je typický pro predikátovou logiku.

Je-li přirozené číslo sudé, pak jeho následník je číslo liché.
Číslo 2 je sudé.

Následník čísla 2 je číslo liché.

Vlastnosti (predikáty), které nás zajímají v tomto úsudku jsou: „být sudým přirozeným číslem", označíme ji S, a „být lichým přirozeným číslem", označíme ji L. Dále máme jeden objekt (konstantu), a to číslo 2. Pak je tu ještě „následník přirozeného čísla". Tady následníka nechápeme jako vlastnost; neptáme se, zda nějaký objekt je následník jiného nebo ne. Tady s následníkem přirozeného čísla

pracujeme jako s přirozeným číslem, tj. jako s objektem. Ovšem tento objekt je zadán nepřímo — pomocí čísla, které mu předchází. Budeme se proto na „následníka“ dívat jako na funkci, označíme ji f, která každému přirozenému číslu přiřadí opět přirozené číslo a to $f\colon n \mapsto n+1$.

Nyní již můžeme zformalizovat druhou a třetí větu: $S(2)$ a $L(f(2))$. První věta bude obsahovat kvantifikátor. Z české formulace na první pohled není úplně jasné který. Tato nepřesnost ve vyjádření, zda náš výrok má platit pro všechny objekty nebo jen pro některé z nich, je vlastní téměř všem přirozeným jazykům. Přeformulujeme proto větu „trochu jiným způsobem“ tak, aby byla kvantifikace jasnější. V našem případě má první věta stejný smysl jako tvrzení „Kdykoli je číslo sudé, pak jeho následník je číslo liché.“ Proto formalizace první věty má tvar: $\forall x\,(S(x) \Rightarrow L(f(x)))$. Celý úsudek je

$$\begin{array}{l} \forall x\,(S(x) \Rightarrow L(f(x))) \\ S(2) \\ \hline L(f(2)) \end{array}$$

4.1.5 Další příklady. Zformalizujme následující věty. Vždy uvedeme, které vlastnosti (predikáty), konstanty a funkce používáme.

a) Petrův otec je varhaník.

b) Něčí otec je varhaník.

c) Každý čtverec reálného čísla je nezáporný.

d) Každé celé číslo má předchůdce.

e) Je-li přirozené číslo větší než 0, je jeho následník větší než 1.

4.1.6 Řešení.

a) Naše objekty jsou lidé. Dále potřebujeme jeden predikát (vlastnost) $V(\_)$, a to vlastnost „být varhaníkem“. Rovněž potřebujeme funkci o, která každému člověku přiřadí jeho otce a nakonec potřebujeme konstantu p pro Petra. Formule má tvar:

$$V(o(p)).$$

b) Tady si vystačíme s predikátem a funkcí z minulého příkladu; „něčí“ popíšeme existenčním kvantifikátorem:

$$\exists x\, V(o(x)).$$

c) Objekty jsou reálná čísla. Potřebujeme jeden predikát $K(\_)$, který označuje vlastnost „být nezáporné číslo“, a funkci f, která každému reálnému číslu přiřadí jeho čtverec, tj. $f\colon x \mapsto x^2$. Formule má tvar

$$\forall x\, K(f(x)).$$

d) Objekty jsou celá čísla. Pro formalizaci této věty potřebujeme vztah mezi dvěma celými čísly. (Vztahu, který se týká dvou objektů, budeme říkat *binární predikát*.) Dvě čísla a, b jsou ve vztahu Q, jestliže první z nich, a,

je předchůdcem čísla druhého, b. Fakt, že dvojice čísel a, b je ve vztahu Q zapíšeme $Q(a, b)$. Věta „Každé celé číslo má předchůdce.“ odpovídá významem větě „Pro každé celé číslo existuje jeho předchůdce.“ a můžeme ji zformalizovat:

$$\forall x\, \exists y\, Q(y, x).$$

e) Objekty jsou přirozená čísla. Mohli bychom sice zavést dva predikáty: Jeden, který znamená vlastnost být větší než 0, a druhý, který znamená vlastnost být větší než 1, ale srozumitelnosti formalizace bychom tím neprospěli. Zavedeme raději jeden binární predikát a to „být větší než“. Tato vlastnost je dobře známá a značí se symbolem „$>$“. Aby naše formalizace byla co nejsrozumitelnější v případě predikátu „být větší“ zvolíme zápis $a > b$ místo přesného leč nesmyslně těžkopádného zápisu $>(a, b)$. Použijeme dvě konstanty a to číslo 0 a číslo 1 a funkci následníka f. Formule má tvar:

$$\forall x\, ((x > 0) \Rightarrow (f(x) > 1)).$$

4.2 Syntaxe predikátové logiky

V tomto oddíle zavedeme syntaxi predikátové logiky, tj. uvedeme pravidla, podle nichž se tvoří syntakticky správné formule predikátové logiky. Význam a pravdivostní hodnota nás bude zajímat až dále.

Správně utvořené formule budou posloupnosti symbolů tzv. *jazyka predikátové logiky.*

4.2.1 Jazyk predikátové logiky $\mathscr{L}$. *Jazyk predikátové logiky* se skládá z

1. *logických symbolů*, tj.:
   a) spočetné množiny individuálních proměnných: $\mathsf{Var} = \{x, y, \ldots, x_1, x_2, \ldots\}$
   b) výrokových logických spojek: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
   c) obecného kvantifikátoru $\forall$ a existenčního kvantifikátoru $\exists$
2. *speciálních symbolů*, tj.:
   a) množiny Pred predikátových symbolů (nesmí být prázdná)
   b) množiny Kons konstantních symbolů (může být prázdná)
   c) množiny Func funkčních symbolů (může být prázdná)
3. pomocných symbolů, jako jsou závorky „(, [,) ,]“ a čárka „,“.

Pro každý predikátový i funkční symbol máme dáno přirozené číslo n větší nebo rovné 1, které nám udává, kolika objektů se daný predikát týká, nebo kolika proměnných je daný funkční symbol. Tomuto číslu říkáme *četnost* nebo též *arita* predikátového symbolu nebo funkčního symbolu.

4.2.2 Poznámka. Predikátové symboly budeme většinou značit velkými písmeny, tj. např. $P, Q, R, \ldots, P_1, P_2, \ldots$; konstantní symboly malými písmeny ze začátku abecedy, tj. $a, b, c, \ldots, a_1, \ldots$, a funkční symboly většinou $f, g, h, \ldots, f_1, f_2, \ldots$. Formule predikátové logiky budeme označovat malými řeckými písmeny (obdobně, jako jsme to dělali pro výrokové formule). Kdykoli se od těchto konvencí odchýlíme, tak v textu na to upozorníme.

Poznamejme, že přestože často budeme mluvit o n-árních predikátových symbolech a n-árních funkčních symbolech, v běžné praxi se setkáme jak s predikáty, tak funkcemi arity nejvýše tři. Nejběžnější jsou predikáty a funkční symboly arity 1, těm říkáme též *unární*, nebo arity 2, těm říkáme též *binární*. Doporučujeme čtenáři, aby se na tomto místě vrátil k příkladům z minulého oddílu a pro každý predikát i funkci určil, jakou má aritu.

Poznamenejme ještě, že někteří autoři konstantní symboly zahrnují pod nulární funkční symboly (tj. funkční symboly arity 0).

4.2.3 Termy. Množina *termů* je definována těmito pravidly:

1. Každá proměnná a každý konstantní symbol je term.
2. Jestliže f je funkční symbol arity n a $t_1, t_2, \ldots, t_n$ jsou termy, pak $f(t_1, t_2, \ldots, t_n)$ je také term.
3. Nic, co nevzniklo konečným použitím pravidel 1 a 2, není term.

4.2.4 Poznámka. Term je zhruba řečeno objekt, pouze může být složitěji popsán než jen proměnnou nebo konstantou. V jazyce predikátové logiky termy vystupují jako „podstatná jména“.

4.2.5 Atomické formule. *Atomická formule* je predikátový symbol P aplikovaný na tolik termů, kolik je jeho arita. Jinými slovy, pro každý predikátový symbol $P \in \mathsf{Pred}$ arity n a pro každou n-tici termů $t_1, t_2, \ldots, t_n$ je $P(t_1, t_2, \ldots, t_n)$ atomická formule.

4.2.6 Formule. Množina *formulí* je definována těmito pravidly:

1. Každá atomická formule je formule.

2. Jsou-li φ a ψ dvě formule, pak $(\neg\varphi)$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \Rightarrow \psi)$, $(\varphi \Leftrightarrow \psi)$ jsou opět formule.

3. Je-li φ formule a x proměnná, pak $(\forall x\, \varphi)$ a $(\exists x\, \varphi)$ jsou opět formule.

4. Nic, co nevzniklo pomocí konečně mnoha použití bodů 1 až 3, není formule.

4.2.7 Poznámka. Formule predikátové logiky jsme definovali obdobně jako výrokové formule: Nejprve jsme definovali „ty nejjednodušší“ formule (atomické formule) a potom pomocí logických spojek a kvatifikátorů konstruujeme složitější formule. Ve výrokové logice byl první krok daleko jednodušší, protože elementární výroky nebyly strukturované. Vlastní konstrukce formulí je však v obou případech podobná.

4.2.8 Konvence.

1. Úplně vnější závorky nepíšeme. Píšeme tedy např. $(\exists x P(x)) \vee R(a, b)$ místo $((\exists x\, P(x)) \vee R(a, b))$.

2. Spojka „negace“ má vždy přednost před výrokovými logickými spojkami a proto píšeme např. $\forall x\, (\neg P(x) \Rightarrow Q(x))$ místo $\forall x\, ((\neg P(x)) \Rightarrow Q(x))$.

4.2.9 Derivační strom formule. Ke každé formuli predikátové logiky můžeme přiřadit její *derivační strom* podobným způsobem jako jsme to udělali v případě výrokových formulí. Rozdíl je v tom, že kvantifikátory považujeme za unární (tj. mají pouze jednoho následníka) a také pro termy vytváříme jejich derivační strom. Listy derivačního stromu jsou vždy ohodnoceny buď proměnnou nebo konstantou. Poznamenejme, že derivačnímu stromu se též říká *syntaktický strom*.

4.2.10 Podformule. *Podformule* formule φ je libovolný podřetězec φ, který je sám formulí. Jinými slovy: Podformule formule φ je každý řetězec odpovídající podstromu derivačního stromu formule φ, určeného vrcholem ohodnoceným predikátovým symbolem, logickou spojkou nebo kvantifikátorem.

4.2.11 Volný a vázaný výskyt proměnné. Máme formuli φ a její derivační strom. List derivačního stromu obsazený proměnnou x je výskyt proměnné x ve formuli φ. Výskyt proměnné x je *vázaný* ve formuli φ, jestliže při postupu od listu ohodnoceného tímto x ve směru ke kořeni derivačního stromu narazíme na kvantifikátor s touto proměnnou. V opačném případě mluvíme o *volném* výskytu proměnné x.

4.2.12 Sentence. Formule, která má pouze vázané výskyty proměnné, se nazývá *sentence*, též *uzavřená formule*. Formuli, která má pouze volné výskyty proměnné, se říká *otevřená formule*.

4.2.13 Legální přejmenování proměnné. Přejmenování výskytů proměnné x ve formuli φ je *legálním* přejmenováním proměnné, jestliže

- jedná se o výskyt vázané proměnné ve φ;
- přejmenováváme všechny výskyty x vázané daným kvantifikátorem;
- po přejmenování se žádný dříve volný výskyt proměnné nesmí stát vázaným výskytem.

4.2.14 Rovnost formulí. Dvě formule považujeme za *stejné*, jestliže se liší pouze legálním přejmenováním vázaných proměnných.

Každou formuli φ lze napsat tak, že každá proměnná má ve formuli buď jen volné výskyty nebo jen vázané výskyty.

4.3 Sémantika predikátové logiky

Nyní se budeme zabývat sémantikou formulí, tj. jejich významem a pravdivostí.

4.3.1 Interpretace jazyka predikátové logiky. *Interpretace* predikátové logiky s predikátovými symboly Pred, konstantními symboly Kons a funkčními symboly Func je dvojice $\langle U, [\![-]\!]\rangle$, kde

- U je neprázdná množina nazývaná *universum*;
- $[\![-]\!]$ je přiřazení, které
  1. každému predikátovému symbolu $P \in \mathsf{Pred}$ arity n přiřazuje podmnožinu $[\![P]\!]$ množiny U^n, tj. n-ární relaci na množině U.
  2. každému konstantnímu symbolu $a \in \mathsf{Kons}$ přiřazuje prvek z U, značíme jej $[\![a]\!]$,
  3. každému funkčnímu symbolu $f \in \mathsf{Func}$ arity n přiřazuje zobrazení množiny U^n do U, značíme je $[\![f]\!]$,

Množina U se někdy nazývá *domain* a označuje D.

4.3.2 Kontext proměnných. Je dána interpretace $\langle U, [\![-]\!]\rangle$. *Kontext proměnných* je zobrazení ρ, které každé proměnné $x \in \mathsf{Var}$ přiřadí prvek $\rho(x) \in U$. Je-li ρ kontext proměnných, $x \in \mathsf{Var}$ a $d \in U$, pak

$$\rho[x := d]$$

označuje kontext proěnných, který má stejné hodnoty jako ρ pouze v proměnné x má hodnotu d. Kontextu proměnných $\rho[x := d]$ říkáme *update* kontextu ρ o hodnotu d v x.

4.3.3 Interpretace termů při daném kontextu proměnných. Je dána interpretace $\langle U, [\![-]\!]\rangle$ a kontext proměnných ρ. Pak termy interpretujeme následujícím způsobem.

1. Je-li term konstatní symbol $a \in \mathsf{Kons}$, pak jeho hodnota je prvek $[\![a]\!]_\rho = [\![a]\!]$. Je-li term proměnná x, pak jeho hodnota je $[\![x]\!]_\rho = \rho(x)$.

2. Je-li $f(t_1, \ldots, t_n)$ term. pak jeho hodnota je

   $$[\![f(d_1, \ldots, d_n)]\!]_\rho = [\![f]\!]([\![t_1]\!]_\rho, \ldots, [\![t_n]\!]_\rho).$$

   [Jinými slovy, hodnota termu $f(t_1, \ldots, t_n)$ je funkční hodnota funkce $[\![f]\!]$ provedené na n-tici prvků $[\![t_1]\!]_\rho, \ldots, [\![t_n]\!]_\rho$ z U.]

Poznamenejme, že neobsahuje-li term t proměnnou, pak jeho hodnota nezáleží na kontextu proměnných ρ, ale pouze na interpretaci.

Tuto formální definici si můžete přiblížit ještě takto. Vezmeme term t a utvoříme jeho derivační strom. Listy stromu ohodnotíme tak, jak nám říká interpretace (pro konstantní symboly) a kontext proměnných. Pak jdeme v derivačním stromu směrem ke kořeni. Vrchol, který odpovídá n-árnímu funkčnímu symbolu f a má následníky ohodnoceny prvky d_1, d_2, ..., d_n (v tomto pořadí zleva doprava), ohodnotíme prvkem $[\![f]\!](d_1, \ldots, d_n)$, tj. obrazem n-tice $(d_1, \ldots, d_n)$ v zobrazení $[\![f]\!]$. Prvek, kterým je ohodnocen kořen, je hodnota celého termu v dané interpretaci a daném kontextu. Uvědomte si, že se jedná o přesně stejný postup jako např. při vyhodnocování algebraických výrazů.

4.3.4 Pravdivostní hodnota formule v dané interpretaci a daném kontextu. Nejprve definujeme *pravdivost formulí v dané interpretaci* $\langle U, [\![-]\!]\rangle$ při daném kontextu proměnných ρ:

1. Nechť φ je atomická formule. Tj. $\varphi = P(t_1, \ldots, t_n)$, kde P je predikátový symbol arity n a $t_1, \ldots, t_n$ jsou termy. Pak φ je pravdivá v interpretaci $\langle U, [\![-]\!]\rangle$ a kontextu ρ právě tehdy, když

   $$([\![t_1]\!]_\rho, \ldots, [\![t_n]\!]_\rho) \in [\![P]\!].$$

   Jinými slovy: φ je v naší interpretaci pravdivá právě tehdy, když n-tice hodnot termů $([\![t_1]\!]_\rho, \ldots, [\![t_n]\!]_\rho)$ má vlastnost $[\![P]\!]$.

2. Jsou-li φ a ψ formule, jejichž pravdivost v interpretaci $\langle U, [\![-]\!]\rangle$ a kontextu ρ již známe, pak

   - $\neg\varphi$ je pravdivá právě tehdy, když φ není pravdivá.

- $\varphi \wedge \psi$ je pravdivá právě tehdy, když φ i ψ jsou pravdivé.
- $\varphi \vee \psi$ je nepravdivá právě tehdy, když φ i ψ jsou nepravdivé.
- $\varphi \Rightarrow \psi$ je nepravdivá právě tehdy, když φ je pravdivá a ψ je nepravdivá.
- $\varphi \Leftrightarrow \psi$ je pravdivá právě tehdy, když buď obě formule φ a ψ jsou pravdivé, nebo obě formule φ a ψ jsou nepravdivé.

3. Je-li φ formule a x proměnná, pak

- $\forall x \varphi(x)$ je pravdivá právě tehdy, když fromule φ je pravdivá *v každém* kontextu $\rho[x := d]$, kde d je prvek U.
- $\exists x \, \varphi(x)$ je pravdivá právě tehdy, když fromule φ je pravdivá *v aspoň jednom* kontextu $\rho[x := d]$, kde d je prvek U..

4.3.5 Pravdivostní hodnota sentence. Sentence φ je *pravdivá v interpretaci* $\langle U, [\![-]\!] \rangle$ právě tehdy, když je pravdivá v každém kontextu proměnných ρ.

Poznamenejme, že pro sentence v předchozí definici jsme mohli požadovat pravdivost v alespoň jednom kontextu.

4.3.6 Příklad. Je dán jazyk predikátové logiky $\mathscr{L}$, kde $\mathsf{Pred} = \{P, Q\}$, $\mathsf{Func}\{f, g\}$ a $\mathsf{Kons} = \{a, b, c\}$; P a f jsou unární, Q a g jsou binární. Uvažujme interpretaci $\langle U, [\![-]\!]\rangle$ definovanou takto:

- $U = \mathbb{N}$;
- 1. $[\![a]\!] = 0$, $[\![b]\!] = 1$,
  2. $[\![f]\!]$ je následník, tj. $[\![f]\!](n) = n + 1$,
     $[\![g]\!]$ je součet, tj. $[\![g]\!](m, n) = m + n$;
  3. $[\![P]\!]$ je množina sudých čísel,
     $[\![Q]\!]$ je množina všech dvojic přirozených čísel (m, n), kde m je menší nebo rovno n, (tj. $[\![Q]\!]$ je relace $\leq$ na množině $\mathbb{N}$).

Rozhodněte o pravdivosti následujících sentencí

a) $P(g(f(a), g(a, f(b))))$,

b) $Q(g(a, b), f(b))$,

c) $P(b) \Rightarrow (\forall x\, P(x))$,

d) $\exists x\, \forall y\, Q(y, x)$,

e) $\forall y\, \exists x\, Q(y, x)$.

4.3.7 Řešení.

a) Nejprve musíme ohodnotit term $g(f(a), g(a, f(b)))$: Máme

$$[\![g(f(a), g(a, f(b)))]\!] = [\![f(a)]\!] + [\![g(a, f(b))]\!] = ([\![a]\!] + 1) + ([\![a]\!] + [\![f(b)]\!]) =$$

$$= ([\![a]\!] + 1) + ([\![a]\!] + ([\![b]\!] + 1)) = (0 + 1) + (0 + (1 + 1)) = 1 + (1 + 1) = 3.$$

Formule bude pravdivá, jestliže číslo 3 je sudé. Ale číslo 3 není sudé, tedy celá formule je nepravdivá v interpretaci $\langle U, [\![-]\!]\rangle$.

b) Postupujeme obdobně jako v předchozím případě: Nejprve interpretujeme termy: Zřejmě platí $[\![g(a, b)]\!] = 0 + 1 = 1$ a $[\![f(b)]\!] = 1 + 1 = 2$. Dvojice $(1, 2)$ je prvkem $[\![Q]\!]$, neboť číslo 1 je menší než číslo 2 a tedy naše formule je pravdivá v $\langle U, [\![-]\!]\rangle$.

c) V tomto případě zjistíme pravdivost velmi lehce. Formule $P(b)$ je nepravdivá v interpretaci $\langle U, [\![-]\!]\rangle$, protože $[\![b]\!] = 1$ a to není sudé. Proto implikace $P(b) \Rightarrow \psi$ je pravdivá pro jakoukoli sentenci, tedy i pro $\psi = \forall x\, P(x)$.

d) Nejprve si naši formuli pro danou interpretaci „přečteme“. Formule říká: Existuje přirozené číslo x tak, že pro všechna přirozená čísla y je y menší nebo rovno x. Jinými slovy: Existuje přirozené číslo x, které je větší nebo rovno všem přirozeným číslům. A to zřejmě není pravda, neboť pro každé přirozené číslo m najdeme přirozené číslo větší, např $m + 1$.

e) Opět si pomůžeme tím, že danou formuli v interpretaci „přečteme“: Ke každému přirozenému číslu y existuje přirozené číslo x tak, že y je menší nebo rovno x. Tato formule je pravdivá v dané interpretaci: pro číslo $d \in \mathbb{N}$ (které dosadíme za y) stačí zvolit číslo $d+1$ (to dosadíme za x) a opravdu máme d menší nebo rovno $d+1$. Tedy naše formule je pravdivá v $\langle U, [\![-]\!]\rangle$.

4.3.8 Model sentence. Interpretace $\langle U, [\![-]\!]\rangle$, ve které je sentence φ pravdivá, se nazývá *model sentence* φ.

4.3.9 Tautologie, kontradikce, splnitelná sentence. Sentence φ se nazývá *tautologie*, jestliže je pravdivá v každé interpretaci. Sentence se nazývá *kontradikce*, jestliže je nepravdivá v každé interpretaci. Nazývá se *splnitelná*, jestliže je pravdivá v aspoň jedné interpretaci.

Také jsme mohli formulovat předchozí definice pomocí pojmu „model“. Tautologie je sentence, pro kterou je každá interpretace jejím modelem; sentence je splnitelná, má-li model; sentence je kontradikce, nemá-li model.

4.3.10 Následující sentence jsou tautologie. (P je unární predikátový symbol, Q je binární predikátový symbol a a je konstantní symbol.)

1. $(\forall x\, P(x)) \Rightarrow P(a)$;
2. $P(a) \Rightarrow (\exists x\, P(x))$;
3. $\neg(\forall x\, P(x)) \Leftrightarrow (\exists x\, \neg P(x))$;
4. $\neg(\exists x\, P(x)) \Leftrightarrow (\forall x\, \neg P(x))$;
5. $(\forall x\, \forall y\, Q(x,y)) \Leftrightarrow (\forall y\, \forall x\, Q(x,y))$;
6. $(\exists x\, \exists y\, Q(x,y)) \Leftrightarrow (\exists y\, \exists x\, Q(x,y))$.

4.3.11 Následující sentence jsou splnitelné formule:

1. $\forall x\, \exists y\, Q(x,y)$,
2. $\forall x\, \forall y\, (x + y = y + x)$,

kde Q a $=$ jsou binární predikátové symboly, $+$ je binární funkční symbol. (Opět upozorňujeme, že místo zápisu $=(t_1, t_2)$ a $+(x,y)$ používáme čitelnější zápis $t_1 = t_2$ a $x + y$.)

4.3.12 Zvláštní příklady kontradikcí neuvádíme. Kontradikce jsou přesně ty formule, jejichž negace je tautologie. Tak např. formule $(\forall x\, P(x) \wedge \neg(\forall x\, P(x))$ je kontradikce. Je dobré si uvědomit, že jde o „dosazení“ formule $\forall x\, P(x)$ do výrokové kontradikce $p \wedge \neg p$.

4.3.13 Splnitelné množiny sentencí. Množina sentencí M je *splnitelná* právě tehdy, když existuje interpretace $\langle U, [\![-]\!]\rangle$, v níž jsou všechny sentence z M pravdivé. Takové interpretaci pak říkáme *model* množiny sentencí M.

Množina sentencí M je *nesplnitelná*, jestliže ke každé interpretaci $\langle U, [\![-]\!]\rangle$ existuje formule z M, která je v $\langle U, [\![-]\!]\rangle$ nepravdivá.

Z poslední definice vyplývá, že prázdná množina sentencí je splnitelná. (Porovnejte s výrokovou logikou.)

4.3.14 Příklad. Rozhodněme, zda následující množiny jsou splnitelné nebo nesplnitelné.

1. $M = \{\forall x\, (P(x) \Rightarrow Q(x)), P(a), \exists x\, (\neg Q(x))\}$,
2. $N = \{\forall x\, (P(x) \Rightarrow Q(x)), P(a), \neg(\exists x\, Q(x))\}$,

kde P a Q jsou unární predikátové symboly, a je konstantní symbol.

4.3.15 Řešení. Má-li množina model, je splnitelná; nemá-li model, je nesplnitelná. Proto se buď budeme snažit najít model dané množiny, nebo se budeme snažit ukázat, že model neexistuje. Uvědomte si, že se jedná o dvě rozdílné strategie.

Ad. 1. Zkusme nejprve formule dané množiny „přečíst". Dostáváme
„Každý prvek, který má vlastnost P, má i vlastnost Q.",
„Prvek a má vlastnost P.",
„Existuje prvek, který nemá vlastnost Q.".
Mohou být všechna tato tvrzení pravdivá současně? Zdá se, že ano. Z prvních dvou tvrzení vyplývá, že prvek a musí mít vlastnost Q, ale můžeme mít ještě nějaký jiný prvek, který vlastnost Q nemá (pak ale nesmí mít ani vlastnost P). To nás vede k této interpretaci:

$$U = \{c, d\},\ [\![a]\!] = c,\ [\![P]\!] = \{c\},\ [\![Q]\!] = \{c\}.$$

(tj. náš „svět" U má dva prvky c, d, z nichž prvek c má obě vlastnosti $[\![P]\!]$ a $[\![Q]\!]$, prvek d nemá žádnou z těchto vlastností; konstanta $[\![a]\!] = c$).

Jiný model je např. $U = \mathbb{N}$, $[\![a]\!] = 0$, vlastnost P interpretujeme jako „být dělitelný 6" a vlastnost Q jako být „sudý". Pak opravdu každé číslo dělitelné 6 je sudé; číslo 0 je dělitelné 6; a existuje přirozené číslo, které není sudé (např. číslo 3).

Ještě jiný model by byl např. tento: U je množina všech studentů elektrofakulty, a je Petr Vopršálek, vlastnost P interpretujeme jako student „má zkoušku z fyziky", vlastnost Q interpretujeme jako student „má zápočet z fyziky". Pak všechny sentence jsou v této interpretaci pravdivé za předpokladu, že na fakultě je aspoň jeden student, který nemá zápočet z fyziky, a že Petr Vopršálek je studentem elektrofakulty a zkoušku z fyziky má.

Tedy množina M je splnitelná.

Ad 2. Množina N má první dvě sentence stejné jako množina M, pouze třetí sentence je

„Není pravda, že existuje prvek, který má vlastnost Q."

Z minulého rozboru víme, že první dvě sentence zajišťují, že prvek a má vlastnost Q. Tedy v žádné interpretaci, v níž jsou pravdivé první dvě sentence, nemůže být $[\![Q]\!] = \emptyset$. Tudíž třetí sentence nemůže být pravdivá. Ukázali jsme, že množina N je nesplnitelná.

4.4 Tautologická ekvivalence

4.4.1 Tautologická ekvivalence sentencí. Řekneme, že dvě sentence φ a ψ jsou *tautologicky ekvivalentní* právě tehdy, když mají stejné modely, tj. jsou pravdivé ve stejných interpretacích. Jinými slovy, mají stejnou pravdivostní hodnovu ve všech interpretacích.

Někdy se říká, že sentence jsou *sémanticky* ekvivalentní místo, že jsou tautologicky ekvivalentní.

4.4.2 Poznámka. Dá se jednoduše dokázat, že tautologická ekvivalence je relace ekvivalence na množině všech sentencí daného jazyka $\mathscr{L}$ a že má podobné vlastnosti jako tautologická ekvivalence formulí výrokové logiky.

4.4.3 Tvrzení. Nechť φ a ψ jsou sentence. Pak platí:

$$\varphi \models\!\models \psi \quad \text{právě tehdy, když} \quad \varphi \Leftrightarrow \psi \text{ je tautologie.}$$

4.4.4 Příklad.

1. $\forall x\, \forall y\, Q(x,y) \models\!\models \forall y\, \forall x\, Q(x,y)$,
2. $\exists x\, \exists y\, Q(x,y) \models\!\models \exists y\, \exists x\, Q(x,y)$.

4.4.5 Ukážeme ještě několik tautologických ekvivalencí typických pro predikátovou logiku.

Platí následující tautologické ekvivalence (P a Q jsou unární predikátové symboly).

1. $(\forall x\, P(x)) \wedge (\forall x\, Q(x)) \models\!\models \forall x\, (P(x) \wedge Q(x))$;
2. $(\exists x\, P(x)) \vee (\exists x\, Q(x)) \models\!\models \exists x\, (P(x) \vee Q(x))$;
3. $(\forall x\, P(x)) \vee (\forall y\, Q(y)) \models\!\models \forall x\, \forall y\, (P(x) \vee Q(y))$;
4. $(\exists x\, P(x)) \wedge (\exists y\, Q(y)) \models\!\models \exists x\, \exists y\, (P(x) \wedge Q(y))$.

4.5 Sémantický důsledek

Obdobně jako ve výrokové logice definujeme i v predikátové logice pojem *sémantický důsledek* (též *konsekvent*, *tautologický důsledek*); tentokrát však jen pro množiny sentencí.

4.5.1 Sémantický důsledek. Řekneme, že sentence φ je *sémantickým důsledkem*, též *konsekventem* množiny sentencí S právě tehdy, když každý model množiny S je také modelem sentence φ. Tento fakt značíme

$$S \models \varphi.$$

Můžeme též říci, že sentence φ *není* konsekventem množiny sentencí S, jestliže existuje model množiny S, který není modelem sentence φ. To znamená, že existuje interpretace $\langle U, [\![-]\!]\rangle$, v níž je pravdivá každá sentence z množiny S a není pravdivá formule φ. Jedná se tedy o obdobný pojem jako ve výrokové logice, pouze místo o pravdivostním ohodnocení mluvíme o interpretaci.

4.5.2 Příklad. Zjistěme, zda platí $N \models \varphi$, kde $N = \{\forall x\, (P(x) \Rightarrow Q(x)), P(a)\}$ a $\varphi = \exists x\, Q(x)$.

Vezměme libovolnou interpretaci $\langle U, [\![-]\!]\rangle$, v níž jsou pravdivé obě sentence $\forall x\, (P(x) \Rightarrow Q(x))$ a $P(a)$. Protože formule $P(a)$ je pravdivá, prvek $c = [\![a]\!]$ má vlastnost $[\![P]\!]$. Protože i formule $\forall x\, (P(x) \Rightarrow Q(x))$ je pravdivá a prvek c má vlastnost $[\![P]\!]$, musí prvek c mít taky vlastnost $[\![Q]\!]$, tj. $c \in [\![Q]\!]$. Proto sentence $\exists x\, Q(x)$ je pravdivá.

4.5.3 Konvence. Jestliže množina sentencí S je jednoprvková, tj. $S = \{\psi\}$, pak píšeme $\psi \models \varphi$ místo $\{\psi\} \models \varphi$. Je-li množina S prázdná, píšeme $\models \varphi$ místo $\emptyset \models \varphi$.

4.5.4 Obdobně jako pro výrokovou logiku, dostáváme řadu jednoduchých prozorování. Pro množiny sentencí M, N a sentenci φ platí:

1. Je-li $\varphi \in M$, je $M \models \varphi$.

   Protože sentence φ leží v M, musí každý model množiny M být i modelem φ.

2. Je-li $N \subseteq M$ a $N \models \varphi$, je i $M \models \varphi$.

   Předpokládejme, že $N \models \varphi$. Každý model množiny M je také modelem množiny N a proto je modelem φ. Ukázali jsme $M \models \varphi$.

3. Je-li φ tautologie, pak $M \models \varphi$ pro každou množinu sentencí M.

   Je-li φ tautologie, je pravdivá v každé interpretaci, tedy i v těch interpretacích, které jsou modelem množiny M.

4. Je-li $\models \varphi$, pak φ je tautologie.

   Každá interpretace je modelem prázdné množiny sentencí. Proto $\models \varphi$ znamená, že každá interpretace musí být modelem φ. Tedy φ je tautologie.

5. Je-li M nesplnitelná množina, pak $M \models \varphi$ pro každou sentenci φ.

   Kdyby předchozí tvrzení nebylo pravdivé, pak by existovala interpretace, která by byla modelem množiny M a sentence φ by v ní byla nepravdivá. Ale množina M model nemá; naše tvrzení tedy platí.

4.5.5 Tvrzení. Nechť φ a ψ jsou sentence. Pak platí:

$$\varphi \models\!\models \psi \quad \text{právě tehdy, když} \quad \varphi \models \psi \text{ a } \psi \models \varphi.$$

4.5.6 Tvrzení. Nechť φ a ψ jsou sentence. Pak platí:

$$\varphi \models \psi \quad \text{právě tehdy, když} \quad \varphi \Rightarrow \psi \text{ je tautologie.}$$

4.5.7 Věta. Pro každou množinu sentencí S a každou sentenci φ platí:

$$S \models \varphi \quad \text{právě tehdy, když} \quad S \cup \{\neg\varphi\} \quad \text{je nesplnitelná množina.}$$

Tato věta i obě předcházející tvrzení se dají zdůvodnit stejným způsobem jako obdobná tvrzení ve výrokové logice. Pouze místo pojmu pravdivostní ohodnocení používáme pojem interpretace.

Kapitola 5

Rezoluční metoda v predikátové logice

Rezoluční metoda v predikátové logice je, jak napovídá její název, obdobná stejnojmenné metodě ve výrokové logice. Ovšem vzhledem k bohatší vnitřní struktuře formulí predikátové logiky je složitější. Používá se v logickém programování a je základem programovacího jazyka Prolog.

Nejprve zavedeme literály a klausule v predikátové logice.

5.0.8 Literál. *Literál* je atomická formule (tzv. *pozitivní literál*), nebo negace atomické formule (tzv. *negativní literál*). *Komlementární literály* jsou dva literály, z nichž jeden je negací druhého.

5.0.9 Klausule. *Klausule* je sentence taková, že všechny kvantifikátory jsou obecné a stojí na začátku sentence (na jejich pořadí nezáleží) a za nimi následují literál nebo disjunkce literálů.

Tedy např. $R(a,b)$, $\forall x\,(\neg P(x) \vee Q(a,b))$, $\forall x\,\forall y\,(R(x,y) \Rightarrow P(x))$ jsou klausule; $\exists x\,P(x)$, $\forall x\,\neg(P(x) \Rightarrow Q(x))$ *nejsou* klausule.

5.1 Převedení sentence na klausální tvar

Nejprve vyřešíme, jak pro danou množinu formulí M najít množinu klausulí S takovou, že množina M je splnitelná právě tehdy, když je splnitelná množina S. (Všimněte si, že tentokrát nepožadujeme, aby množiny S a M byly tautologicky ekvivalentní — to totiž obecně není možné.)

Jedná se o obdobu konjunktivní normální formy pro danou formuli ve výrokové logice. Toto nám následně pomůže využít rezoluční metodu pro zjištění, zda daná množina sentencí je splnitelná i zda daná sentence sémanticky vyplývá z dané množiny sentencí.

5.1.1 Tvrzení. Pro každou sentenci φ existuje množina klausulí S_φ taková, že sentence φ je splnitelná právě tehdy, když S_φ je splnitelná.

5.1.2 Uvedeme postup, kterým pro danou sentenci φ zkonstruujeme množinu klausulí S_φ.

1. Přejmenujeme proměnné formule φ tak, aby každý vstup kvantifikátoru vázal jinou proměnnou. (Tj. např. formuli $\forall x\, P(x) \vee \forall x\, Q(x,a)$ nahradíme formulí $\forall x\, P(x) \vee \forall y\, Q(y,a)$.)

2. Spojky $\Rightarrow$, $\Leftrightarrow$ nahradíme spojkami $\neg$, $\vee$ a $\wedge$ na základě známých tautologických rovností

$$\begin{aligned} \alpha \Rightarrow \beta \quad &\models\!\dashv \quad \neg\alpha \vee \beta \\ \alpha \Leftrightarrow \beta \quad &\models\!\dashv \quad (\neg\alpha \vee \beta) \wedge (\alpha \vee \neg\beta) \end{aligned}$$

3. Přesuneme spojku $\neg$ „co nejníže“ v derivačním stromu formule, tj. až před atomické formule. Použijeme k tomu vztahy

$$\begin{aligned} \neg\exists x\, \alpha \quad &\models\!\dashv \quad \forall x\, \neg\alpha \\ \neg\forall x\, \alpha \quad &\models\!\dashv \quad \exists x\, \neg\alpha \\ \neg(\alpha \vee \beta) \quad &\models\!\dashv \quad \neg\alpha \wedge \neg\beta \\ \neg(\alpha \wedge \beta) \quad &\models\!\dashv \quad \neg\alpha \vee \neg\beta \\ \neg\neg\alpha \quad &\models\!\dashv \quad \alpha \end{aligned}$$

4. Přesuneme spojku $\vee$ „co nejníže“ v derivačním stromu formule pomocí vztahů

$$\begin{aligned} \alpha \vee (\beta \wedge \gamma) \quad &\models\!\dashv \quad (\alpha \vee \beta) \wedge (\alpha \vee \gamma) \\ \alpha \vee (\forall x\, \beta) \quad &\models\!\dashv \quad \forall x\, (\alpha \vee \beta) \\ \alpha \vee (\exists x\, \beta) \quad &\models\!\dashv \quad \exists x\, (\alpha \vee \beta) \end{aligned}$$

   Přitom dáváme přednost první rovnosti. Teprve v případě, že první rovnost nelze aplikovat, používáme další dvě rovnosti. Uvědomte si, že druhá rovnost je opravdu tautologická ekvivalence pouze proto, že formule α neobsahuje proměnnou x (viz krok 1).

5. Použijeme tautologickou ekvivalenci

$$\forall x\, (\alpha \wedge \beta) \quad \models\!\dashv \quad (\forall x\, \alpha) \wedge (\forall x\, \beta)$$

   k distribuci obecného kvantifikátoru. Jestliže nyní formule neobsahuje existenční kvantifikátor, máme formuli ψ, která je konjunkcí klausulí. Tuto formuli tedy nahradíme množinou S_φ jejích klausulí.
   V případě, že formule ψ obsahuje existenční kvantifikátor, provedeme skolemisaci, která je vysvětlena v další části.

5.2 Skolemizace

Poznamenejme, že termíny „skolemizace“, „skolemizační konstanta“ a „skolemizační funkční symbol“ jsou odvozeny od jména norského matematika — logika Thoralfa Skolema.

Ve všech předchozích krocích uvedeného postupu jsme vždy nahrazovali formuli formulí s ní tautologicky ekvivalentní; v posledním kroku pak formuli ψ množinou klausulí opět s ψ tautologicky ekvivalentní. To už není pravda pro skolemizaci. Zde nahradíme formuli ψ formulí ψ' takovou, že formule ψ je splnitelná právě tehdy, když je splnitelná formule ψ' (obecně ale ne pro stejnou interpretaci). Dříve než ukážeme obecný postup, uvedeme dva příklady.

5.2.1 Příklad 1. Najděme klausuli ψ', která je splnitelná právě tehdy, když je splnitelná sentence $\psi = \exists x\, P(x)$.

5.2.2 Řešení příkladu 1. Vezměme sentenci $\psi' = P(a)$, kde a je nějaký nový konstantní symbol. Není obtížné nahlédnout, že formule ψ je tautologickým důsledkem formule ψ'. Kdykoli v nějaké interpretaci $\langle U, [\![-]\!]\rangle$ je pravdivá formule ψ', pak je v této interpretaci pravdivá i formule ψ. Naopak to však neplatí, může existovat interpretace, která konstantní symbol a interpretuje tak, že $[\![a]\!]$ je zrovna některý prvek z U, který neleží v $[\![P]\!]$ (tj. nemá vlastnost P). Na druhé straně, kdykoli formule ψ je pravdivá v interpretaci $\langle U, [\![-]\!]\rangle$, pak existuje prvek $d_0 \in U$ takový, že $d_0 \in [\![P]\!]$. To znamená, že změníme-li přiřazení $[\![-]\!]$ tak, že konstantní symbol a interpretujeme jako d_0, pak v této nové interpretaci je formule ψ' pravdivá.

Proto formuli $\exists x\, P(x)$ nahradíme formulí $P(a)$.

Konstantní symbol a použitý v minulém příkladě, se nazývá *skolemizační konstanta*.

5.2.3 Příklad 2. Najděme klausuli ψ', která je splnitelná právě tehdy, když je splnitelná sentence $\psi = \forall x\, \exists y\, Q(x, y)$.

5.2.4 Řešení příkladu 2. Kdybychom položili $\psi' = \forall x Q(x, a)$, dostaneme klausuli, která je podstatně „silnější“ než původní formule ψ [uvědomte si, že formule ψ' by podle předchozího příkladu odpovídala formuli $\exists y\, \forall x\, Q(x, y)$]. Doporučujeme čtenáři, aby si zjistil význam obou formulí při interpretaci, kdy U je množina přirozených čísel a $Q(x, y)$ znamená, že přirozené čislo x je menší nebo rovno přirozenému číslu y.

Formuli ψ nahradíme formulí $\psi' = \forall x Q(x, f(x))$, kde f je nový unární funkční symbol. Nyní již opět platí, že formule ψ je splnitelná právě tehdy, když je splnitelná formule ψ'. Podrobné zdůvodnění je podobné jako v předchozím příkladě.

Funkčnímu symbolu f z minulého příkladu se říká *skolemizační funkce*.

5.2.5 Obecný postup (pokračování bodů 1 – 5 z postupu 5.1.2

6. Obsahuje-li formule existenční kvantifikátor, nahradíme každou uzavřenou podformuli tvaru $\forall x_1 \ldots \forall x_n\, \exists y\, \alpha(y)$ formulí $\forall x_1 \ldots \forall x_n\, \alpha(f(x_1, \ldots, x_n))$, kde f je libovolný *nový* funkční symbol arity n. Je-li $n = 0$, použijeme *nový* konstantní symbol a. Tomuto procesu se říká *skolemizace*, funkčnímu symbolu f *skolemizační funkce*, konstantě a *skolemizační konstanta*. Pokračujeme podle kroku 5 z 5.1.2.

Uvědomte si, že proměnné $x_1, \ldots, x_n$ jsou právě všechny proměnné vázané obecným kvantifikátorem, na které narazíme při postupu derivačním stromem od $\exists y$ směrem ke kořeni.

5.2.6 Příklad 3. Převeďme na klausální tvar následující formuli

$$\varphi = \neg\forall x\, [\exists y\, Q(x, y) \Rightarrow \forall y\, \exists z\, \neg P(x, y, z)].$$

Jinými slovy, najděte množinu klausulí S_φ, která je splnitelná právě tehdy, když je splnitelná sentence φ.

5.2.7 Řešení příkladu 3. Krok 1. Přejmenujeme proměnnou y na t v podformuli $\forall y\,\exists z\,\neg P(x,y,z)$, dostaneme

$$\varphi = \neg\forall x\,[\exists y\,Q(x,y) \Rightarrow \forall t\,\exists z\,\neg P(x,t,z)].$$

Krok 2. Nahradíme spojku $\Rightarrow$:

$$\neg\forall x\,[\neg(\exists y\,Q(x,y)) \vee \forall t\,\exists z\,\neg P(x,t,z)].$$

Krok 3. Přesuneme spojku $\neg$ až před atomické formule:

$$\begin{aligned}\neg\forall x\,[\neg(\exists y\,Q(x,y)) \vee \forall t\,\exists z\,\neg P(x,t,z)] \quad &\models\!\dashv \\ \models\!\dashv \quad \exists x\,\neg[\neg(\exists y\,Q(x,y)) \vee \forall t\,\exists z\,\neg P(x,t,z)] \quad &\models\!\dashv \\ \models\!\dashv \quad \exists x\,[\exists y\,Q(x,y) \wedge \neg(\forall t\,\exists z\,\neg P(x,t,z))] \quad &\models\!\dashv \\ \models\!\dashv \quad \exists x\,[\exists y\,Q(x,y) \wedge \exists t\,\forall z\,P(x,t,z)].\end{aligned}$$

Krok 4 a 5 není potřeba. Protože sentence obsahuje existenční kvantifikátory, provedeme skolemizaci.

Postupně odstraňujeme existenční kvantifikátory a to v pořadí, v jakém se objevují v derivačním stromu.

$$\exists x\,[\exists y\,Q(x,y) \wedge \exists t\,\forall z\,P(x,t,z)] \rightsquigarrow \exists y\,Q(a,y) \wedge \exists t\,\forall z\,P(a,t,z),$$

zavedli jsme nový konstantní symbol a místo proměnné x a dosadili ho za *všechny* výskyty proměnné x.

$$\exists y\,Q(a,y) \wedge \exists t\,\forall z\,P(a,t,z) \rightsquigarrow Q(a,b) \wedge \exists t\,\forall z\,P(a,t,z) \rightsquigarrow Q(a,b) \wedge \forall z\,P(a,c,z).$$

Zavedli jsme nové konstantní symboly b, c; b místo proměnné y, c místo proměnné t. Dostali jsme formuli, která je konjunkcí dvou klausulí, proto

$$S_\varphi = \{Q(a,b), \forall z\,P(a,c,z)\}.$$

5.3 Rezolventy klausulí

5.3.1 Připomneňme, že:

- Literál je atomická formule nebo negace atomické formule.
- Klasule je sentence taková, že všechny kvantifikátory jsou obecné a stojí na začátku formule a za nimi pak následuje literál nebo disjunkce literálů.
- Prázdná klausule je klausule, která neobsahuje žádný literál a je tedy kontradikce.
- Dva literály jsou komplementární, jestliže jeden je negací druhého.

Při zápisu klausulí budeme vynechávat kvantifikátory; to můžeme proto, že víme že všechny kvantifikátory jsou obecné a na jejich pořadí nezáleží.

Ve výrokové logice jsme rezolventy vytvářeli tak, že jsme si vždy vzali dvě klausule, které obsahovaly dvojici komplementárních literálů a výsledná rezolventa byla disjunkcí všech ostatních literálů z obou klausulí. Situace v predikátové logice je složitější. Postup, jak vytváříme rezolventy v predikátové logice, si ukážeme na příkladech.

5.3.2 Příklad. Pokusme se najít rezolventu klausulí $K_1 = P(x) \vee \neg Q(x,y)$ a $K_2 = Q(x,y) \vee R(y)$, kde P a R jsou unární predikátové symboly a Q je binární predikátový symbol, x, y jsou proměnné.

5.3.3 Řešení. Rezolventu najdeme snadno: Klausule K_1 a K_2 obsahují dvojici komplementárních literálů, totiž $\neg Q(x,y)$ je literál K_1 a $Q(x,y)$ je literál K_2. Rezolventou klausulí K_1 a K_2 je tedy klausule $K = P(x) \vee R(y)$, přesněji $K = \forall x\, \forall y\, (P(x) \vee R(y))$.

5.3.4 Poznámka. Ukážeme, že klausule K je sémantickým důsledkem klausulí K_1 a K_2. Tím současně ukážeme, že množina $S = \{K_1, K_2\}$ je splnitelná právě tehdy, když je splnitelná množina $M = \{K_1, K_2, K\}$. (V tomto případě platí dokonce, že množiny S a M mají stejné modely.)

Ukažme, že každý model klausulí K_1 a K_2 je také modelem klausule K. Všechny tři klausule přepišme do tvaru sentencí: $K_1 = \forall x\, \forall y\, (P(x) \vee \neg Q(x,y))$, $K_2 = \forall x\, \forall y\, (Q(x,y) \vee R(y))$ a $K = \forall x\, \forall y (P(x) \vee R(y))$.

Uvažujme libovolnou interpretaci $\langle U, [\![-]\!] \rangle$, v níž jsou obě klausule K_1 a K_2 pravdivé. Vybereme libovolné prvky $d, d' \in U$. Pak formule $Q(x,y)$ po dosazení prvku d za proměnnou x a prvku d' za proměnnou y je buď 1) pravdivá nebo 2) nepravdivá v interpretaci $\langle U, [\![-]\!] \rangle$.

Ad 1) Protože formule $\neg Q(x,y)$ je nepravdivá pro $x := d$ a $y := d'$, musí být (z důvodů pravdivosti klausule K_1) pro $x := d$ pravdivá formule $P(x)$. Tedy formule $P(d)$ je pravdivá.

Ad 2) Protože formule $Q(x,y)$ je nepravdivá pro $x := d$ a $y := d'$, musí být (z důvodů pravdivosti klausule K_2) pro $y := d'$ pravdivá formule $R(y)$. Tedy formule $R(d')$ je pravdivá.

Ukázali jsme, že pro každé $d, d' \in U$ platí: Dosadíme-li prvek d za proměnnou x a prvek d' za proměnnou y, dostáváme pravdivou formuli $P(d) \vee R(d')$. To znamená, že klausule K je pravdivá v $\langle U, [\![-]\!] \rangle$.

Ukázali jsme, že $\{K_1, K_2\} \models K$.

V dalších příkladech již nebudeme podrobně ukazovat, že rezolventa dvou klausulí je konsekventem klausulí, z nichž vznikla. Úvahy jsou vždy obdobné.

5.3.5 Příklad. Hledejme rezolventu klausulí $K_1 = P(x) \vee \neg Q(x)$ a $K_2 = Q(a) \vee R(b)$, kde P, Q jsou unární predikátové symboly a a, b jsou konstanty.

5.3.6 Řešení. Klausule K_1 a K_2 neobsahují dvojici komplementárních literálů, neboť negace literálu $Q(a)$ není literál $\neg Q(x)$ a naopak. Přitom však literál $\neg Q(x)$ odpovídá formuli $\forall x\, \neg Q(x)$, a tedy zahrnuje i $\neg Q(a)$. Proto při substituci konstanty a za proměnnou x dostáváme klausule

$$K_1' = P(a) \vee \neg Q(a) \quad \text{a} \quad K_2' = Q(a) \vee R(b)$$

a jejich rezolventa je klausule $P(a) \vee R(b)$.

5.3.7 Příklad. Hledejme rezolventu klausulí

$$K_1 = \neg P(x, y) \vee Q(x, y, a) \quad \text{a} \quad K_2 = \neg Q(g(v), z, z) \vee R(v, z),$$

kde P, R jsou binární predikátové symboly, Q je predikátový symbol arity 3 a a je konstanta.

5.3.8 Řešení. Nejprve se pokusíme vhodnou substitucí vytvořit z $Q(x, y, a)$ a $\neg Q(g(v), z, z)$ dvojici komplementárních literálů. Je zřejmé, že toho můžeme dosáhnout jedině tehdy, když za proměnnou z dosadíme konstantu a. Tím dostaneme literály

$$Q(x, y, a) \qquad \neg Q(g(v), a, a).$$

Volba další substituce je opět zřejmá. Nyní musíme konstantu a dosadit i za proměnnou y. Dostaneme

$$Q(x, a, a) \qquad \neg Q(g(v), a, a).$$

Zbývá za proměnnou x dosadit term $g(v)$ a dostaneme komplementární literály

$$Q(g(v), a, a) \qquad \neg Q(g(v), a, a).$$

Uvědomte si, že k tomu, abychom dostali rezolventu, je nutné provést zvolenou substituci na celé klausule K_1 a K_2. Tím dostaneme klausule K_1' a K_2', jejichž rezolventa bude i rezolventou klausulí K_1, K_2. Tedy $K_1' = \neg P(g(v), a) \vee \vee Q(g(v), a, a)$, $K_2' = \neg Q(g(v), a, a) \vee R(v, a)$ a hledaná rezolventa je $K = \neg P(g(v), a) \vee R(v, a)$.

5.3.9 Poznámka. Poznamenejme, že hledáme „nejobecnější“ substituci, po jejíž aplikaci dostaneme dvojici komplementárních literálů. V předchozím příkladě můžeme volit také tuto substituci: Za proměnné y, z, v dosadíme konstantu a a za proměnnou x term $g(a)$. Provedením této substituce na klausule K_1 a K_2 dostaneme klausule

$$K_1'' = \neg P(g(a), a) \vee Q(g(a), a, a) \quad \text{a} \quad K_2'' = \neg Q(g(a), a, a) \vee R(a, a).$$

Tyto klasule mají rezolventu $K' = \neg P(g(a), a) \vee R(a, a)$. Je dobré si uvědomit, že rezolventu K' dostaneme z rezolventy K dosazením konstanty a za proměnnou v. Aby platil rezoluční princip analogický rezolučnímu principu ve výrokové logice, musíme však k množině klausulí vždy přidávat ty nejobecnější rezolventy.

5.3.10 Příklad. Hledejme rezolventu klausulí

$$K_1 = \neg P(x) \vee Q(f(x), a) \quad \text{a} \quad K_2 = \neg Q(y, y) \vee R(f(y), z),$$

kde P je unární predikátový symbol, Q a R jsou binární predikátové symboly, f je unární funkční symbol a a je konstanta.

5.3.11 Řešení. Postupujeme obdobně jako v minulém příkladě: Snažíme se najít takovou substituci, abychom z literálů $Q(f(x), a)$ a $\neg Q(y, y)$ dostali dvojici komplementárních literálů. Je zřejmé, že za proměnnou y musíme dosadit konstantu a. Dostaneme

$$Q(f(x), a) \qquad \neg Q(a, a).$$

Nyní jsme v situaci, kdy nemůžeme další substituci provést. Existují zřejmě interpretace funkčního symbolu f a konstanty a tak, že pro žádné dosazení za proměnnou x nedostaneme hodnotu konstanty a. Tudíž hledaná substituce neexistuje a klausule K_1 a K_2 nemají rezolventu.

5.4 Rezoluční princip

Nejprve uvedeme příklad a teprve potom zformulujeme vlastní princip.

5.4.1 Příklad. Rozhodněme, zda je splnitelná množina klausulí

$$S = \{\neg P(x, y) \vee Q(x, y, a), \neg Q(g(v), z, z) \vee R(v, z), \neg R(b, a), P(x, a)\},$$

kde P, Q, R jsou predikátové symboly odpovídajících arit, g je unární funkční symbol a a je konstanta.

5.4.2 Řešení. Z příkladu 5.3.7 víme, že rezolventou klausulí

$$\neg P(x, y) \vee Q(x, y, a) \quad \text{a} \quad \neg Q(g(v), z, z) \vee R(v, z)$$

je klausule $K = \neg P(g(v), a) \vee R(v, a)$. Klausule K a klausule $P(x, a)$ mají rezolventu $K' = R(v, a)$ (při substituci termu $g(v)$ za proměnnou x). Provedeme-li v klausuli $K' = R(v, a)$ substituci konstanty b za proměnnou v, dostaneme dvojici klausulí $R(b, a)$ a $\neg R(b, a)$. Jejich rezolventa je prázdná klausule F. Proto je množina S nesplnitelná.

5.4.3 Poznámka. Je dobré si uvědomit, že kdybychom jako rezolventu klausulí

$$\neg P(x, y) \vee Q(x, y, a) \quad \text{a} \quad \neg Q(g(v), z, z) \vee R(v, z)$$

vzali klausuli $\neg P(g(a), a) \vee R(a, a)$, prázdnou klausuli F bychom již neodvodili.

5.4.4 Rezoluční princip. Je dána množina klausulí S. Označme

$$\begin{aligned} R(S) &= S \cup \{K \mid K \text{ je nejobecnější rezolventa některých klausulí z } S\} \\ R^0(S) &= S \\ R^{i+1}(S) &= R(R^i(S)) \quad \text{pro} \quad i \in \mathbb{N} \\ R^\star(S) &= \bigcup\{R^i(S) \mid i \geq 0\}. \end{aligned}$$

Množina klausulí S je splnitelná právě tehdy, kdyz $R^\star(S)$ neobsahuje prázdnou klausuli F.

5.4.5 Zatím jsme mluvili o nalezení nejobecnější substituce pro dva literály tak, aby z nich vznikly komplementární literály. Neukázali jsme ale, jak ji obecně najít. Tento problém řeší následující unifikační algoritmus. Nechť $\neg L_1$ a L_2 jsou dva literály, o nichž chceme rozhodnout, zda se mohou stát dvojicí komplementárních literálů. Algoritmus pracuje s řetězci L_1 a L_2.

5.4.6 Unifikační algoritmus.

Vstup: Dva positivní literály L_1, L_2, které nemají společné proměnné.
Výstup: Hlášení `neexistuje` v případě, že hledaná substituce neexistuje,
v opačném případě substituce ve tvaru množiny prvků tvaru x/t,
kde x je proměnná, za kterou se dosazuje,
a t je term, který se za proměnnou x dosazuje.

1. Položme $E_1 := L_1$, $E_2 := L_2$, $\theta := \emptyset$.
2. Jsou-li E_1, E_2 prázdné řetězce, stop. Množina θ určuje hledanou substituci. V opačném případě položíme $E_1 := E_1\theta$, $E_2 := E_2\theta$ (tj. na E_1, E_2 provedeme substituci θ).
3. Označíme X první symbol řetězce E_1, Y první symbol řetězce E_2.
4. Je-li $X = Y$, odstraníme X a Y z počátku E_1 a E_2. Jsou-li X a Y predikátové nebo funkční symboly, odstraníme i jim příslušné závorky a jdeme na krok 2.
5. Je-li X proměnná, neděláme nic.
Je-li Y proměnná (a X nikoli), přehodíme E_1, E_2 a X, Y.
Není-li ani X ani Y proměnná, stop. Výstup `neexistuje`.
6. Je-li Y proměnná nebo konstanta, položíme $\theta := \theta \cup \{X/Y\}$. Odstraníme X a Y ze začátků řetězců E_1 a E_2 (spolu s čárkami, je-li třeba) a jdeme na krok 2.
7. Je-li Y funkční symbol, označíme Z výraz skládající se z Y a všech jeho argumentů (včetně závorek a čárek). Jestliže Z obsahuje X, stop, výstup `neexistuje`.
V opačném případě položíme $\theta := \theta \cup \{X/Z\}$, odstraníme X a Z ze začátků E_1 a E_2 (odstraníme čárky, je-li třeba) a jdeme na krok 2.

Práci unifikačního algoritmu si ukážeme na příkladě.

5.4.7 Příklad. Pomocí unifikačního algoritmu hledejme nejobecnější substituci pro literály $L_1 = Q(x, y, a)$ a $L_2 = Q(g(v), z, z)$.

5.4.8 Řešení. Algoritmus pracuje v těchto krocích:

1. $E_1 := Q(x, y, a)$, $E_2 := Q(g(v), z, z)$, $\theta := \emptyset$.
2. $X := Q$, $Y := Q$, $X = Y$. Odstraníme Q spolu s oběma závorkami, tj. máme $E_1 := x, y, a$, $E_2 := g(v), z, z$.
3. $X := x$, $Y := g$. Protože $X \neq Y$ a Y je funkční symbol, položíme $Z := g(v)$, $\theta := \{x/g(v)\}$. Odstraníme x z E_1, $g(v)$ z E_2 a dostaneme $E_1 := y, a$, $E_2 := z, z$.

4. $X := y$, $Y := z$. Protože $X \neq Y$ a Y je proměnná, položíme $\theta := \{x/g(v), y/z\}$. Odstraníme y z E_1, z z E_2 a dostaneme $E_1 := a$, $E_2 := z$.

5. $X := a$, $Y := z$. Protože $X \neq Y$, X není proměnná, ale Y je proměnná, přehodíme X a Y a současně také E_1 a E_2. Tím dostaneme $X := z$, $Y := a$, $E_1 := z$, $E_2 := a$. Nyní je Y konstantní symbol. Položíme proto $\theta := \{x/g(v), y/z, z/a\}$, odstraníme z z E_1, a z E_2 a dostaneme $E_1 = E_2 = \emptyset$. Stop, θ je hledaná substituce.

5.5 Využití rezoluční metody v predikátové logice

5.5.1 Příklad. Pomocí rezoluční metody rozhodněme, zda platí

$$\{\forall x\,\forall y\,((P(x) \wedge Q(x,y)) \Rightarrow R(y)), \forall x\,Q(f(x), g(x)), P(f(a))\} \models R(g(a)),$$

kde P, Q a R jsou predikátové symboly (odpovídajících arit) a a je konstantní symbol.

5.5.2 Řešení. Označme

$$S = \{\forall x\,\forall y\,((P(x) \wedge Q(x,y)) \Rightarrow R(y)), \forall x\,Q(f(x), g(x)), P(f(a))\}$$

a $\varphi = R(g(a))$. Formule φ je sémantickým důsledkem množiny S právě tehdy, když množina $S \cup \{\neg\varphi\}$ je nesplnitelná. Zjišťujeme tedy splnitelnost nebo nesplnitelnost množiny sentencí

$$T = S \cup \{\neg\varphi\} =$$

$$= \{\forall x\,\forall y\,((P(x) \wedge Q(x,y)) \Rightarrow R(y)), \forall x\,Q(f(x), g(x)), P(f(a)), \neg R(g(a))\}.$$

Nejprve přepíšeme formule množiny T tak, aby každý kvantifikátor vázal jinou proměnnou. Dostaneme např. množinu

$$T = \{\forall x\,\forall y\,((P(x) \wedge Q(x,y)) \Rightarrow R(y)), \forall z\,Q(f(z), g(z)), P(f(a)), \neg R(g(a))\}.$$

Nyní převedeme každou formuli z množiny T na klausální tvar: Kromě první formule již všechny formule v klausálním tvaru jsou. Upravme první formuli:

$$\forall x\,\forall y\,((P(x) \wedge Q(x,y)) \Rightarrow R(y)) \models\!\!\!\models \forall x\,\forall y\,(\neg P(x) \vee \neg Q(x,y) \vee R(y))$$

Rezoluční metodou zjistíme, zda množina klausulí

$$M = \{\neg P(x) \vee \neg Q(x,y) \vee R(y), Q(f(z), g(z)), P(f(a)), \neg R(g(a))\}$$

je splnitelná.

Nejprve vybereme predikátový symbol P a utvoříme všechny rezolventy klausulí podle literálu obsahujícího tento predikátový symbol: Literál P obsahují klausule $\neg P(x) \vee \neg Q(x,y) \vee R(y)$ a klausule $C_1 = P(f(a))$. Abychom dostali dvojici komplementárních literálů, provedeme na první klausuli substituci $x/f(a)$. Dostaneme $C_2 = \neg P(f(a)) \vee \neg Q(f(a), y) \vee R(y)$. Rezolventa klausulí C_1 a C_2 je klausule $C_3 = \neg Q(f(a), y) \vee R(y)$. Vybereme predikátový symbol

R. Rezolventu podle literálu, který obsahuje R můžeme dostat z klausulí C_3 a $C_4 = \neg R(g(a))$. K tomu ale potřebujeme v klausuli C_3 provést substituci $y/g(a)$. Dostáváme $C_3' = \neg Q(f(a), g(a)) \vee R(g(a))$; rezolventa klausulí C_3' a C_4 je klausule $C_5 = \neg Q(f(a), g(a))$. Vybereme predikátový symbol Q, ten obsahují dvě klausule: C_5 a klausule $Q(f(z), g(z))$. Substitucí z/a dostaneme dvojici klausulí $C_5 = \neg Q(f(a), g(a))$ a $Q(f(a), g(a))$, jejichž rezolventou je prázdná klausule F. Ukázali jsme, že $F \in R^3(M)$ a množina M je nesplnitelná. Proto platí $S \models \varphi$.

5.5.3 Poznámka. Všimněte si, že jsme **nejprve** vytvořili množinu $S \cup \{\neg\varphi\}$ a pak teprve upravovali sentence na klausální tvar. To je nutné proto, že používáme-li skolemizaci, nenahrazujeme sentenci sentencí tautologicky ekvivalentní. Je-li sentence $\varphi = \exists x\, P(x)$, pak $\neg\varphi = \forall x\, P(x)$; kdežto pro sentenci $P(a)$ je její negace rovna $\neg P(a)$.