

# 10 Syntaxe a sémantika výrokové a predikátové logiky. Sémantický důsledek a tautologická ekvivalence. Booleovský kalkul. Rezoluční metoda (A0B01LGR)

## 10.1 Výroková logika

### 10.1.1 Výroky

Máme danou neprázdnou množinu  $A$  tzv. *atomických výroků* (též jim říkáme *logické 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 (atomický výrok)  $a \in A$  je výroková formule.
2. Jsou-li  $\alpha, \beta$  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 výrokové formule.

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

**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 logické proměnné označujeme malými písmeny např.  $a, b, c, \dots$  nebo  $x, y, z, \dots$ , výrokové formule označujeme malými řeckými písmeny např.  $\alpha, \beta, \gamma, \dots$  nebo  $\varphi, \psi, \dots$ . Také většinou nebudeme ve formulích psát ty nejvíc vnější závorky - tj. píšeme  $a \vee (b \Rightarrow c)$  místo  $(a \vee (b \Rightarrow c))$ .

### 10.1.2 Syntaktický strom formule

To jak formule vznikla podle bodů 1 a 2, si můžeme znázornit na *syntaktickém stromu*, též *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  $(\alpha \wedge \beta)$ ,  $(\alpha \vee \beta)$ ,  $(\alpha \Rightarrow \beta)$  odpovídá levý následník formuli  $\alpha$ , pravý následník formuli  $\beta$ . Listy stromu jsou ohodnoceny logickými proměnnými.

### 10.1.3 Podformule

Ze syntaktického stromu formule  $\alpha$  jednoduše poznáme všechny její podformule: *Podformule* formule  $\alpha$  jsou všechny formule odpovídající podstromům syntaktického stromu formule  $\alpha$ .

### 10.1.4 Pravdivostní ohodnocení

*Pravdivostní ohodnocení*, též pouze *ohodnocení formulí*, je zobrazení  $u : P(A) \rightarrow \{0, 1\}$ , které splňuje pravidla

1.  $\neg\alpha$  je pravdivá právě tehdy, když  $\alpha$  je nepravdivá, tj.  $u(\neg\alpha) = 1$  právě tehdy, když  $u(\alpha) = 0$ ;
2.  $\alpha \wedge \beta$  je pravdivá právě tehdy, když  $\alpha$  a  $\beta$  jsou obě pravdivá, tj.  $u(\alpha \wedge \beta) = 1$  právě tehdy, když  $u(\alpha) = u(\beta) = 1$ ;
3.  $\alpha \vee \beta$  je nepravdivá právě tehdy, když  $\alpha$  a  $\beta$  jsou obě nepravdivé, tj.  $u(\alpha \vee \beta) = 0$  právě tehdy, když  $u(\alpha) = u(\beta) = 0$ ;
4.  $\alpha \Rightarrow \beta$  je nepravdivá právě tehdy, když  $\alpha$  je pravdivá a  $\beta$  nepravdivá, tj.  $u(\alpha \Rightarrow \beta) = 1$  právě tehdy, když  $u(\alpha) = 1$  a  $u(\beta) = 0$ ;
5.  $\alpha \Leftrightarrow \beta$  je pravdivá právě tehdy, když buď obě formule  $\alpha$  a  $\beta$  jsou pravdivé nebo obě jsou nepravdivé, tj.  $u(\alpha \Leftrightarrow \beta) = 1$  právě tehdy, když  $u(\alpha) = u(\beta)$ .

### 10.1.5 Pravdivostní tabulky

Vlastnosti, které ohodnocení formulí musí mít, znázorňujeme též pomocí tzv. pravdivostních tabulek logických spojek. Jsou to:

$\alpha$	$\neg\alpha$
0	1
1	0

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

### 10.1.6 Věta

Každé zobrazení  $u_0 : A \rightarrow \{0, 1\}$  jednoznačně určuje ohodnocení  $u : P(A) \rightarrow \{0, 1\}$  takové, že  $u_0(a) = u(a)$  pro všechna  $a \in A$ .

### 10.1.7 Důsledek

Dvě ohodnocení  $u, v : P(A) \rightarrow \{0, 1\}$  jsou shodná právě tehdy, když pro všechny logické proměnné  $x \in A$  platí  $u(x) = v(x)$ .

### 10.1.8 Tautologie, kontradikce, splnitelné formule

Formule se nazývá *tautologie*, jestliže je pravdivá ve všech ohodnoceních formulí; nazývá se *kontradikce*, jestliže je nepravdivá ve všech ohodnoceních formulí. Formule je *splnitelná*, jestliže existuje aspoň jedno ohodnocení formulí, ve kterém je pravdivá.

**Příklady:**

1. Formule  $\alpha \vee \neg\alpha$ ,  $\alpha \Rightarrow \alpha$ ,  $\alpha \Rightarrow (\beta \Rightarrow \alpha)$  jsou tautologie.
2. Formule  $a \vee b$ ,  $(a \Rightarrow b) \Rightarrow a$  jsou splnitelné, ale ne tautologie.
3. Formule  $\alpha \vee \neg\alpha$  je kontradikce. Kontradikce je také každá negace tautologie.

### 10.1.9 Tautologická ekvivalence formulí

Řekneme, že formule  $\varphi$  a  $\psi$  jsou *tautologicky ekvivalentní* (také *sémanticky ekvivalentní*), jestliže pro každé ohodnocení  $u$  platí  $u(\varphi) = u(\psi)$ .

**Tvrzení:** Pro každé formule  $\alpha, \beta, \gamma$  platí:

- $\alpha \models \alpha$ ,
- je-li  $\alpha \models \beta$ , pak i  $\beta \models \alpha$ ,
- je-li  $\alpha \models \beta$  a  $\beta \models \gamma$ , pak i  $\alpha \models \gamma$ .

Jsou-li  $\alpha, \beta, \gamma, \delta$  formule splňující  $\alpha \models \beta$  a  $\gamma \models \delta$ , pak platí

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

**Příklad:** Pro každou formuli  $\alpha$  je formule  $\alpha \Rightarrow (\beta \Rightarrow \alpha)$  tautologie.

Ano, máme

$$\alpha \Rightarrow (\beta \Rightarrow \alpha) \models \neg\alpha \vee (\neg\beta \vee \alpha) \models (\neg\alpha \vee \alpha) \vee \neg\beta,$$

kde poslední formule je tautologie.

### 10.1.10 Tvrzení

Pro každé formule  $\alpha, \beta, \gamma$  platí

- $\alpha \wedge \alpha \models \alpha$ ,  $\alpha \vee \alpha \models \alpha$  (idempotence  $\wedge$  a  $\vee$ );
- $\alpha \wedge \beta \models \beta \wedge \alpha$ ,  $\alpha \vee \beta \models \beta \vee \alpha$  (komutativita  $\wedge$  a  $\vee$ );

- $\alpha \wedge (\beta \wedge \gamma) \models (\alpha \wedge \beta) \wedge \gamma, \alpha \vee (\beta \vee \gamma) \models (\alpha \vee \beta) \vee \gamma$  (asociativita  $\wedge$  a  $\vee$ );
- $\alpha \wedge (\beta \vee \alpha) \models \alpha, \alpha \vee (\beta \wedge \alpha) \models \alpha$  (absorpce  $\wedge$  a  $\vee$ );
- $\neg\neg\alpha \models \alpha$ ;
- $(\alpha \Rightarrow \beta) \models (\neg\alpha \vee \beta)$ .

**Poznámka:** Platí  $\alpha \models \beta$  právě tehdy, když  $\alpha \Leftrightarrow \beta$  je tautologie.

### 10.1.11 Další spojky

Každá formule s jednou (nebo žádnou) logickou proměnnou představuje zobrazení z množiny  $\{0, 1\}$  do množiny  $\{0, 1\}$ . Existují čtyři taková zobrazení:

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

Funkce  $f_1$  je konstantní 0, jedná se o kontradikci a budeme ji značit **F**. Podobně funkce  $f_4$  je tautologie (konstantní 1), značíme je **T**. Funkce  $f_2$  je vlastně logická proměnná  $x$  a funkce  $f_3$  je  $\neg x$ . Tedy nemáme další unární spojky.

### 10.1.12 NAND

Logická spojka  $|$ , nazývaná NAND (také Sheffův operátor), je definována

$$x | y \models (x \wedge y).$$

### 10.1.13 NOR

Logická spojka  $\downarrow$ , nazývaná NOR (také Peiceova šipka), je definována

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

### 10.1.14 XOR

Logická spojka  $\oplus$ , nazývaná XOR (také vylučovací nebo), je definována

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

### 10.1.15 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  $n$ -tici 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 *Booleova funkce*. Naopak platí, že pro každou Booleovu 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*.

### 10.1.16 Booleova funkce

*Booleovou funkcí  $n$  proměnných*, kde  $n$  je přirozené číslo, rozumíme každé zobrazení  $f : \{0,1\}^n \rightarrow \{0,1\}$ , tj. zobrazení, které každé  $n$ -tici  $(x_1, x_2, \dots, x_n)$  nul a jedniček přiřazuje nulu nebo jedničku (označenou  $f(x_1, x_2, \dots, x_n)$ ).

### 10.1.17 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*.

**Věta:** Ke každé Booleově funkci  $f$  existuje formule v *DNF* odpovídající  $f$ .

**Důsledek:** Ke každé formuli  $\alpha$  existuje formule  $\beta$ , která je v *DNF* a navíc  $\alpha \models \beta$ .

### 10.1.18 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*.

**Věta:** Ke každé Booleově funkci  $f$  existuje formule v *CNF* odpovídající  $f$ .

**Důsledek:** Ke každé formuli  $\alpha$  existuje formule  $\beta$ , která je v *CNF* a navíc  $\alpha \models \beta$ .

### 10.1.19 Booleovský kalkul

Víme, že pro pravdivostní ohodnocení formulí platí:

$$u(a \vee b) = \max \{u(a), u(b)\} = \max \{x, y\},$$

$$u(a \wedge b) = \min \{u(a), u(b)\} = \min \{x, y\},$$

$$u(\neg a) = 1 - u(a) = 1 - x.$$

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

### 10.1.20 Booleovské operace

To motivuje zavedení booleovských operací (pro hodnoty 0,1):

$$\text{součin} \quad x \cdot y = \min \{x, y\},$$

$$\text{logický součet} \quad x + y = \max \{x, y\},$$

$$\text{doplňek} \quad \bar{x} = 1 - x.$$

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

**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.  $\bar{\bar{x}} = x$ ;
7.  $\overline{x + y} = \bar{x} \cdot \bar{y}$ ,  $\overline{x \cdot y} = \bar{x} + \bar{y}$ ;
8.  $x + \bar{x} = 1$ ,  $x \cdot \bar{x} = 0$ ;
9.  $x \cdot 0 = 0$ ,  $x \cdot 1 = x$ ;
10.  $x + 1 = 1$ ,  $x + 0 = x$ .

### 10.1.21 Booleovy funkce v DNF a CNF

Nyní můžeme pro Booleovu funkci psát pomocí výše uvedených operací, např.

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

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 Booleovy funkce  $f$ . Obdobně jako jsme Booleovu funkci  $f$  napsali v disjunktivní normální formě, můžeme ji také napsat v *konjunktivní normální formě* a to takto:

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

**Věta:** Každou Booleovu funkci lze napsat v disjunktivní normální formě i v konjunktivní normální formě.

### 10.1.22 Sémantický důsledek

#### 10.1.22.1 Množina formulí pravdivá v ohodnocení

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  $\varphi \in S$ . Jinými slovy, množina formulí  $S$  je nepravdivá v ohodnocení  $u$ , jestliže existuje formule  $\varphi \in S$ , která je nepravdivá v ohodnocení  $u$ .

Fakt, že množina formulí  $S$  je pravdivá v ohodnocení  $u$  zapisujeme též  $u(S) = 1$ , fakt, že  $S$  je nepravdivá v  $u$ , zapisujeme také  $u(S) = 0$ .

**Poznámka:** Prázdná množina formulí je pravdivá v každém ohodnocení.

#### 10.1.22.2 Splnitelná množina formulí

Řekneme, že množina formulí  $S$  je *splnitelná*, jestliže existuje pravdivostní ohodnocení  $u$ , v němž je  $S$  pravdivá. V opačném případě se množina  $S$  nazývá *nesplnitelná*.

Poznamenejme, že prázdná množina formulí je splnitelná.

#### 10.1.22.3 Sémantický důsledek

Řekneme, že formule  $\varphi$  je *konsekventem*, též *sémantickým* nebo *tautologickým důsledkem* množiny formulí  $S$ , jestliže  $\varphi$  je pravdivá v každém ohodnocení  $u$ , v němž je pravdivá  $S$ .

Fakt, že formule  $\varphi$  je konsekventem množiny  $S$ , označujeme  $S \models \varphi$ . Je-li množina  $S$  prázdná, píšeme  $\models \varphi$  místo  $\emptyset \models \varphi$ . Je-li množina  $S$  jednoprvková, tj.  $S = \{\alpha\}$ , píšeme  $\alpha \models \varphi$  místo  $\{\alpha\} \models \varphi$ .

#### 10.1.22.4 Příklady

Pro každé tři formule  $\alpha, \beta, \gamma$  platí:

1.  $\{\alpha, \alpha \Rightarrow \beta\} \models \beta$ .
2.  $\{\alpha \Rightarrow \beta, \beta \Rightarrow \gamma\} \models (\alpha \Rightarrow \gamma)$ .
3.  $\{\alpha \Rightarrow \beta, \neg\beta\} \models \neg\alpha$ .
4.  $\{\alpha \vee \beta, \alpha \Rightarrow \gamma, \beta \Rightarrow \gamma\} \models \gamma$ .

#### 10.1.22.5 Tvzení

1. Je-li  $S$  množina formulí a  $\varphi \in S$ , pak  $\varphi$  je konsekventem  $S$ , tj.  $S \models \varphi$  pro každou  $\varphi \in S$ .
2. Tautologie je konsekventem každé množiny formulí  $S$ .
3. Formule  $\varphi$  je tautologie právě tehdy, když  $\models \varphi$ .
4. Každá formule je konsekventem nesplnitelné množiny formulí.

### 10.1.22.6 Poznámka

Uvědomme si, že  $\alpha = \beta$  právě tehdy, když platí současně  $\alpha \models \beta$  a také  $\beta \models \alpha$ .

### 10.1.22.7 Tvrzení

Pro každé dvě formule  $\alpha$  a  $\beta$  platí:

$\alpha \models \beta$  právě tehdy, když  $\alpha \Rightarrow \beta$  je tautologie.

### 10.1.22.8 Věta

Pro množinu formulí  $S$  a formuli  $\varphi$  platí:

$S \models \varphi$  právě tehdy, když  $S \cup \{\neg\varphi\}$  je nespíitelná.

### 10.1.22.9 Věta o dedukci

Pro množinu formulí  $S$  a formule  $\varphi$  a  $\psi$  platí

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

## 10.1.23 Rezoluční metoda ve výrokové logice

Rezoluční metoda rozhoduje, zda daná množina klausulí je splnitelná nebo je nespíitelná. Tím je také "universální metodou" pro řešení základních problémů ve výrokové logice, neboť:

1. Daná formule  $\varphi$  je sémantickým důsledkem množiny formulí  $S$  právě tehdy, když množina  $S \cup \{\neg\varphi\}$  je nespíitelná.
2. Ke každé formuli  $\alpha$  existuje množina klausulí  $S_\alpha$  taková, že  $\alpha$  je pravdivá v ohodnocení  $u$  právě tehdy, když v tomto ohodnocení je pravdivá množina  $S_\alpha$ .

### 10.1.23.1 Klausule

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

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



### 10.1.23.2 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  $D = \text{res}_p(C_1, C_2)$ .

### 10.1.23.3 Tvrzení

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

### 10.1.23.4 Tvrzení

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

### 10.1.23.5 Rezoluční princip

Označme

$$R(S) = S \cup \{D \mid D \text{ je rezolventa některých klausulí z } S\}$$

$$R^0(S) = S$$

$$R^{i+1}(S) = R(R^i(S)) \text{ pro } i \in \mathbb{N}$$

$$R^*(S) = \bigcup \{R^i(S) \mid i \geq 0\}.$$

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

### 10.1.23.6 Věta (Rezoluční princip)

Množina klausulí  $S$  je splnitelná právě tehdy, když  $R^*(S)$  neobsahuje prázdnou klausuli  $F$ .

### 10.1.23.7 Základní postup

Předchozí věta dává návod, jak zjistit, zda daná množina klausulí je splnitelná nebo je nesplnitelná:

1. Formule množiny  $M$  převedeme do CNF a  $M$  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  $R^*(S)$ .

3. Obsahuje-li  $R^*(S)$  prázdnou klausuli, je množina  $S$  (a tedy i množina  $M$ ) nesplnitelná, v opačném případě je  $M$  splnitelná.

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

### 10.1.23.8 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 sestavit 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 pozitivní 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  $S_1 = M_0 \cup N$ .

**Tvrzení:** Množina klausulí  $S_1$  zkonstruovaná výše je splnitelná právě tehdy, když je splnitelná množina  $S$ .

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 klausuli  $F$ . Tedy  $S$  je nesplnitelná.
2. Dostaneme prázdnou množinu klausulí. V tomto případě je množina  $S$  splnitelná.

## 10.2 Predikátová logika

### 10.2.1 Syntaxe predikátové logiky

Nejprve 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 řetězce (posloupnosti) symbolů tzv. *jazyka predikátové logiky*.

### 10.2.1.1 Jazyk predikátové logiky $\mathcal{L}$

*Jazyk predikátové logiky* se skládá z

1. *logických symbolů*, tj.:

- a) spočetné množiny individuálních proměnných:  $Var = \{x, y, \dots, x_1, x_2, \dots\}$
- 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  $n$  kolika objektů se daný predikát týká, nebo kolika proměnných je daný funkční symbol. Tomuto číslu říkáme *arita* nebo též *četnost* predikátového symbolu nebo funkčního symbolu. Funkční symboly mají aritu větší nebo rovnu 1, predikátové symboly připouštíme i arity 0.

### 10.2.1.2 Poznámka

Predikátové symboly budeme většinou značit velkými písmeny, tj. např.  $P, Q, R, \dots, P_1, P_2, \dots$ ; konstantní symboly malými písmeny ze začátku abecedy, tj.  $a, b, c, \dots, a_1, a_2, \dots$ , a funkční symboly většinou  $f, g, h, \dots, f_1, f_2, \dots$ . 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*.

Predikátové symboly arity 0 představují nestrukturované výroky (netýkají se žádného objektu). Tímto způsobem se v predikátové logice dá popsat i výrok: „Prší”.

### 10.2.1.3 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  $t_1, t_2, \dots, t_n$  jsou termy, pak  $f(t_1, t_2, \dots, t_n)$  je také term.
3. Nic, co nevzniklo konečným použitím pravidel 1 a 2, není term.

**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“.

#### 10.2.1.4 Atomické formule

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

#### 10.2.1.5 Formule

Množina *formulí* je definována těmito pravidly:

1. Každá atomická formule je formule.
2. Jsou-li  $\varphi$  a  $\psi$  dvě formule, pak  $(\neg\varphi)$ ,  $(\varphi \wedge \psi)$ ,  $(\varphi \vee \psi)$ ,  $(\varphi \Rightarrow \psi)$ ,  $(\varphi \Leftrightarrow \psi)$  jsou opět formule.
3. Je-li  $\varphi$  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.

#### 10.2.1.6 Konvence

1. Úplně vnější závorky nepíšeme. Píšeme tedy např.  $(\exists xP(x)) \vee R(a, b)$  místo  $((\exists xP(x)) \vee R(a, b))$ .
2. Spojka „negace“ má vždy přednost před výrokovými logickými spojkami a proto píšeme např.  $\forall x(\neg P(x) \Rightarrow Q(x))$  místo  $\forall x((\neg P(x)) \Rightarrow Q(x))$ .

#### 10.2.1.7 Syntaktický strom formule

Ke každé formuli predikátové logiky můžeme přiřadit její *syntaktický strom* (též *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 syntaktický strom. Listy syntaktického stromu jsou vždy ohodnoceny buď proměnnou nebo konstantou.

### 10.2.1.8 Podformule

Podformule formule  $\varphi$  je libovolný podřetězec  $\varphi$ , který je sám formulí. Jinými slovy: Podformule formule  $\varphi$  je každý řetězec odpovídající podstromu syntaktického stromu formule  $\varphi$ , určenému vrcholem ohodnoceným predikátovým symbolem, logickou spojkou nebo kvantifikátorem.

### 10.2.1.9 Volný a vázaný výskyt proměnné

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

### 10.2.1.10 Sentence

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

### 10.2.1.11 Legální přejmenování proměnné

Přejmenování výskytů proměnné  $x$  ve formuli  $\varphi$  je legálním přejmenováním proměnné, jestliže

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

### 10.2.1.12 Rovnost formulí

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

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

## 10.2.2 Sémantika predikátové logiky

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

### 10.2.2.1 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*;

- $\langle [-] \rangle$  je přiřazení, které
  1. každému predikátovému symbolu  $P \in Pred$  arity  $n$  přiřazuje podmnožinu  $[P]$  množiny  $U^n$ , tj.  $n$ -ární relaci na množině  $U$ .
  2. každému konstantnímu symbolu  $a \in Kons$  přiřazuje prvek z  $U$ , značíme jej  $[a]$ ,
  3. každému funkčnímu symbolu  $f \in Func$  arity  $n$  přiřazuje zobrazení množiny  $U^n$  do  $U$ , značíme je  $[f]$ ,

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

### 10.2.2.2 Kontext proměnných

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

$$p[x := d]$$

označuje kontext proměnných, který má stejné hodnoty jako  $\rho$ , a liší se pouze v proměnné  $x$ , kde má hodnotu  $d$ . Kontextu proměnných  $p[x := d]$  též říkáme *update* kontextu  $\rho$  o hodnotu  $d$  v  $x$ .

### 10.2.2.3 Interpretace termů při daném kontextu proměnných

Je dána interpretace  $\langle U, [-] \rangle$  a kontext proměnných  $\rho$ . Pak termy interpretujeme následujícím způsobem.

1. Je-li term konstantní symbol  $a \in Kons$ , pak jeho hodnota je prvek  $[a]_\rho = [a]$ . Je-li term proměnná  $x$ , pak jeho hodnota je  $[x]_\rho = \rho(x)$ .
2. Je-li  $f(t_1, t_2, \dots, t_n)$  term, pak jeho hodnota je

$$[f(t_1, t_2, \dots, t_n)]_\rho = [f]([t_1]_\rho, \dots, [t_n]_\rho).$$

Jinými slovy, hodnota termu  $f(t_1, t_2, \dots, t_n)$  je funkční hodnota funkce  $[f]$  provedené na  $n$ -tici prvků  $[t_1]_\rho, \dots, [t_n]_\rho$  z  $U$ .

### 10.2.2.4 Pravdivostní hodnota formule v dané interpretaci a daném kontextu

Nejprve definujeme *pravdivost* formulí v dané interpretaci  $\langle U, [-] \rangle$  při daném kontextu proměnných  $\rho$ :

1. Nechť  $\varphi$  je atomická formule. Tj.  $\varphi = P(t_1, t_2, \dots, t_n)$ , kde  $P$  je predikátový symbol arity  $n$  a  $t_1, t_2, \dots, t_n$  jsou termy. Pak  $\varphi$  je pravdivá v interpretaci  $\langle U, [-] \rangle$  a kontextu  $\rho$  právě tehdy, když

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

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

2. Jsou-li  $\varphi$  a  $\psi$  formule, jejichž pravdivost v interpretaci  $\langle U, [-] \rangle$  a kontextu  $\rho$  již známe, pak

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

3. Je-li  $\varphi$  formule a  $x$  proměnná, pak

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

#### 10.2.2.5 Pravdivostní hodnota sentence

Sentence  $\varphi$  je *pravdivá v interpretaci*  $\langle U, [-] \rangle$  právě tehdy, když je pravdivá v každém kontextu proměnných  $\rho$ .

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

#### 10.2.2.6 Model sentence

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

#### 10.2.2.7 Tautologie, kontradikce, splnitelná sentence

Sentence  $\varphi$  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.

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.  $P(a) \Rightarrow (\exists x P(x))$ ;

Následující sentence jsou splnitelné formule:

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

kde  $Q$  = jsou binární predikátové symboly,  $+$  je binární funkční symbol.

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  $p \wedge \neg p$ .

### 10.2.2.8 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.

## 10.2.3 Tautologická ekvivalence

### 10.2.3.1 Tautologická ekvivalence sentencí

Řekneme, že dvě sentence  $\varphi$  a  $\psi$  jsou *tautologicky ekvivalentní* právě tehdy, když mají stejné modely, tj. jsou pravdivé ve stejných interpretacích. Jinými slovy, mají stejnou pravdivostní hodnotu ve všech interpretacích.

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

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

### 10.2.3.2 Tvzení

Nechť  $\varphi$  a  $\psi$  jsou sentence. Pak platí:

$\varphi \models \psi$  právě tehdy, když  $\varphi \Leftrightarrow \psi$  je tautologie.

Tautologické ekvivalence: ( $P$  a  $Q$  jsou unární predikátové symboly.)

1.  $\neg(\forall x P(x)) \models (\exists x \neg P(x))$ ,
2.  $\neg(\exists x P(x)) \models (\forall x \neg P(x))$ .
3.  $(\forall x P(x)) \wedge (\forall x Q(x)) \models \forall x (P(x) \wedge Q(x))$ ;



## 10.2.4 Sémantický důsledek

### 10.2.4.1 Sémantický důsledek

Řekneme, že sentence  $\varphi$  je *sémantickým důsledkem*, též *konsekventem* množiny sentencí  $S$  právě tehdy, když každý model množiny  $S$  je také modelem sentence  $\varphi$ . Tento fakt značíme

$$S \models \varphi.$$

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

### 10.2.4.2 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  $\emptyset \models \varphi$ .

Obdobně jako pro výrokovou logiku, dostáváme řadu jednoduchých pozorování. Pro množiny sentencí  $M$ ,  $N$  a sentence  $\varphi$  platí:

1. Je-li  $\varphi \in M$ , je  $M \models \varphi$ .
2. Je-li  $N \subseteq M$  a  $N \models \varphi$ , je i  $M \models \varphi$ .
3. Je-li  $\varphi$  tautologie, pak  $M \models \varphi$  pro každou množinu sentencí  $M$ .
4. Je-li  $\models \varphi$ , pak  $\varphi$  je tautologie.
5. Je-li  $M$  nespílitelná množina, pak  $M \models \varphi$  pro každou sentence  $\varphi$ .

### 10.2.4.3 Tvrzení

Nechť  $\varphi$  a  $\psi$  jsou sentence. Pak platí:

$\varphi \models \psi$  právě tehdy, když  $\varphi \models \psi$  a  $\psi \models \varphi$ .

### 10.2.4.4 Tvrzení

Nechť  $\varphi$  a  $\psi$  jsou sentence. Pak platí:

$\varphi \models \psi$  právě tehdy, když  $\varphi \Rightarrow \psi$  je tautologie.

### 10.2.4.5 Věta

Pro každou množinu sentencí  $S$  a každou sentence  $\varphi$  platí:

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

### 10.2.5 Rezoluční metoda v predikátové logice

Rezoluční metoda v predikátové logice je 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.

#### 10.2.5.1 Literál

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

#### 10.2.5.2 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.

Ve výrokové logice jsme pro každou formuli  $\alpha$  našli k ní tautologicky ekvivalentní množinu klausulí  $S_\alpha$  a to tak, že  $\alpha$  i  $S_\alpha$  byly pravdivé ve stejných pravdivostních ohodnocení. Takto jednoduchá situace v predikátové logice není. Ukážeme si, jak k dané sentenci  $\varphi$  najít množinu klausulí  $S_\varphi$  a to tak, že  $\varphi$  je splnitelná právě tehdy, když množina  $S_\varphi$  je splnitelná.

#### 10.2.5.3 Rezolventy klausulí

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.

**Poznámka:** Ne vždy rezolventa existuje.

#### 10.2.5.4 Příklad

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

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  $K = \forall x \forall y (P(x) \vee R(y))$ .

#### 10.2.5.5 Unifikační algoritmus

Vstup: Dva pozitivní 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  $\theta$  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  $\theta$ ).
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.

#### 10.2.5.6 Rezoluční princip

Je obdobný jako rezoluční princip ve výrokové logice:

Je dána množina klausulí  $S$ . Označme

$$R(S) = S \cup \{K \mid K \text{ je nejobecnější rezolventa některých klausulí z } S\}$$

$$R^0(S) = S$$

$$R^{i+1}(S) = R(R^i(S)) \text{ pro } i \in \mathbb{N}$$

$$R^*(S) = \bigcup \{R^i(S) \mid i \geq 0\}.$$

Množina klausulí  $S$  je splnitelná právě tehdy, když  $R^*(S)$  neobsahuje prázdnou klausuli  $F$ .

Jestliže je množina  $S$  konečná, existuje přirozené číslo  $n_0$  takové, že  $R^{n_0}(S) = R^{n_0+1}(S)$ . Pak  $R^*(S) = R^{n_0}(S)$ .