1 Převody do normálních forem

Příklad 1.1: Vyjádřete následující formule v DNF pomocí pravdivostní tabulky a pomocí převodu logických spojek.

a) $(A \Rightarrow B) \Rightarrow C$

b) $(A \Leftrightarrow B) \vee \neg C$

c) $(A \Leftrightarrow B) \Rightarrow (C \vee D)$

Formule je v disjunktivní normální formě (DNF), pokud má tvar $\alpha_1 \vee \ldots \vee \alpha_n$, *kde* $\alpha_i = A_{i1} \wedge \ldots \wedge A_{ij_i}$ *a každé* A_{ij} *je výroková proměnná nebo její negace.*

Řešení 1.1:

a) Pro formuli $(A \Rightarrow B) \Rightarrow C$ vytvoříme pravdivostní tabulku.

A	B	C	$A \Rightarrow B$	$(A \Rightarrow B) \Rightarrow C$
0	0	0	1	0
0	0	1	1	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

Pro každou interpretaci, ve které je formule pravdivá, přidáme do vytvořené formule novou formuli, která bude obsahovat konjunkci všech výrokových symbolů z původní formule, následovně: Pokud je výrokovému symbolu v dáné interpretaci přiřazena pravdivostní hodnota 0, přidáme do formule negaci tohoto symbolu. V opačném případě do formule přidáme symbol samotný.

Výsledkem je pak formule v tzv. *úplné disjunktivní normální formě*:
$(\neg A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge \neg C) \vee (A \wedge \neg B \wedge C) \vee (A \wedge B \wedge C)$

Při přímem převodu postupujeme následovně: Ekvivalence a implikace nahrazujeme disjunkcemi, konjunkcemi a negacemi, pak uplatníme de Morganova pravidla a associativitu a distributivitu.

$$\begin{aligned} (A \Rightarrow B) \Rightarrow C &\Leftrightarrow (\neg A \vee B) \Rightarrow C \Leftrightarrow \\ &\Leftrightarrow \neg(\neg A \vee B) \vee C \Leftrightarrow \\ &\Leftrightarrow (A \wedge \neg B) \vee C \end{aligned}$$

b) Formuli $(A \Leftrightarrow B) \vee \neg C$ převedeme do DNF následovně:

$$\begin{aligned} (A \Leftrightarrow B) \vee \neg C &\Leftrightarrow ((A \Rightarrow B) \wedge (B \Rightarrow A)) \vee \neg C \Leftrightarrow \\ &\Leftrightarrow ((\neg A \vee B) \wedge (\neg B \vee A)) \vee \neg C \Leftrightarrow \\ &\Leftrightarrow (\neg A \wedge \neg B) \vee (\neg A \wedge A) \vee (B \wedge \neg B) \vee (B \wedge A) \vee \neg C \\ &\Leftrightarrow (\neg A \wedge \neg B) \vee (B \wedge A) \vee \neg C \end{aligned}$$

c) Přímým použitím převodních vztahů dostaneme:

$$\begin{aligned}(A \Leftrightarrow B) \Rightarrow (C \vee D) &\Leftrightarrow ((A \Rightarrow B) \wedge (B \Rightarrow A)) \Rightarrow (C \vee D) \Leftrightarrow \\ &\Leftrightarrow \neg((A \Rightarrow B) \wedge (B \Rightarrow A)) \vee (C \vee D) \Leftrightarrow \\ &\Leftrightarrow \neg((\neg A \vee B) \wedge (\neg B \vee A)) \vee (C \vee D) \Leftrightarrow \\ &\Leftrightarrow (A \wedge \neg B) \vee (B \wedge \neg A) \vee C \vee D\end{aligned}$$

□

Příklad 1.2: Vyjádřete následující formule v CNF, a to pomocí pravdivostní tabulky a pomocí převodu logických spojek.

a) $(A \Leftrightarrow B) \Rightarrow (\neg A \wedge C)$

b) $(A \Rightarrow B) \Leftrightarrow (A \Rightarrow C)$

Formule je v konjunktivní normální formě (CNF), pokud má tvar $\alpha_1 \wedge \ldots \wedge \alpha_n$*, kde* $\alpha_i = A_{i1} \vee \ldots \vee A_{ij_i}$ *a* A_{ij} *je výroková proměnná nebo její negace.*

Řešení 1.2:

a) Z pravdivostní tabulky pro formuli $(A \Leftrightarrow B) \Rightarrow (\neg A \wedge C)$ vytvoříme ekvivalentní formuli v CNF takto:

A	B	C	$A \Leftrightarrow B$	$\neg A \wedge C$	$(A \Leftrightarrow B) \Rightarrow (\neg A \wedge C)$
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	1	0	1	1
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	1	0	0

Pro každou interpretaci, ve které je formule nepravdivá, přidáme do vytvářené formule novou formuli, která bude obsahovat disjunkci všech výrokových symbolů z původní formule, následovně: Pokud je výrokovému symbolu v dáné interpretaci přiřazena pravdivostní hodnota 1, přidáme do formule negaci tohoto symbolu. V opačném případě do formule přidáme symbol samotný.

Výsledkem je tentokrát formule v tzv. *úplné konjunktivní normální formě*:

$$(A \vee B \vee C) \wedge (\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee \neg C)$$

Přímým užitím převodních vztahů pak dostaneme:

$$\begin{aligned}(A \Leftrightarrow B) \Rightarrow (\neg A \wedge C) &\Leftrightarrow ((A \Rightarrow B) \wedge (B \Rightarrow A)) \Rightarrow (\neg A \wedge C) \Leftrightarrow \\ &\Leftrightarrow ((\neg A \vee B) \wedge (\neg B \vee A)) \Rightarrow (\neg A \wedge C) \Leftrightarrow \\ &\Leftrightarrow \neg((\neg A \vee B) \wedge (\neg B \vee A)) \vee (\neg A \wedge C) \Leftrightarrow \\ &\Leftrightarrow (\neg(\neg A \vee B) \vee \neg(\neg B \vee A)) \vee (\neg A \wedge C) \Leftrightarrow \\ &\Leftrightarrow ((A \wedge \neg B) \vee (B \wedge \neg A)) \vee (\neg A \wedge C) \Leftrightarrow \\ &\Leftrightarrow \ldots\end{aligned}$$

Uplatněním distributivních zákonů dostaneme požadovaný tvar.

b) Přímým převodem dostaneme:

$$\begin{aligned}
& (A \Rightarrow B) \Leftrightarrow (A \Rightarrow C) \Leftrightarrow \\
\Leftrightarrow\ & ((A \Rightarrow B) \Rightarrow (A \Rightarrow C)) \wedge ((A \Rightarrow C) \Rightarrow (A \Rightarrow B)) \Leftrightarrow \\
\Leftrightarrow\ & (\neg(A \Rightarrow B) \vee (A \Rightarrow C)) \wedge (\neg(A \Rightarrow C) \vee (A \Rightarrow B)) \Leftrightarrow \\
\Leftrightarrow\ & ((A \wedge \neg B) \vee (\neg A \vee C)) \wedge ((A \wedge \neg C) \vee (\neg A \vee B)) \Leftrightarrow \\
\Leftrightarrow\ & (((A \vee \neg A) \wedge (\neg B \vee \neg A)) \vee C) \wedge \\
& \wedge(((A \vee \neg A) \wedge (\neg C \vee \neg A)) \vee B) \Leftrightarrow \\
\Leftrightarrow\ & (\neg B \vee \neg A \vee C) \wedge (\neg C \vee \neg A \vee B)
\end{aligned}$$

Poznámka: *Některé formule jsou zároveň v disjunktivní i konjunktivní normální formě, například $A \vee B \vee \neg C$ nebo $\neg A \wedge B$ nebo třeba $\neg C$.*

Poznámka: *Každá formule má nějakou odpovídající DNF a CNF, tyto normální formy však nejsou určeny jednoznačně. Například CNF formule $(P \Rightarrow Q) \wedge (Q \Rightarrow R) \wedge (R \Rightarrow P)$ může být $(\neg P \vee Q) \wedge (\neg Q \vee R) \wedge (\neg R \vee P)$, ale stejně tak i $(\neg P \vee R) \wedge (\neg R \vee Q) \wedge (\neg Q \vee P)$.* □

Příklad 1.3: Převeďte na prenexovou normální formu formule:

a) $\forall y(\exists x P(x,y) \Rightarrow Q(y,z)) \wedge \exists y(\forall x R(x,y) \vee Q(x,y))$

b) $\exists x R(x,y) \Leftrightarrow \forall y P(x,y)$

c) $(\forall x \exists y Q(x,y) \vee \exists x \forall y P(x,y)) \wedge \neg \exists x \exists y P(x,y)$

d) $\neg(\forall x \exists y P(x,y) \Rightarrow \exists x \exists y R(x,y)) \wedge \forall x(\neg \exists y Q(x,y))$

Formule je v prenexové normání formě (PNF), pokud jsou všechny kvatifikátory na začátku formule, tj. formule má tvar $Q_1x_1Q_2x_2 \dots Q_nx_n\varphi$, kde $Q_1, \dots, Q_n \in \{\forall, \exists\}$ a φ je formule bez kvantifikátorů (tzv. otevřená formule).

Řešení 1.3: Pro převod do prenexové formy používáme tato pravidla:

$$\overrightarrow{Qx}\ \neg\forall y\varphi \quad \Leftrightarrow \quad \overrightarrow{Qx}\ \exists y\neg\varphi \tag{1}$$

$$\overrightarrow{Qx}\ \neg\exists y\varphi \quad \Leftrightarrow \quad \overrightarrow{Qx}\ \forall y\neg\varphi \tag{2}$$

$$\overrightarrow{Qx}\ (\forall y\varphi \oplus \psi) \quad \Leftrightarrow \quad \overrightarrow{Qx}\ \forall z(\varphi(y/z) \oplus \psi) \tag{3}$$

$$\overrightarrow{Qx}\ (\exists y\varphi \oplus \psi) \quad \Leftrightarrow \quad \overrightarrow{Qx}\ \exists z(\varphi(y/z) \oplus \psi) \tag{4}$$

Symbol $\overrightarrow{Qx}$ označuje vektor kvantifikátorů, které již splňují požadavky na PNF. Symbol $\oplus$ v rovnicích (3) a (4) zastupuje logickou spojku $\wedge$ nebo $\vee$. Výrazem (y/z) rozumíme substituci proměnné y za proměnnou z (proměnná y je ve formuli φ nahrazena proměnnou z), která se nevyskytuje nikde ve formuli.

a) užitím uvedených pravidel dostaneme z

$$\forall y(\exists x P(x,y) \Rightarrow Q(y,z)) \wedge \exists y(\forall x R(x,y) \vee Q(x,y))$$

formuli v PNF takto[1]:

$$
\begin{aligned}
& \forall y(\exists x P(x,y) \Rightarrow Q(y,z)) \wedge \exists y(\forall x R(x,y) \vee Q(x,y)) \Leftrightarrow \\
\Leftrightarrow\ & \forall y(\neg(\exists x P(x,y)) \vee Q(y,z)) \wedge \exists y(\forall x R(x,y) \vee Q(x,y)) \Leftrightarrow \\
\Leftrightarrow\ & \forall y(\forall x \neg P(x,y) \vee Q(y,z)) \wedge \exists y(\forall x(R(x,y) \vee Q(x,y)) \Leftrightarrow \\
\Leftrightarrow\ & \forall y_1[(\forall x \neg P(x,y_1) \vee Q(y_1,z)) \wedge \exists y(\forall x R(x,y) \vee Q(x,y))] \Leftrightarrow \\
\Leftrightarrow\ & \forall y_1 \forall x_1[(\neg P(x_1,y_1) \vee Q(y_1,z)) \wedge \exists y(\forall x R(x,y) \vee Q(x,y))] \Leftrightarrow \\
\Leftrightarrow\ & \forall y_1 \forall x_1 \exists y_2[(\neg P(x_1,y_1) \vee Q(y_1,z)) \wedge (\forall x R(x,y_2) \vee Q(x,y_2))] \Leftrightarrow \\
\Leftrightarrow\ & \forall y_1 \forall x_1 \exists y_2[(\neg P(x_1,y_1) \vee Q(y_1,z)) \wedge \forall x_2(R(x_2,y_2) \vee Q(x,y_2))] \Leftrightarrow \\
\Leftrightarrow\ & \forall y_1 \forall x_1 \exists y_2 \forall x_2[(\neg P(x_1,y_1) \vee Q(y_1,z)) \wedge (R(x_2,y_2) \vee Q(x,y_2))]
\end{aligned}
$$

b) formuli $\exists x R(x,y) \Leftrightarrow \forall y P(x,y)$ převedeme následovně:

$$
\begin{aligned}
& \exists x R(x,y) \Leftrightarrow \forall y P(x,y) \Leftrightarrow \\
\Leftrightarrow\ & (\exists x R(x,y) \Rightarrow \forall y P(x,y)) \wedge (\forall y P(x,y) \Rightarrow \exists x R(x,y)) \Leftrightarrow \\
\Leftrightarrow\ & (\neg \exists x R(x,y) \vee \forall y P(x,y)) \wedge (\neg \forall y P(x,y) \vee \exists x R(x,y)) \Leftrightarrow \\
\Leftrightarrow\ & (\forall x \neg R(x,y) \vee \forall y P(x,y)) \wedge (\exists y \neg P(x,y) \vee \exists x R(x,y)) \Leftrightarrow \\
\Leftrightarrow\ & \forall x_1[(\neg R(x_1,y) \vee \forall y P(x,y)) \wedge (\exists y \neg P(x,y) \vee \exists x R(x,y))] \Leftrightarrow \\
\Leftrightarrow\ & \forall x_1 \forall y_1[(\neg R(x_1,y) \vee P(x,y_1)) \wedge (\exists y \neg P(x,y) \vee \exists x R(x,y))] \Leftrightarrow \\
\Leftrightarrow\ & \forall x_1 \forall y_1[(\neg R(x_1,y) \vee P(x,y_1)) \wedge \exists y_2(\neg P(x,y_2) \vee \exists x R(x,y))] \Leftrightarrow \\
\Leftrightarrow\ & \forall x_1 \forall y_1 \exists y_2[(\neg R(x_1,y) \vee P(x,y_1)) \wedge (\neg P(x,y_2) \vee \exists x R(x,y))] \Leftrightarrow \\
\Leftrightarrow\ & \forall x_1 \forall y_1 \exists y_2[(\neg R(x_1,y) \vee P(x,y_1)) \wedge \exists x_2(\neg P(x,y_2) \vee R(x_2,y))] \Leftrightarrow \\
\Leftrightarrow\ & \forall x_1 \forall y_1 \exists y_2 \exists x_2[(\neg R(x_1,y) \vee P(x,y_1)) \wedge (\neg P(x,y_2) \vee R(x_2,y))]
\end{aligned}
$$

□

2 Skolemizace a unifikace

Začneme připomenutím následujících definic.

- Formule bez volných proměnných se nazývá *sentence*.
- Nechť $\varphi(x_1,\ldots,x_n)$ je formule s volnými proměnnými $x_1,\ldots,x_n$, kde $n \geq 0$. Jejím *univerzálním uzávěrem* rozumíme formuli $\forall x_1 \ldots \forall x_n \varphi(x_1,\ldots,x_n)$.
- Formule $\varphi(x_1,\ldots,x_n)$ je *pravdivá v interpretaci* I právě tehdy, když je v této interpretaci pravdivý její univerzální uzávěr, tj. pokud $\varphi(x_1,\ldots,x_n)$ je pravdivá v interpretaci I pro všechny valuace.
- Podobně, formule $\varphi(x_1,\ldots,x_n)$ je *splnitelná*, je-li splnitelný její univerzální uzávěr, tj. pokud existuje interpretace I taková, že $\varphi(x_1,\ldots,x_n)$ je pravdivá pro všechny valuace.
- Formule φ,ψ jsou *equisatisfiable* (ekvivalentní vzhledem ke splnitelnosti), pokud jsou obě splnitelné nebo obě nesplnitelné.

[1] Při převodu pro zjednodušení vynécháme krok, který vysune kvantifikátory v levé části formule mezi vnější (značíme symbolem “[”) a vnitřní závorku.

- Nechť T je množina formulí. Formule φ je *logickým důsledkem* T (nebo φ *logicky vyplývá* z T, píšeme $T \models \varphi$,), pokud je φ pravdivá v každé interpretaci I, ve které jsou pravdivé všechny formule z T.

Nyní si ukážeme, jak lze převést úlohu *Dokažte, že φ je logickým důsledkem* T do klauzulární formy (tj. konjunktivní normální formy), kde již může být dořešena rezoluční metodou.

a) Nejprve nahradíme všechny formule s volnými proměnnými v T jejich univerzálními uzávěry. Vzniklou množinu sentencí označíme T'. Má-li formule φ volné proměnné $x_1, \ldots, x_n$, nahradíme ji jejím univerzálním uzávěrem $\forall x_1 \ldots \forall x_n \varphi(x_1, \ldots, x_n)$. Z výše uvedených definic plyne, že $T \models \varphi$ právě tehdy, když $T' \models \forall x_1 \ldots \forall x_n \varphi(x_1, \ldots, x_n)$.

b) $T' \models \forall x_1 \ldots \forall x_n \varphi(x_1, \ldots, x_n)$ právě tehdy, když je množina sentencí $T' \cup \{\neg \forall x_1 \ldots \forall x_n \varphi(x_1, \ldots, x_n)\} = T' \cup \{\exists x_1 \ldots \exists x_n \neg \varphi(x_1, \ldots, x_n)\}$ nesplnitelná.

c) Všechny formule z množiny $T' \cup \{\exists x_1 \ldots \exists x_n \neg \varphi(x_1, \ldots, x_n)\}$ převedeme do prenexního normální formy (PNF) a skolemizujeme (skolemizace vytvoří formuli, která je equisatisfiable původní formuli). Získanou množinu sentencí v PNF a bez existenčních kvantifikátorů označíme T''.

d) Nyní z každé formule z T'' odstraníme část s kvantifikátory (korektnost tohoto kroku opět vyplývá z výše uvedených definic) a zbylou část formule převedeme do konjunktivní normální formy. Konjunkci všech takto získaných formulí nazveme ψ. Platí $T \models \varphi$ právě tehdy, když ψ je nesplnitelná. Formuli ψ přepíšeme na množinu klauzulí a každou klauzuli nahradíme množinou jejích literálů. K důkazu nesplnitelnosti takto vytvořené množiny klauzulí můžeme použít rezoluční metodu.

Pořadí některých částí výše uvedeného postupu lze zaměnit, např. převod do konjunktivní normální formy lze provést zaráz s převodem do PNF.

Příklad 2.1: Proveďte skolemizaci následujících formulí v PNF:

a) $\forall y_1 \forall x_1 \exists y_2 \forall x_2 [(\neg P(x_1, y_1) \vee Q(y_1, a)) \wedge (R(x_2, y_2) \vee Q(x_1, y_2))]$

b) $\forall x_1 \forall y_1 \exists y_2 \exists x_2 [(\neg R(x_1, y_2) \vee P(b, y_1)) \wedge (\neg P(x_1, y_2) \vee R(x_2, b))]$

c) $\exists x_1 \forall y_1 \exists x_2 (S(y_1) \vee R(x_1, x_2))$

Řešení 2.1: Každý výskyt proměnné x, která je ve formuli kvantifikovaná existenčně, nahradíme termem $f(y_1, \ldots, y_n)$, kde f je nový funkční symbol a $y_1, \ldots, y_n$ jsou všechny univerzálně kvantifikované proměnné, které se vyskytují v sekvenci kvantifikátorů před proměnnou x. Pokud se před x žádné takové proměnné nevyskytují, nahradíme x nulárním funkčním symbolem, tj. konstantou.

Skolemizace se aplikuje na formule bez volných proměnných. Formuli s volnými proměnnými je tedy třeba před skolemizací nahradit jejím univerzálním uzávěrem (univerzální uzávěr i skolemizace zachovávají splnitelnost formule).

a) První formuli upravíme následovně:

$$\begin{array}{rrll} & \forall y_1 \forall x_1 \exists y_2 \forall x_2 & [(\neg P(x_1, y_1) \vee Q(y_1, a)) \wedge & \\ & & \wedge(R(x_2, y_2) \vee Q(x_1, y_2))] & \rightarrow \\ \rightarrow & \forall y_1 \forall x_1 \forall x_2 & [(\neg P(x_1, y_1) \vee Q(y_1, a)) \wedge & \\ & & \wedge(R(x_2, f(y_1, x_1)) \vee Q(x_1, f(y_1, x_1)))] & \end{array}$$

b) Stejným způsobem upravíme formuli:

$$\begin{array}{rrll} & \forall x_1 \forall y_1 \exists y_2 \exists x_2 & [(\neg R(x_1, y_2) \vee P(b, y_1)) \wedge & \\ & & \wedge(\neg P(x_1, y_2) \vee R(x_2, b))] & \rightarrow \\ \rightarrow & \forall x_1 \forall y_1 \exists x_2 & [(\neg R(x_1, f(x_1, y_1)) \vee P(b, y_1)) \wedge & \\ & & \wedge(\neg P(x_1, f(x_1, y_1)) \vee R(x_2, b))] & \rightarrow \\ \rightarrow & \forall x_1 \forall y_1 & [(\neg R(x_1, f(x_1, y_1)) \vee P(b, y_1)) \wedge & \\ & & \wedge(\neg P(x_1, f(x_1, y_1)) \vee R(g(x_1, y_1), b))] & \end{array}$$

c) Skolemizace třetí formule ukazuje náhradu proměnné novou konstantou:

$$\begin{array}{rrll} & \exists x_1 \forall y_1 \exists x_2 & (S(y_1) \vee R(x_1, x_2)) & \rightarrow \\ \rightarrow & \forall y_1 \exists x_2 & (S(y_1) \vee R(a, x_2)) & \rightarrow \\ \rightarrow & \forall y_1 & (S(y_1) \vee R(a, f(y_1))) & \end{array}$$

□

Příklad 2.2: Najděte nejobecnější unifikátory (angl. *most general unifiers*, *mgu*) následujících množin literálů:

a) $S = \{P(x, f(y), z), P(g(a), f(w), u), P(v, f(b), c)\}$

b) $T = \{Q(h(x, y), w), Q(h(g(v), a), f(v)), Q(h(g(v), a), f(b))\}$

Řešení 2.2: Při hledání *mgu* hledáme rozdíly mezi výrazy a substituujeme volné proměnné tak dlouho, dokud vstupní množina neobsahuje právě jeden výraz.[2]

Rozdíl $D(S)$ mezi výrazy z množiny S definujeme jako množinu podvýrazů všech $E \in S$ začínajících na první (nejlevější) pozici, na které mají výrazy E různou hodnotu.

a) Nejobecnější unifikátor pro množinu

$$S = \{P(x, f(y), z), P(g(a), f(w), u), P(v, f(b), c)\}$$

zkonstruujeme následovně.

1. $S_0 = S$, $|S_0| \neq 1$ a je tedy co unifikovat. $D(S) = \{x, g(a), v\}$. Máme čtyři možnosti jak substituovat ($[x/g(a)]$, $[x/v]$, $[v/g(a)]$ a $[v/x]$). Zvolíme první možnost, tedy $\phi_1 = [x/g(a)]$ a aplikujeme ϕ_1 na S_0, čímž obdržíme množinu S_1:

$$S_1 = S_0\phi_1 = \{P(g(a), f(y), z), P(g(a), f(w), u), P(v, f(b), c)\}$$

[2] Viz materiály k předmětu *IB101 Úvod do logiky a logického programování* (šestá přednáška).

2. $D(S_1) = \{g(a), v\}$, $\phi_2 = [v/g(a)]$.

$$S_2 = S_1\phi_2 = \{P(g(a), f(y), z), P(g(a), f(w), u), P(g(a), f(b), c)\}$$

3. $D(S_2) = \{y, w, b\}$, $\phi_3 = [y/w]$.

$$S_3 = S_2\phi_2 = \{P(g(a), f(w), z), P(g(a), f(w), u), P(g(a), f(b), c)\}$$

4. $D(S_3) = \{w, b\}$, $\phi_3 = [w/b]$.

$$S_4 = S_3\phi_3 = \{P(g(a), f(b), z), P(g(a), f(b), u), P(g(a), f(b), c)\}$$

5. $D(S_4) = \{z, u, c\}$, $\phi_3 = [z/u]$.

$$S_5 = S_4\phi_4 = \{P(g(a), f(b), u), P(g(a), f(b), u), P(g(a), f(b), c)\}$$

6. $D(S_5) = \{u, c\}$, $\phi_3 = [u/c]$.

$$S_6 = S_5\phi_5 = \{P(g(a), f(b), c)\}$$

7. $|S_6| = 1$, takže algoritmus pro nalezení unifikátoru ukončí výpočet. Nejobecnějším unifikátorem je posloupnost substitucí:

$$\begin{aligned} mgu(S) &= \phi_1\phi_2\phi_3\phi_4\phi_5 = \\ &= [x/g(a)][v/g(a)][y/w][w/b][z/u][u/c] \end{aligned}$$

b) Aplikací popsaného postupu dostaneme pro množinu T tento nejobecnější unifikátor:

$$mgu(T) = [x/g(v)][y/a][w/f(v)][v/b]$$

□

Příklad 2.3: Najděte všechny rezolventy následujících dvojic klauzulí:

a) $C_1 = \{P(x, y), P(y, z)\}, C_2 = \{\neg P(u, f(u))\}$

b) $C_1 = \{P(x, x), \neg R(x, f(x))\}, C_2 = \{R(x, y), Q(y, z)\}$

c) $C_1 = \{P(x, y), \neg P(x, x), Q(x, f(x), z)\}, C_2 = \{\neg Q(f(x), x, z), P(x, z)\}$

Řešení 2.3: Před rezolucí je třeba přejmenovat proměnné v klauzulích tak, aby obě klauzule používaly různé proměnné. Rezoluce v predikátové logice probíhá obdobně jako v logice propoziční. Rezoluční pravidlo je definováno následovně:[3] Mějme klauzule C_1 a C_2 bez společných proměnných ve tvaru:

$$\begin{aligned} C_1 &= C_1' \cup \{P(\vec{x_1}), \ldots, P(\vec{x_n})\} \\ C_2 &= C_2' \cup \{\neg P(\vec{y_1}), \ldots, \neg P(\vec{y_m})\}. \end{aligned}$$

Je-li substituce ϕ nejobecnějším unifikátorem množiny

$$\{P(\vec{x_1}), \ldots, P(\vec{x_n}), P(\vec{y_1}), \ldots, P(\vec{y_m})\},$$

pak rezolventou C_1 a C_2 je $C_1'\phi \cup C_2'\phi$.

Poznámka:

[3] Viz materiály k předmětu *IB101 Úvod do logiky a logického programování* (sedmá přednáška).

- *Přejmenování proměnných je nutné:* $\{\{P(x)\}, \{\neg P(f(x))\}\}$ *je nesplnitelná množina klauzulí, ovšem literály* $P(x)$ *a* $P(f(x))$ *nejsou unifikovatelné.*
- *Rezolvování na více literálech je nutné:* $\{\{\neg P(x), \neg P(y)\}, \{P(x), P(y)\}\}$ *je nesplnitelná množina klauzulí, ovšem rezolvováním na jednom literálu bychom se nikdy nedostali k prázdné klauzuli.*

Na zadané klauzule aplikujeme rezoluční pravidlo a pokusíme se nalézt všechny rezolventy, které lze odvodit.

a) Přejmenování proměnných není v tomto případě nutné, protože klauzule neobsahují společné proměnné.

1. nejdříve zkusíme odvodit prázdnou klauzuli $\square$. Protože však nelze unifikovat množinu $\{P(x, y), P(y, z), P(u, f(u))\}$, není $\square$ rezolventou C_1 a C_2.
2. zkusíme tedy unifikovat množinu $\{P(x, y), P(u, f(u))\}$. Zjistíme, že *mgu* této množiny je posloupnost substitucí $\phi = [x/u][y/f(u)]$ a můžeme tedy odvodit rezolventu

$$\begin{aligned} C_1'\phi \cup C_2'\phi &= \\ &= (C_1 \setminus \{P(x, y)\})\phi \cup (C_2 \setminus \{\neg P(u, f(u))\})\phi = \\ &= \{P(y, z)\}\phi \cup \emptyset = \\ &= \{P(f(u), z)\}. \end{aligned}$$

3. nakonec můžeme unifikovat množinu $\{P(y, z), P(u, f(u))\}$ pomocí *mgu* $\phi = [y/u][z/f(u)]$. Rezolventou je poté klauzule $\{P(x, u)\}$.

$\square$

3 Rezoluce

Příklad 3.1: Dokažte, že platí následující vyplývání:

a) $\{\forall x P(x, x), \forall x \forall y \forall z((P(x, y) \wedge P(y, z)) \Rightarrow P(z, x))\} \models \forall x \forall y (P(x, y) \Rightarrow P(y, x))$

b) $\{\forall x \forall y \forall z((P(x, y) \wedge P(y, z)) \Rightarrow P(x, z)), \forall x \forall y (P(x, y) \Rightarrow P(y, x))\} \models \forall x \forall y \forall z((P(x, y) \wedge P(z, y)) \Rightarrow P(x, z))$

Řešení 3.1: Stejně jako ve výrokové logice provedeme důkaz sporem pomocí rezoluční metody. Protože jsou všechny formule v zadání tohoto příkladu uzavřené, stačí nalézt vyvrácení množiny $S \cup \{\neg\varphi\}$, kde S je množina formulí z předpokladů a φ je formule ze závěru logického vyplývání.

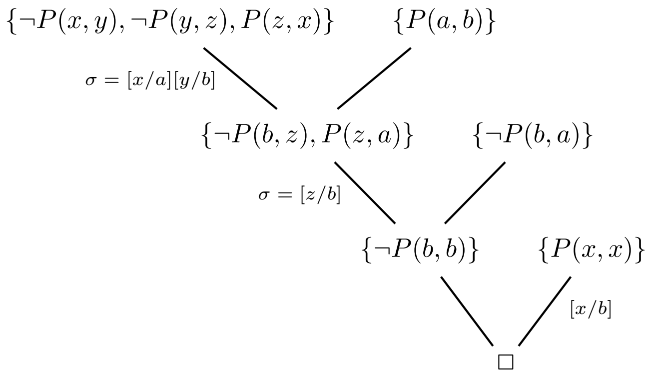
Množinu formulí $S \cup \{\neg\varphi\}$ tedy převedeme na množinu klauzulí[4] a pomocí rezolučního pravidla pro predikátovou logiku se pokusíme odvodit prázdnou klauzuli.

[4] Podrobný obecný postup transformace zadání na množinu klauzulí je popsán v materiálech ke druhému cvičení.

a) Znegováním závěru získáme formuli $\neg\varphi = \exists x \exists y \neg(P(x,y) \Rightarrow P(y,x))$. Převodem této formule a množiny předpokladů do PNF, odstraněním kvantifikací a převodem do CNF obdržíme množinu klauzulí

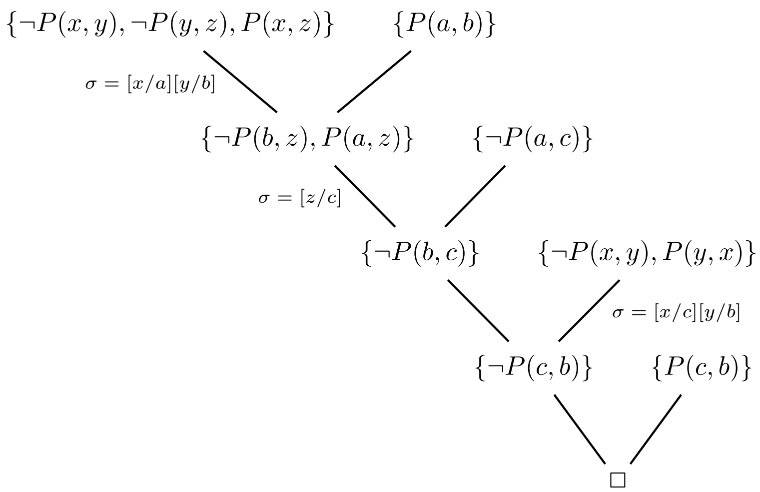
$$\{\{P(x,x)\}, \{\neg P(x,y), \neg P(y,z), P(z,x)\}, \{P(a,b)\}, \{\neg P(b,a)\}\}.$$

Z ní můžeme vytvořit např. tento strom rezolučního vyvrácení:



b) Obdobně jako v předchozím případě vyvrátíme množinu klauzulí:

$$\{\{\neg P(x,y), \neg P(y,z), P(x,z)\}, \{\neg P(x,y), P(y,x)\},$$
$$\{P(a,b)\}, \{P(c,b)\}, \{\neg P(a,c)\}\}.$$



V obou případech se nám podařilo odvodit z množiny $S \cup \{\neg\varphi\}$ prazdnou klauzuli a tím jsme dokázali platnost logického vyplývání $S \models \varphi$. $\square$

Příklad 3.2: Převeďte následující tvrzení v přirozeném jazyce na formule predikátové logiky a dokažte jejich platnost.

a) Předpokládejte, že platí následující tři tvrzení:

- Existuje drak (označme $D/1$).

- Draci spí ($S/1$) nebo loví ($L/1$).
- Když jsou draci hladoví ($H/1$), tak nespí.

Důsledek: *Když jsou draci hladoví, tak loví.*

b) Předpokládejte, že platí následující dvě tvrzení:

- Všichni holiči ($B/1$) holí ($S/2$) každého, kdo se neholí sám.
- Žádný holič neholí někoho, kdo se holí sám.

Důsledek: *Holiči neexistují.*

Řešení 3.2: Převedeme věty přirozeného jazyka na formule predikátové logiky a rezolučním principem dokážeme platnost tvrzení.

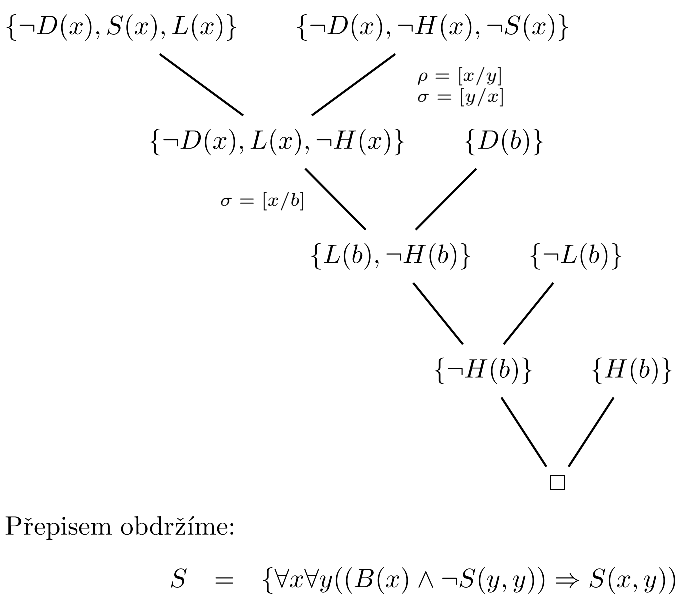
a) Přepisem obdržíme následující množinu formulí:

$$\begin{aligned} S &= \{\exists x D(x), \\ &\quad \forall x(D(x) \Rightarrow (S(x) \vee L(x))), \\ &\quad \forall x((D(x) \wedge H(x)) \Rightarrow \neg S(x))\} \\ \varphi &= \forall x((D(x) \wedge H(x)) \Rightarrow L(x)) \end{aligned}$$

Všechny předpoklady z S a negaci závěru φ převedeme na klauzulární tvar a dokazujeme nesplnitelnost takto vzniklé množiny klauzulí:

$$\begin{aligned} S' &= \{\{D(a)\}, \{\neg D(x), S(x), L(x)\}, \{\neg D(x), \neg H(x), \neg S(x)\}, \\ &\quad \{D(b)\}, \{H(b)\}, \{\neg L(b)\}\} \end{aligned}$$

V dalším textu budeme při zápisu rezoluce obvykle používat ρ pro označení přejmenování proměnných v klauzulích a σ pro označení *mgu* substituce.



b) Přepisem obdržíme:

$$\begin{aligned} S &= \{\forall x \forall y((B(x) \wedge \neg S(y,y)) \Rightarrow S(x,y)), \\ &\quad \forall y(S(y,y) \Rightarrow \neg \exists x(B(x) \wedge S(x,y)))\} \\ \varphi &= \neg \exists x B(x) \end{aligned}$$

Množinu formulí $S \cup \{\neg\varphi\}$ převedeme na množinu klauzulí S', ke které pak zkonstruujeme rezoluční vyvrácení.

$$S' = \{\{\neg B(x), S(y,y), S(x,y)\}, \{\neg S(y,y), \neg B(x), \neg S(x,y)\}, \{B(a)\}\}$$

$\{\neg B(x), S(y,y), S(x,y)\}$ $\{\neg S(y,y), \neg B(x), \neg S(x,y)\}$

$\rho = [x/v][y/w]$
$\sigma = [x/y][w/y][v/y]$

$\{\neg B(y)\}$ $\{B(a)\}$

$\sigma = [y/a]$

$\square$

$\square$