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.

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.

Příklad 1.3: Víme, že můžeme provádět následující chemické reakce:

$$\begin{aligned} \mathrm{MgO} + \mathrm{H_2} &\rightarrow \mathrm{Mg} + \mathrm{H_2O} \\ \mathrm{C} + \mathrm{O_2} &\rightarrow \mathrm{CO_2} \\ \mathrm{CO_2} + \mathrm{H_2O} &\rightarrow \mathrm{H_2CO_3} \end{aligned}$$

K dispozici máme dále sloučeniny C, H_2, O_2 a MgO. Rezolucí dokažte, že H_2CO_3 logicky vyplývá ze zadaných reakcí.

Příklad 1.4: 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 $Q_1x_1Q_2x_2\ldots Q_nx_n\varphi$, kde $Q_1,\ldots,Q_n \in \{\forall,\exists\}$ a φ je formule bez kvantifikátorů (tzv. otevřená formule).

2 Skolemizace a unifikace

Začneme připomenutím následujících definic.

- Formle 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.
- 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) $\exists x_1 \forall y_1 \exists x_2 (S(y_1) \vee R(x_1, x_2))$

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

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

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

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.*

Příklad 3.3: Vyvraťte následující množinu klauzulí

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

pomocí obecné rezoluce, lineární rezoluce, LI rezoluce, LD rezoluce a SLD rezoluce[1].

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

4 SLD-stromy a rezoluce v Prologu

Příklad 4.1: Nalezněte rezoluční vyvracení množiny klauzulí zadaných jako program a dotaz v Prologu.

```
1. r :- p, q.       5. t.
2. s :- p, q.       6. q.
3. v :- t, u.       7. u.
4. w :- v, s.       8. p.

?- w.
```

Příklad 4.2: Vytvořte SLD-strom pro následující program (`Program 1.`) a dotaz v Prologu a zjistěte, jak se projeví změna pořadí klauzulí (`Program 2.`) v definici programu na výsledné podobě SLD-stromu.

```
Program 1:                          Program 2:
 1. p :- q,r.     4. r :- q.         1. p :- r.       4. r.
 2. p :- r.       5. r.              2. p :- q,r.     5. r :- q.
 3. q :- p.                          3. q :- p.

 ?- p,q.
```

Příklad 4.3: Sestrojte SLD-stromy pro následující programy a dotazy v jazyce Prolog:

```
Program 1:                       Program 2:
 1. p :- a,r.  5. r :- t,a.       1. p :- s,t.  5.  r :- w.
 2. a :- b.    6. r :- s.         2. p :- q.    6.  r.
 3. a.         7. s.              3. q.         7.  s.
 4. b :- a.                       4. q :- r.    8.  t :- w.

 ?- p.                            ?- p.
```

Příklad 4.4: Napište SLD-strom pro následující program a dotaz.

```
1. p(f,g).      3. p(Z,X) :- p(X,Y), p(Y,Z).
2. p(X,X).

?- p(Y,f).
```

Příklad 4.5: Vytvořte SLD-strom všech možných řešení k následujícímu programu a dotazu:

```
1. p(X,Y) :- q(X,Z), r(Z,Y).        7. s(X) :- t(X,a).
2. p(X,X) :- s(X).                  8. s(X) :- t(X,b).
3. q(X,b).                          9. s(X) :- t(X,X).
4. q(b,a).                         10. t(a,b).
5. q(X,a) :- r(a,X).               11. t(b,a).
6. r(b,a).

?- p(X,X).
```

Příklad 4.6: Zkonstruujte SLD vyvrácení cíle `?- reverse([a,b,c],X).` za předpokladu, že máme nadefinován predikát `reverse/2` následovně:

```
reverse(L1,L2) :- rev(L1,[],L2).
rev([H|T],A,L) :- rev(T,[H|A],L).
rev([],L,L).
```

5 Tabla ve výrokové logice

Příklad 5.1: Sestrojte *ukončené tablo* s kořenem

$$F(((C \vee d) \wedge (D \vee \neg d)) \Leftrightarrow (C \vee D)),$$

kde C, D, d jsou výrokové symboly.

Příklad 5.2: Pomocí tabel dokažte, že následující formule jsou tautologické:

a) $\neg(p \Rightarrow q) \;\Rightarrow\; (q \Rightarrow p)$

b) $((p \vee q) \Rightarrow (p \vee r)) \;\Rightarrow\; (p \vee (q \Rightarrow r))$

Příklad 5.3: Dokažte, že platí následující logické vyplývání:

$$\{q \Rightarrow r, r \Rightarrow (p \wedge q), p \Rightarrow (q \vee r)\} \models (p \Leftrightarrow q).$$

6 Tabla v predikátové logice

Příklad 6.1: Pomocí tabel dokažte, že následující formule jsou tautologie.

a) $\Phi_1 \equiv \forall x \varphi(x) \Rightarrow \exists x \varphi(x)$

b) $\Phi_2 \equiv \forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\forall x P(x) \Rightarrow \forall x Q(x))$

c) $\Phi_3 \equiv \forall x(\varphi(x) \wedge \psi(x)) \Leftrightarrow (\forall x \varphi(x) \wedge \forall x \psi(x))$

d) $\Phi_4 \equiv \exists y \forall x(P(x,y) \Leftrightarrow P(x,x)) \Rightarrow \neg \forall x \exists y \forall z(P(z,y) \Leftrightarrow \neg P(z,x))$

Příklad 6.2: Dokažte, že formule $\forall x P(x)$ logicky vyplývá z předpokladů:

$$\forall x((Q(x) \vee R(x)) \Rightarrow \neg S(x))$$
$$\forall x((R(x) \Rightarrow \neg P(x)) \Rightarrow (Q(x) \wedge S(x)))$$

Příklad 6.3: Pomocí tabel dokažte platnost následujících tvrzení.

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.*

7 Tabla v modální logice

Při konstrukci ukončeného kontradiktorického tabla v modální logice postupujeme podobně jako v případě logiky prvního řádu. Máme-li dokázat, že formule φ je tautologií (tj. platí ve všech světech všech Kripkeho rámců nad použitým jazykem modální logiky), stačí zkonstruovat kontradiktorické tablo s kořenem $Fw \Vdash \varphi$. Každý uzel vytvářeného binárního stromu pak obsahuje výraz $Fv \Vdash \varphi$ nebo $Tv \Vdash \varphi$. Narozdíl od logiky prvního řádu je tedy nutné brát v úvahu také svět, ve kterém danou položku redukujeme.

Cesta v tablu je *sporná (kontradiktorická)*, pokud se na ní vyskytuje položka $Tv \Vdash \varphi$ a zároveň $Fv \Vdash \varphi$ pro nějaký svět v a formuli φ.

Kvůli technickému zjednodušení přijímáme konvenci, že jazyk modální logiky neobsahuje funkční symboly. Při expanzi položek $Tv \Vdash (\forall x)\varphi(x)$ a $Fv \Vdash (\exists x)\varphi(x)$ proto místo ground termů používáme pouze konstanty. Musíme však volit takové konstanty, o kterých víme, že se vyskytují ve světě v. Taková konstanta je buď součástí jazyka modální logiky, nebo se na expandované cestě již vyskytuje v nějaké položce týkající se světa v nebo nějakého jeho předchůdce (tj. světa w, pro který se na cestě vyskytuje wSv). Další možností je naopak volit konstantu, která se nevyskytuje nikde v celém tablu. Při expanzi položek $Tv \Vdash (\exists x)\varphi(x)$ a $Fv \Vdash (\forall x)\varphi(x)$ dosadíme namísto x libovolnou konstantu, která se dosud nevyskytuje v žádné položce tabla.

Položky s operátory $\Box$ a $\Diamond$ redukujeme následovně. Při redukci položky $Tv \Vdash \Diamond\varphi$ nebo $Fv \Vdash \Box\varphi$ na cestu nejdříve přidáme výraz vSw, kde w je nějaký nový svět, který ještě v tablu nebyl použit. Poté přidáme nový uzel $Tw \Vdash \varphi$ resp. $Fw \Vdash \varphi$. Položky typu $Tv \Vdash \Box\varphi$ a $Fv \Vdash \Diamond\varphi$ expandujeme na $Tw \Vdash \varphi$ a $Fw \Vdash \varphi$, kde w je libovolný svět, pro který je na expandované cestě položka vSw. Pokud žádný takový svět neexistuje, můžeme položky považovat za redukované.

Kořeny atomických tabel pro položky $Tv \Vdash (\forall x)\varphi(x)$, $Fv \Vdash (\exists x)\varphi(x)$, $Tv \Vdash \Box\varphi$ a $Fv \Vdash \Diamond\varphi$ bychom neměli při redukci vynechávat.

Příklad 7.1: Pomocí tabel dokažte, že následující formule jsou tautologie.

a) $\Phi_1 \equiv (\Box\forall x\varphi(x)) \Rightarrow (\forall x\Box\varphi(x))$

b) $\Phi_2 \equiv (\Box(\varphi \Rightarrow \psi)) \Rightarrow ((\Box\varphi \Rightarrow \Box\psi))$

c) $\Phi_3 \equiv \neg\Diamond(\neg(\varphi \wedge \exists x\psi(x)) \wedge \exists x(\varphi \wedge \psi(x)))$, x není volné ve formuli φ

d) $\Phi_4 \equiv \Diamond\exists x(\varphi(x) \Rightarrow \Box\psi) \Rightarrow \Diamond(\forall x\varphi(x) \Rightarrow \Box\psi)$, x není volné ve formuli ψ

Příklad 7.2: Mějme modální tablo s kořenem $Fw \Vdash \forall x\Box\varphi(x) \Rightarrow \Box\forall x\varphi(x)$ definované v obrázku 1. Rozhodněte o korektnosti uvedeného důkazu a své tvrzení zdůvodněte.

$$
\begin{array}{c}
Fw \Vdash \forall x \Box \varphi(x) \Rightarrow \Box \forall x \varphi(x) \\
| \\
Tw \Vdash \forall x \Box \varphi(x) \\
| \\
Fw \Vdash \Box \forall x \varphi(x) \\
| \\
wSv \\
| \\
Fv \Vdash \forall x \varphi(x) \\
| \\
Fv \Vdash \varphi(c) \\
| \\
Tw \Vdash \forall x \Box \varphi(x)^* \\
| \\
Tw \Vdash \Box \varphi(c) \\
| \\
Tw \Vdash \Box \varphi(c) \\
| \\
Tv \Vdash \varphi(c) \\
| \\
\otimes
\end{array}
$$

Obrázek 1:

Příklad 7.3: Dokažte platnost úsudků:

a) $\{\varphi\} \models \Box\varphi$

b) $\{\forall x \varphi(x)\} \models \Box \forall x \varphi(x)$

c) $\{\forall x \varphi(x)\} \models \forall x \Box \varphi(x)$

d) $\{\varphi \Rightarrow \Box\varphi\} \models \Box\varphi \Rightarrow \Box\Box\varphi$

8 Induktivní logické programování

Příklad 8.1: Jsou dány výrokové symboly a, b, c, d, e a tabulka

a	b	c	d	e	*třída*
1	1	1	0	1	+
1	1	0	1	1	+
1	1	0	1	0	−

$+$ = pozitivní příklady
$-$ = negativní příklad

Najděte všechny specializace výrokové formule b. Vyznačte, které z nich pokrývají negativní příklad.

Příklad 8.2: Napište část specializačního grafu s kořenem `neteř(X,Y)`, která bude obsahovat klauzuli:

```
neteř(X,Y) :- žena(X), sourozenec(Y,Z), rodič(Z,X).
```