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 $\Box \in \Delta$ existuje formule α obsahující pouze spojky z množiny Π a taková, že $\alpha \models\!\!\!\dashv x \Box y$,
2. pro každou unární spojku $\Diamond \in \Delta$ existuje formule β obsahující pouze spojky z množiny Π a taková, že $\beta \models\!\!\!\dashv \Diamond x$,
3. pro každou nulární spojku $K \in \Delta$ existuje formule γ obsahující pouze spojky z množiny Π a taková, že $\gamma \models\!\!\!\dashv K$,

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:

   $$(a \Rightarrow b) \models\!\!\!\dashv (\neg a \vee b) \quad \text{a} \quad (a \wedge b) \models\!\!\!\dashv \neg(\neg a \vee \neg b).$$

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

   $$(a \vee b) \models\!\!\!\dashv \neg(\neg a \wedge \neg b).$$

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

   Obdobně jako výše si stačí uvědomit, že

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

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

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

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

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

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 $\alpha \models\!\!\!\dashv \beta$.

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 $\alpha \models\!\!\!\dashv \beta$.

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):

$$\begin{array}{lrcl} \text{součin} & x \cdot y & = & \min\{x, y\}, \\ \text{logický součet} & x + y & = & \max\{x, y\}, \\ \text{doplněk} & \overline{x} & = & 1 - x. \end{array}$$

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.