**1.8.16 Příklad.** Pomocí rezoluční metody rozhodněte, zda je splnitelná množina klauzulí

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

Jestliže je splnitelná, najděte pravdivostní ohodnocení, ve kterém je $S$ pravdivá.

**Řešení:** Postupujeme obdobně jako v minulém příkladě. Dostaneme následující tabulku 1.2. (Uvědomte si, že tvar tabulky je určen pořadím výběru jednotlivých logických proměnných.)

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

Tabulka 1.2: Příklad tabulky pro splnitelnou množinu

Všimněte si, že v předposledním řádku jsme nepřidali sloupec pro jednu resolventu; je to proto, že $res_d(\neg a \vee \neg d, a \vee d)$ je tautologie $a \vee \neg a$.

Z tabulky je patrné, že množina $S$ je splnitelná, protože nakonec jsme získali prázdnou množinu klauzulí — nezbyl žádný neoznačený sloupec — a prázdná množina je pravdivá ve všech ohodnoceních, tudíž je splnitelná.

Nyní z tabulky 1.2 odvodíme pravdivostní ohodnocení, ve kterém je množina $S$ pravdivá. Postup je znázorněn v tabulce 1.3.

| | $a \vee \neg d$ | $\neg b \vee \neg c$ | $b \vee d$ | $\neg b \vee \neg e$ | $a \vee c \vee d$ | $\neg a \vee \neg d$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| $e$: | | | | 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 1.3: Konstrukce pravdivostního ohodnocení

Poslední řádek tabulky byl ohodnocen proměnnou $a$. Protože v tomto řádku máme 1, prohlásíme literál $a$ za pravdivý, tj. položíme $u(a) = 1$. Označíme si v tabulce **všechny** klauzule, které se touto volbou $u(a) = 1$ stanou pravdivé (označeny jsou šipkou s číslem 1).

Nyní přistoupíme k předposlednímu řádku tabulky. Sloupce, které nejsou označené šipkou 1, obsahují v tomto řádku už jen 0. Rozšíříme proto pravdivostní ohodnocení $u$ tak, aby literál $\neg d$ by pravdivý, tj. položíme $u(d) = 0$. Označíme šipkou s indexem 2 ty klauzule, které se touto volbou stanou pravdivé (a nestaly se pravdivé již v předchozím kroku).

Obdobným způsobem postupujeme v tabulce nahoru až zajistíme, že všechny klauzule obsažené v tabulce budou pravdivé: dostaneme $u(b) = 1$, $u(c) = 0$ a $u(e) = 0$.

Je snadné se přesvědčit, že v takto definovaném pravdivostním ohodnocení je původní množina klauzulí $S$ pravdivá.

**1.8.17 Poznámka.** V předchozím příkladě jsme definovali pravdivostní ohodnocení pro všechny logické proměnné, které se v množině $S$ nacházely. To se nemusí stát vždy. V případě, že zajistíme pravdivost všech klauzulí nezávisle na některé logické proměnné, znamená to, že tuto proměnnou můžeme definovat libovolně.

Uvědomte si, že z jedné tabulky obecně nedostaneme **všechna** pravdivostní ohodnocení, ve kterých je původní množina pravdivá. Při jiném pořadí proměnných jsme mohli dostat jiná ohodnocení.

Též se může stát, že při zpětném postupu v některém řádku, řekněme logické proměnné $t$, nezbude ani 0, ani 1. V takovém případě musíme zkusit obě možnosti — definovat $u(t) = 1$ a když tato volba nevede k cíli, tak $u(t) = 0$. Jedna z možností povede k cíli.

**1.8.18 Důkaz tvrzení 1.8.12.** Připomeňme postup, jehož správnost budeme dokazovat.
Máme konečnou množinu klauzulí $S$, kde žádná klauzule není tautologií. Zvolíme jednu logickou proměnnou (označme ji $x$), která se v některé z klauzulí z $S$ vyskytuje.

- Rozdělíme klauzule množiny $S$ do tří skupin:

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

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

  $M_{\neg x}$ se skládá ze všech klauzulí množiny $S$, které obsahují negativní literál $\neg x$.
- Označíme $N$ množinu všech resolvent klauzulí množiny $S$ podle literálu $x$, které nejsou tautologiemi; tj. $N = \{D \mid D = res_x(C, C'), C \in M_x, C' \in M_{\neg x} \text{ a } D \text{ není tautologie}\}$.
- Položíme $S_1 = M_0 \cup N$.

Dokážeme, že $S_1$ je splnitelná právě tehdy, když je splnitelná množina $S$.

**Důkaz:** Předpokládejme, že množina klauzulí $S$ je splnitelná. To znamená, že existuje pravdivostní ohodnocení $u$ takové, že $u(S) = 1$.

Pak $u(M_0) = 1$, protože $M_0 \subseteq S$. Navíc $u(N) = 1$, protože v $N$ jsou jen resolventy klauzulí z množiny $S$ a ty jsou (podle 1.8.7) pravdivé v $u$. Tedy $u(S_1) = 1$ a $S_1$ je splnitelná množina.

Předpokládejme, že množina $S_1$ je splnitelná. Existuje proto ohodnocení $u$ takové, že $u(S_1) = 1$.

Nyní $u(M_0) = 1$, protože $M_0 \subseteq S_1$. Ukážeme, že můžeme zadefinovat hodnotu $u(x)$ tak, aby také obě množiny $M_x$ a $M_{\neg x}$ byly pravdivé v $u$.

Může nastat jeden ze dvou případů

1. Všechny klauzule v $M_x$ jsou pravdivé nezávisle na ohodnocení logické proměnné $x$; jinými slovy, v každé klauzuli z $M_x$ je pravdivý aspoň jeden literál jiný než $x$.

   V tomto případě položme $u(x) := 0$, tj. $u(\neg x) = 1$ a všechny klauzule z $M_{\neg x}$ jsou automaticky pravdivé v $u$, protože obsahují literál $\neg x$. Tedy $u(S) = 1$.

2. Existuje klauzule $C \in M_x$, jejíž všechny literály různé od literálu $x$ jsou nepravdivé; jinými slovy, $u(C \setminus x) = 0$.

   Uvažujme libovolnou klauzuli $C' \in M_{\neg x}$. Ukážeme, že $u(C' \setminus \neg x) = 1$, tj. ukážeme, že $C'$ je pravdivá v $u$ nezávisle na literálu $\neg x$.

   Protože $D = res_x(C, C')$ patří do množiny $N$, je $u(D) = 1$. Přitom $D = (C \setminus x) \vee (C' \setminus \neg x)$. Víme, že $u(C \setminus x) = 0$, musí proto být $u(C' \setminus \neg x) = 1$, a tudíž klauzule $C' \setminus \neg x$ je pravdivá v $u$ nezávisle na hodnotě $x$. To nám umožňuje definovat $u(x) = 1$. Tím jsme dostali ohodnocení, ve kterém jsou pravdivé obě množiny $M_x$ i $M_{\neg x}$; tedy je pravdivá i celá množina $S$.

## 1.9 Přirozená dedukce

V předchozím textu jsme se zabývali sémantickým důsledkem ve výrokové logice. Matematická logika studuje také tzv. logický důsledek, to je otázku, který výrok/formule logicky vyplývá z dané množiny výroků/formulí. Formalizace tohoto pojmu je zhruba následující: Formule $\alpha$ je logickým důsledkem množiny formulí $S$, jestliže je možno ji odvodit z $S$ v některém „správném“ odvozovacím systému. Správný odvozovací systém pro nás bude ten, který je

- *bezesporný* — odvozováním z prázdné množiny předpokladů nevyplývá aspoň jedna formule,
- a *úplný* — vše, co je pravdivé v něm odvodíme, tj. odvodíme každý sémantický důsledek.

V následujícím textu přiblížíme odvozovací systém, který se nazývá přirozená dedukce. Uvědomte si, že se bude jednat jen o velmi stručný částečně „intuitivní“ úvod do problematiky

Zmíníme zde dva odvozovací systémy:

- **Hilbertův systém**: obsahuje tři typy axiomů — formulí, které se „mohou použít“ kdekoli, a jedno jediné odvozovací pravidlo, tzv. *Modus ponens*, kde z formulí $\alpha$ a $\alpha \Rightarrow \beta$ lze vyvodit formuli $\beta$.
- **Přirozená dedukce**, odvozovací systém, který nemá axiomy, ale využívá pomocných předpokladů uvnitř odvození.

**1.9.1 Přirozená dedukce ve výrokové logice.** Systém odvozovacích pravidel přirozené dedukce obsahuje pro každou logickou spojku $\neg$, $\vee$, $\wedge$ a $\Rightarrow$ pravidla. Pravidla, která spojku „zavádí“, tj. umožňují napsat formuli s touto spojkou, druhá spojku „odstraňují“. Prvním pravidlům říkáme *I-pravidla* („I“ pro introduction, tj. zavedení), druhým *E-pravidla* („E“ pro elimination, neboli odstranění).

**1.9.2 Odvození.** Posloupnost formulí $\varphi_1$, $\varphi_2$, ..., $\varphi_n$ se nazývá *odvození z předpokladů $S$* právě tehdy, když

- každá formule $\varphi_i$ je buď předpoklad (tj. $\varphi_i \in S$), nebo je pomocný předpoklad, nebo vznikla z předcházejících formulí pomocí některého odvozovacího pravidla, která popíšeme dále;
- všechny pomocné předpoklady jsou již pasivní, též vysvětlíme dále.

$\square$

**1.9.3 Odvozovací pravidla.** Jsou dány formule $\varphi$, $\psi$ a $\alpha$. Nejprve uvedeme pravidla, která nepotřebují tzv. boxy — „vložená odvození“:

I-pravidla pro $\wedge$: $\dfrac{\varphi,\quad \psi}{\varphi \wedge \psi}$ $\dfrac{\varphi,\quad \psi}{\psi \wedge \varphi}$ E-pravidla pro $\wedge$: $\dfrac{\varphi \wedge \psi}{\varphi}$ $\dfrac{\varphi \wedge \psi}{\psi}$

E-pravidlo pro $\Rightarrow$ (také zvané Modus Ponens): $\dfrac{\varphi,\quad \varphi \Rightarrow \psi}{\psi}$

I-pravidla pro $\vee$: $\dfrac{\varphi}{\varphi \vee \psi}$ $\dfrac{\varphi}{\psi \vee \varphi}$ E-pravidlo pro $\neg$: $\dfrac{\neg\neg\varphi}{\varphi}$

Další pravidla vyžadují vložená odvození, též „boxy“. Boxy se používají tak, že box začíná tzv. pomocným předpokladem, pro každý box pouze jeden pomocný předpoklad, který uvnitř celého boxu můžeme „volně“ používat. Box je možné uzavřít pouze na základě některého pravidla, které boxy obsahuje. Tím, že toto pravidlo použijeme, stává se pomocný předpoklad „pasivní“ a box uzavíráme.

První z nich je

I-pravidlo pro $\Rightarrow$: $$\dfrac{\boxed{\begin{array}{c}\varphi\\ \vdots\\ \psi\end{array}}}{\varphi \Rightarrow \psi}$$

Při používání I-pravidla pro $\Rightarrow$ budeme říkat, že $\varphi$ je pomocný předpoklad *aktivní* uvnitř celého pododvození. Dospějeme-li k závěru (tj. formuli $\varphi \Rightarrow \psi$), řekneme, že jsme tento pomocný předpoklad *eliminovali* a on se stal *pasivním* pomocným předpokladem.

Další pravidla s pododvozeními jsou

E-pravidlo pro $\vee$:

$$\frac{\varphi \vee \psi, \quad \boxed{\begin{array}{c}\varphi \\ \vdots \\ \alpha\end{array}}\,, \quad \boxed{\begin{array}{c}\psi \\ \vdots \\ \alpha\end{array}}}{\alpha} \qquad \text{nebo} \qquad \frac{\varphi \vee \psi, \quad \varphi \Rightarrow \alpha, \quad \psi \Rightarrow \alpha}{\alpha}$$

I-pravidlo pro $\neg$:

$$\frac{\boxed{\begin{array}{c}\varphi \\ \vdots \\ \alpha \wedge \neg\alpha\end{array}}}{\neg\varphi}$$

**1.9.4 Logický důsledek.** Formule $\varphi$ je *logický důsledek množiny formulí $S$* (říkáme jim *předpoklady*), též *logicky vyplývá z $S$*, jestliže existuje odvození z $S$ takové, že $\varphi_n = \varphi$. Zapisujeme $S \vdash \varphi$. □

**1.9.5 Theorém.** Formule $\alpha$ se nazývá *theorém*, jestliže je logickým důsledkem prázdné množiny předpokladů. □

**1.9.6 Věta o úplnosti.** Pro každou množinu formulí $S$ a formuli $\varphi$ platí

$$S \vdash \varphi \quad \text{právě tehdy, když} \quad S \models \varphi.$$

□

**1.9.7 Poznámka.** Ukázat, že formule $\alpha$, která má odvození z předpokladů $S$, je sémantickým důsledkem množiny $S$, není obtížné. Je to proto, že závěr každého pravidla přirozené dedukce je sémantickým důsledkem předpokladů. Toto ukazuje, že přirozená dedukce je bezesporným odvozovacím systémem.

Druhou implikaci, totiž, že každý sémantický důsledek $\alpha$ množiny formulí $S$ má též odvození z $S$, je daleko těžší dokázat. Důkaz je nepřímý; ukáže se, že není možné, aby odvození neměl. Tato implikace pak říká, že přirozená dedukce je úplný odvozovací systém.

**1.9.8** Z věty o úplnosti také vyplývá následující tvrzení.
**Tvrzení.** Formule je theorém právě tehdy, když je to tautologie. □