Kapitola 1

Úvod do logiky

Slovo *logika* se používá ve dvou významech:

- soubor principů, které jsou základem uspořádání prvků nějakého systému (např. počítačového programu, elektronického zařízení, komunikačního protokolu)
- uvažování prováděné podle striktních pravidel zachovávajících platnost

V informatice se tyto dva významy setkávají: nejprve formálně popíšeme daný systém, a poté o něm *formálně uvažujeme* (v praktických aplikacích automaticky), tj. odvozujeme platné *inference* o systému, za použití nějakého *(formálního) dokazovacího systému*.

Mezi praktické aplikace logiky v informatice patří například software verification, logic programming, SAT solving, automated reasoning, nebo knowledge-based representation. Kromě toho je logika (ve větší míře než matematika) základním nástrojem pro popis teoretické informatiky. Více o aplikacích logiky najdete v příloze A; v příloze B je shrnuta historie logiky.

1.1 Výroková logika

Nyní si ukážeme logiku v akci na dvou příkladech ze života (hledače pokladů, a teoretického informatika):

1.1.1 Příklad: Hledání pokladu

Příklad 1.1.1. Při hledání pokladu v dračí sluji jsme narazili na rozcestí dvou chodeb. Víme, že na konci každé chodby je buď poklad, nebo drak, ale ne obojí. Trpaslík, kterého jsme na rozcestí potkali, nám řekl, že: "Alespoň jedna z těch dvou chodeb vede k pokladu", a po dalším naléhání (a menším úplatku), ještě řekl, že "První chodba vede k drakovi." Je známo, že trpaslíci, které člověk potká v dračí sluji, buď vždy mluví pravdu, nebo vždy lžou. Kterou cestou se máme vydat?

1.1.2 Formalizace ve výrokové logice

Začneme tím, že situaci a naše znalosti formalizujeme ve výrokové logice. *Výrok* je tvrzení, kterému můžeme přiřadit pravdivostní hodnotu: *pravdivý* (*True*, 1), nebo *lživý* (*False*, 0).

Některé výroky lze vyjádřit pomocí jednodušších výroků a logických spojek, např. "(Trpaslík lže,) *právě když* (druhá chodba vede k drakovi.)" nebo "(První chodba vede k pokladu) *nebo* (první chodba vede k drakovi.)" Pokud výrok takto rozložit nelze, říká se mu *prvovýrok*, *atomický výrok*, nebo *výroková proměnná*.

Popíšeme tedy celou situaci pomocí *výrokových proměnných*. Můžeme si je také představit jako jednoduché zjišťovací (ano/ne) otázky, na které musíme znát odpověď, abychom věděli vše o dané situaci. Jako naše výrokové proměnné zvolme "Poklad je v první chodbě." (označme p_1), a "Poklad je v druhé chodbě." (p_2). V úvahu přichází i jiné výrokové proměnné, např. "V první chodbě je drak." (d_1) nebo "Trpaslík mluví pravdu." (t). Ty lze ale vyjádřit pomocí $\{p_1, p_2\}$, např. platí t, právě když neplatí p_1. Tj. známe-li pravdivostní hodnoty p_1, p_2, pravdivostní hodnoty d_1, t jsou jednoznačně určené. A menší počet výrokových proměnných znamená menší prohledávací prostor.

Vyjádříme tedy všechny naše znalosti jako (*složené*) výroky a zapíšeme je ve formálním zápisu v *jazyce* výrokové logiky nad množinou prvovýroků $\mathbb{P} = \{p_1, p_2\}$, za použití symbolů reprezentujících logické spojky $\neg$ ("neplatí X", *negace*), $\wedge$ ("X a Y", *konjunkce*), $\vee$ ("X nebo Y", *disjunkce*), $\rightarrow$ ("pokud X, potom Y", *implikace*), $\leftrightarrow$ ("X, právě když Y", *ekvivalence*) a závorek (,). Zde je dobré zmínit, že disjunkce není exkluzivní, tj. "X nebo Y" platí i pokud platí jak X tak Y, a implikace je čistě logická: "pokud X, potom Y" platí kdykoliv neplatí X nebo platí Y.

Informace o tom, že v chodbě je poklad nebo drak, ale ne obojí, už je zakódovaná v naší volbě výrokových proměnných: přítomnost draka je totéž co absence pokladu. Tvrzení trpaslíka, že "První chodba vede k drakovi." tedy vyjádříme jako "Neplatí, že poklad je v první chodbě.", formálně $\neg p_1$. Tvrzení, že "Alespoň jedna z těch dvou chodeb vede k pokladu." vyjádříme jako "Poklad je v první chodbě nebo poklad je v druhé chodbě.", formálně $p_1 \vee p_2$. Informaci, že trpaslíci buď vždy mluví pravdu, nebo vždy lžou, si přeložíme tak, že buď platí oba naše výroky, nebo platí negace obou našich výroků, formálně:

$$(\neg p_1 \wedge (p_1 \vee p_2)) \vee (\neg(\neg p_1) \wedge \neg(p_1 \vee p_2))$$

Označme tento výrok jako φ (od slova "formule", výrokům se někdy také říká *výrokové formule*). V našem příkladě lze všechny informace vyjádřit jediným výrokem, v praxi ale často potřebujeme výroků více, někdy i nekonečně mnoho (například pokud chceme popsat výpočet nějakého programu a nevíme apriori kolik kroků bude mít), potom popíšeme situaci pomocí množiny výroků, tzv. *teorie*, zde $T = \{\varphi\}$. Výrokům z T říkáme také *axiomy* teorie T[1].

1.1.3 Modely a důsledky

Jsou naše informace dostačující k určení, zda je v některé z chodeb poklad? Jinými slovy, ptáme se, zda je jeden z výroků p_1, p_2 logickým *důsledkem* výroku φ, resp. teorie T. Co to znamená?

Představme si, že existuje více různých "světů" lišících se v tom, co je na konci první a na konci druhé chodby. Například, v jednom ze světů je na konci první chodby poklad a na konci druhé chodby drak. Tento svět můžeme popsat pomocí pravdivostního ohodnocení výrokových proměnných: $p_1 = 1, p_2 = 0$. Takovému ohodnocení říkáme *model* jazyka $\mathbb{P} = \{p_1, p_2\}$ a zapisujeme ho zkráceně jako $v = (1, 0)$ (v od slova "valuation"). Celkem tedy máme čtyři

[1] Terminologie v logice často pochází z jejích aplikace v matematice, viz příloha B o historii logiky.

různé světy, popsané *modely jazyka*

$$\mathrm{M}_{\mathbb{P}} = \{(0,0),(0,1),(1,0),(1,1)\}.$$

Je svět popsaný modelem $v = (1,0)$ konzistentní s informacemi, které máme, tj. *platí* v něm výrok φ, resp. teorie T? Pravdivostní hodnotu (složeného) výroku φ v modelu v, označme ji $v(\varphi)$, můžeme snadno zjistit: Víme, že $v(p_1) = 1$ a $v(p_2) = 0$, takže $v(\neg p_1) = 0$, a také $v(\neg p_1 \wedge (p_1 \vee p_2)) = 0$ (jde o konjunkci dvou výroků, a první z konjunktů je v modelu v nepravdivý). Podobně $v(p_1 \vee p_2) = 1$ (neboť $v(p_1) = 1$), takže $v(\neg(p_1 \vee p_2)) = 0$, a $v(\neg(\neg p_1) \wedge \neg(p_1 \vee p_2)) = 0$. Výrok φ je disjunkcí dvou výroků, z nichž ani jeden v modelu v neplatí, tedy $v(\varphi) = 0$.

Bystrý čtenář jistě vidí stromovou strukturu výroku φ a postupné vyhodnocování $v(\varphi)$ od listů směrem ke kořeni. Formální definice představíme v příští kapitole.

Podobně určíme pravdivostní hodnoty výroku φ v ostatních modelech. Zjistíme, že množina *modelů výroku* φ (resp. *modelů teorie* T), tj. množina všech modelů jazyka, ve kterých platí φ (resp. všechny výroky z teorie T), je

$$\mathrm{M}_{\mathbb{P}}(\varphi) = \mathrm{M}_{\mathbb{P}}(T) = \{(0,1)\}.$$

Vidíme, že naše informace jednoznačně určují model $(0,1)$, tedy svět, ve kterém je v první chodbě drak a ve druhé poklad. Obecně modelů může být více, stačí nám vědět, že v každém modelu φ, resp. T, platí výrok p_2, tedy že p_2 je *důsledkem* teorie T; také říkáme, že p_2 *platí* v teorii T.

1.1.4 Dokazovací systémy

Postup, který jsme zvolili, je velmi neefektivní. Máme-li n výrokových proměnných[2], existuje 2^n modelů jazyka a není prakticky možné ověřovat platnost teorie v každém z nich. Na řadu přichází tzv. *dokazovací systémy*. V daném dokazovacím systému je *důkaz* výroku ψ z teorie T přesně, formálně definovaný syntaktický objekt, který v sobě zahrnuje snadno (mechanicky) ověřitelný "důkaz" (důvod), proč ψ platí v T, a který můžeme hledat (pomocí počítače) čistě na základě struktury výroku ψ a axiomů teorie T ("syntaxe"), tj. aniž bychom se museli zabývat modely ("sémantikou").

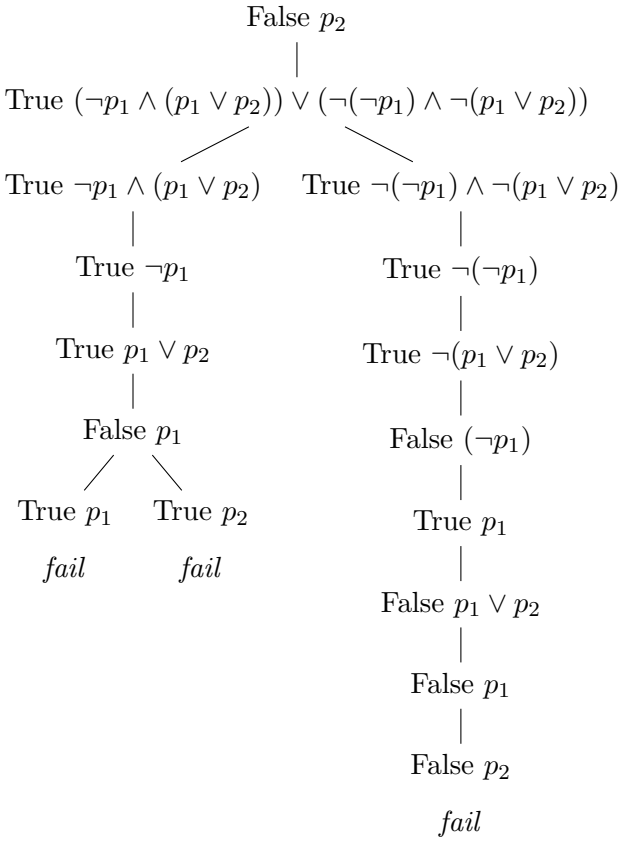
Po důkazovém systému chceme dvě vlastnosti:

- *korektnost*, tj. pokud máme důkaz ψ z T, potom ψ platí v T, a
- *úplnost*, tj. pokud ψ platí v T, potom existuje důkaz ψ z T,

přičemž korektnost je nutností (bez ní nemá smysl důkazy hledat), a úplnost je dobrá vlastnost, ale efektivní důkazový systém může být užitečný, i pokud v něm nelze dokázat vše, co platí.

Zde stručně nastíníme dva důkazové systémy: *metodu analytického tabla* a *rezoluční metodu*. Později budou představeny formálně, a u obou si dokážeme i jejich korektnost a úplnost. Oba tyto důkazové systémy jsou založeny na *důkazu sporem*, tj. předpokládají platnost axiomů z T a negace výroku ψ, a snaží se dojít ke sporu.

[2] V praxi máme běžně tisíce proměnných.



Obrázek 1.1: Tablo důkaz výroku p_2 z teorie T

1.1.5 Tablo metoda

V metodě analytického tabla je důkazem *tablo*: strom jehož vrcholy jsou označkované předpoklady o platnosti výroků. Podívejme se na příklad tabla na obrázku 1.1.

Začneme předpokladem, že neplatí výrok p_2 (neboť dokazujeme sporem). Poté připojíme platnost všech axiomů teorie T (v našem případě je jen jeden: výrok φ sestrojený výše). Dále budujeme tablo tak, že zjednodušujeme výroky v předpokladech, a to podle jistých pravidel, která zaručují následující invariant:

> *Každý model teorie T, ve kterém neplatí p_2, se musí* shodovat *s některou z větví tabla (tj. splňovat všechny předpoklady na dané větvi).*

Výrok φ je disjunkcí dvou výroků, $\varphi = \varphi_1 \vee \varphi_2$. Pokud tedy platí v nějakém modelu, potom buď v tomto modelu platí φ_1, nebo v něm platí φ_2. Rozvětvíme strom podle těchto dvou možností. V dalším kroku máme předpoklad o pravdivosti výroku $\neg p_1 \wedge (p_1 \vee p_2)$. V tom případě musí platit jak $\neg p_1$, tak $p_1 \vee p_2$, připojíme tedy na konec větve oba tyto předpoklady. Pravdivost $\neg p_1$ znamená lživost p_1, a tak dále.

Takto postupujeme, dokud je možné výroky v předpokladech zjednodušit, tj. dokud to nejsou jen výrokové proměnné. Pokud na jedné větvi najdeme dvojici opačných předpokladů o nějakém výroku ψ, tj. že platí, a zároveň že neplatí, víme, že se s touto větví nemůže

shodovat žádný model. Takové větvi říkáme *sporná*. Protože dokazujeme sporem, důkaz je takové tablo, ve kterém je každá větev sporná. Tím je zaručeno, že neexistuje model T, ve kterém neplatí p_2. Z toho plyne, že p_2 platí v každém modelu T, neboli je to důsledek T, což jsme chtěli dokázat.

Zatím se spokojíme s pochopením základní myšlenky této metody, detaily představíme v budoucí kapitole.

1.1.6 Rezoluční metoda

Není těžké naprogramovat systematické hledání tablo důkazu. V praxi se ale používá jiný důkazový systém, který má mnohem jednodušší a efektivnější implementaci: tzv. *rezoluční metoda*. Tato metoda pochází z roku 1965, a je základem většiny *systémů automatického dokazování*, *SAT solverů*, nebo třeba interpreterů jazyka Prolog (o kterém si budeme povídat později).

Rezoluční metoda je založena na faktu, že každý výrok lze ekvivalentně vyjádřit ve speciálním tvaru, v tzv. *konjunktivní normální formě (CNF)*. *Literál* je výroková proměnná p nebo její negace $\neg p$ (tj. literály jsou výroky, které jen určují hodnotu jedné výrokové proměnné). Disjunkci několika literálů, např. $p \vee \neg q \vee \neg r$, říkáme *klauzule*. A výrok je v CNF, pokud je konjunkcí klauzulí. Ke každému výroku ψ existuje *ekvivalentní* výrok ψ' v CNF. Ekvivalentní znamená mající stejný význam (stejné modely), píšeme $\psi \sim \psi'$. Později si ukážeme dvě metody převodu do CNF, nyní jen na našem příkladě: ve výroku

$$(\neg p_1 \wedge (p_1 \vee p_2)) \vee (\neg(\neg p_1) \wedge \neg(p_1 \vee p_2))$$

nejprve nahradíme $\neg(\neg p_1) \sim p_1$ a $\neg(p_1 \vee p_2) \sim (\neg p_1 \wedge \neg p_2)$:

$$(\neg p_1 \wedge (p_1 \vee p_2)) \vee (p_1 \wedge \neg p_1 \wedge \neg p_2)$$

a dále opakovaně použijeme *distributivitu* $\vee$ vůči $\wedge$ (představte si, že $\vee$ je operace násobení a $\wedge$ je operace sčítání):

$$(\neg p_1 \vee p_1) \wedge (\neg p_1 \vee \neg p_1) \wedge (\neg p_1 \vee \neg p_2) \wedge (p_1 \vee p_2 \vee p_2) \wedge (p_1 \vee p_2 \vee \neg p_1) \wedge (p_1 \vee p_2 \vee \neg p_2)$$

Tento výrok už je v CNF, ale dále ho zjednodušíme: vynecháme z klauzulí duplicitní literály, a uvědomíme si, že obsahuje-li klauzule dvojici opačných literálů $p, \neg p$, je to *tautologie* (platí v každém modelu) a proto ji můžeme odstranit. Dostáváme CNF výrok

$$\neg p_1 \wedge (\neg p_1 \vee \neg p_2) \wedge (p_1 \vee p_2)$$

který je ekvivalentní původnímu výroku ϕ Protože chceme dokázat p_2 sporem, přidáme ještě klauzuli $\neg p_2$:

$$\neg p_1 \wedge (\neg p_1 \vee \neg p_2) \wedge (p_1 \vee p_2) \wedge \neg p_2$$

Výrok p_2 platí v teorii T, právě když je tento CNF výrok *nesplnitelný* (nemá žádný model). Výroky v CNF budeme zapisovat také v *množinové reprezentaci*[3]:

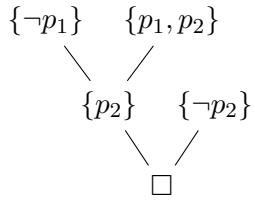
$$\{\{\neg p_1\}, \{\neg p_1, \neg p_2\}, \{p_1, p_2\}, \{\neg p_2\}\}$$

[3] V praktické implementaci bychom mohli použít seznam klauzulí, kde každá klauzule je seznam (unikátních) literálů v nějakém zvoleném uspořádání. Představte si opět tisíce výrokových proměnných a klauzulí.

Rezoluční pravidlo říká, že pokud máme dvojici klauzulí, z nichž jedna obsahuje literál p a druhá opačný literál $\neg p$, potom z nich logicky plyne také jejich *rezolventa*: klauzule vzniklá odstraněním literálu p z první a $\neg p$ z druhé klauzule a sjednocením zbylých literálů. Například, z $p \vee \neg q \vee \neg r$ a $\neg p \vee \neg q \vee s$ můžeme odvodit rezolventu $\neg q \vee \neg r \vee s$. *Rezoluční zamítnutí* formule v CNF je potom posloupnost klauzulí, která končí *prázdnou klauzulí* $\square$ (znamenající spor) a kde každá klauzule je buď z dané formule, nebo je rezolventou nějakých dvou předchozích klauzulí. V našem případě:

$$\{\neg p_1\}, \{p_1, p_2\}, \{p_2\}, \{\neg p_2\}, \square$$

Třetí klauzule je rezolventou první a druhé, pátá je rezolventou třetí a čtvrté. Rezoluci lze také přirozeně znázornit pomocí *rezolučního stromu*, kde listy jsou klauzule z dané formule, a vnitřní vrcholy jsou rezolventy svých potomků:

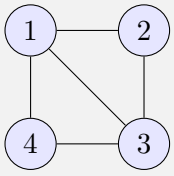


Pokud by formule měla model, musely by v něm platit její klauzule, tedy i postupně všechny rezolventy v posloupnosti, a nakonec prázdná klauzule. Ta ale neplatí v žádném modelu. Máme tedy důkaz sporem a víme, že ve druhé chodbě najdeme poklad.

1.1.7 Příklad: Barvení grafů

Ve druhém příkladu se trochu přiblížíme aplikacím. Na rozdíl od logických hádanek ve formě slovních úloh, různé varianty problému barvení grafů se objevují v rozmanitých úlohách z praxe, od rozvrhovacích problémů přes návrh fyzických a síťových systémů po zpracování obrazu.

Příklad 1.1.2. Najděte vrcholové obarvení následujícího grafu třemi barvami, tj. přiřaďte vrcholům barvy R, G, B tak, aby žádná hrana nebyla monochromatická.



Graf si reprezentujeme jako množinu vrcholů a množinu hran, kde každá hrana je dvojice vrcholů. Lépe se nám bude pracovat s uspořádanými dvojicemi, zvolíme tedy (libovolnou) orientaci hran.

$$\mathcal{G} = \langle V; E \rangle = \langle \{1,2,3,4\}; \{(1,2),(1,3),(1,4),(2,3),(3,4)\} \rangle$$

Začneme opět formalizací ve výrokové logice. Označme si množinu barev jako $C = \{R, G, B\}$. Přirozená volba výrokových proměnných je "vrchol v má barvu c", označme p_v^c, pro každý

vrchol $v \in V$ a každou barvu $c \in C$. Naše (uspořádaná) množina výrokových proměnných má 12 prvků:

$$\mathbb{P} = \{p_v^c \mid v \in V, c \in C\} = \{p_1^R, p_1^G, p_1^B, p_2^R, p_2^G, p_2^B, p_3^R, p_3^G, p_3^B, p_4^R, p_4^G, p_4^B\}$$

Máme celkem $|\,\mathrm{M}_{\mathbb{P}}\,| = 2^{12} = 4096$ modelů jazyka (reprezentovaných 12-dimenzionálními 0–1 vektory). Většinu z nich není nelze interpretovat jako obarvení grafu. Například $v = (1, 1, 0, 0, \ldots, 0)$ říká, že vrchol 1 je obarvený červeně, a také zeleně. Začneme tedy teorií vyjadřující, že každý vrchol má nejvýše jednu barvu. Existuje více způsobů, jak to můžeme vyjádřit. My řekneme pro každý vrchol, že nesmí mít (alespoň) jednu z každé dvojice barev. Tím dostaneme teorii v CNF:

$$\begin{aligned} T_1 = \{&(\neg p_1^R \vee \neg p_1^G) \wedge (\neg p_1^R \vee \neg p_1^B) \wedge (\neg p_1^G \vee \neg p_1^B),\\ &(\neg p_2^R \vee \neg p_2^G) \wedge (\neg p_2^R \vee \neg p_2^B) \wedge (\neg p_2^G \vee \neg p_2^B),\\ &(\neg p_3^R \vee \neg p_3^G) \wedge (\neg p_3^R \vee \neg p_3^B) \wedge (\neg p_3^G \vee \neg p_3^B),\\ &(\neg p_4^R \vee \neg p_4^G) \wedge (\neg p_4^R \vee \neg p_4^B) \wedge (\neg p_4^G \vee \neg p_4^B)\}\\ = \{&(\neg p_v^R \vee \neg p_v^G) \wedge (\neg p_v^R \vee \neg p_v^B) \wedge (\neg p_v^G \vee \neg p_v^B) \mid v \in V\} \end{aligned}$$

Teorii T_1 bychom mohli říkat teorie částečných vrcholových obarvení grafu $\mathcal{G}$. Teorie T_1 má $|\,\mathrm{M}_{\mathbb{P}}(T_1)| = 4^4 = 2^8 = 256$ modelů. (Proč?) Pokud chceme úplná obarvení, přidáme podmínku, že každý vrchol má alespoň jednu barvu.[4]

$$\begin{aligned} T_2 &= T_1 \cup \{p_1^R \vee p_1^G \vee p_1^B, p_2^R \vee p_2^G \vee p_2^B, p_3^R \vee p_3^G \vee p_3^B, p_4^R \vee p_4^G \vee p_4^B\}\\ &= T_1 \cup \{p_v^R \vee p_v^G \vee p_v^B \mid v \in V\}\\ &= T_1 \cup \{\bigvee_{c \in C} p_v^c \mid v \in V\} \end{aligned}$$

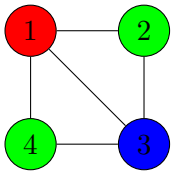
Teorie T_2 má $3^4 = 81$ modelů. Jde o *extenzi* teorie T_1, neboť každý důsledek teorie T_1 platí i v teorii T_2. Platí dokonce, že $\mathrm{M}_{\mathbb{P}}(T_2) \subseteq \mathrm{M}_{\mathbb{P}}(T_1)$.[5] Zbývá přidat podmínku zakazující monochromatické hrany. Pro každou hranu a každou barvu specifikujeme, že alespoň jeden z vrcholů hrany nesmí mít danou barvu. Zde pro názornost naposledy napíšeme úplný seznam výroků, nadále budeme využívat zkráceného zápisu.

$$\begin{aligned} T_3 = T_2 \cup \{&(\neg p_1^R \vee \neg p_2^R) \wedge (\neg p_1^G \vee \neg p_2^G) \wedge (\neg p_1^B \vee \neg p_2^B),\\ &(\neg p_1^R \vee \neg p_3^R) \wedge (\neg p_1^G \vee \neg p_3^G) \wedge (\neg p_1^B \vee \neg p_3^B),\\ &(\neg p_1^R \vee \neg p_4^R) \wedge (\neg p_1^G \vee \neg p_4^G) \wedge (\neg p_1^B \vee \neg p_4^B),\\ &(\neg p_2^R \vee \neg p_3^R) \wedge (\neg p_2^G \vee \neg p_3^G) \wedge (\neg p_2^B \vee \neg p_3^B),\\ &(\neg p_3^R \vee \neg p_4^R) \wedge (\neg p_3^G \vee \neg p_4^G) \wedge (\neg p_3^B \vee \neg p_4^B)\}\\ = T_2 \cup \{&\bigwedge_{c \in C} (\neg p_u^c \vee \neg p_v^c) \mid (u, v) \in E\} \end{aligned}$$

[4]Symbol $\bigvee$ používáme podobně jako symboly $\sum$ pro součet a $\prod$ pro součin: ke zjednodušení zápisu výroku, který je ve formě disjunkce. Např. pokud $v = 1$, potom $\bigvee_{c \in C} p_v^c$ reprezentuje výrok $p_1^R \vee p_1^G \vee p_1^B$. Analogicky $\bigwedge$ pro konjunkci.

[5]Zde vidíme typickou ukázku antimonotónního vztahu tzv. Galoisovy korespondence: čím více vlastností (výroků) požadujeme, tím méně objektů (modelů) splňuje tyto vlastnosti.

Výsledná teorie T_3 je *splnitelná* (má model), právě když graf $\mathcal{G}$ je 3-obarvitelný. Má 6 modelů jednoznačně odpovídajících 3-obarvení grafu $\mathcal{G}$. Model $v = (1,0,0,0,1,0,0,0,1,0,1,0)$ odpovídá následujícímu obarvení, ostatní obarvení získáme permutací barev.



Jakmile máme teorii T_3 formalizující 3-obarvení grafu $\mathcal{G}$, můžeme snadno řešit související otázky, například najít všechna obarvení, ve kterých vrchol 1 je modrý a vrchol 2 zelený: odpovídají modelům teorie $T_3 \cup \{p_1^B, p_2^G\}$. Nebo můžeme dokázat, že vrcholy 2 a 4 musejí být obarveny stejnou barvou. Můžeme použít tablo metodu: v kořeni tabla bude předpoklad

$$\text{False } (p_2^R \wedge p_2^R) \vee (p_2^G \wedge p_2^G) \vee (p_2^B \wedge p_2^B)$$

Nebo můžeme najít rezoluční zamítnutí teorie vzniklé převedením axiomů teorie T_3 do CNF, a přidáním CNF ekvivalentu *negace* výroku $(p_2^R \wedge p_2^R) \vee (p_2^G \wedge p_2^G) \vee (p_2^B \wedge p_2^B)$ (neboť jde o důkaz sporem, tedy pro spor předpokládáme, že *nemají* stejnou barvu).

1.2 Predikátová logika

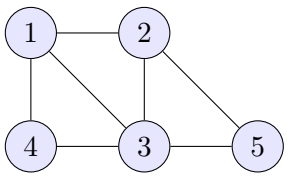
Nyní si velmi stručně a neformálně představíme *predikátovou logiku*. Predikátová logika se zabývá vlastnostmi objektů a vztahy mezi objekty. Například:

Všichni lidé jsou smrtelní.
Sókratés je člověk.
Sókratés je smrtelný.

Ve skutečnosti výroková logika vznikla později (asi o století) než Aristotelova predikátová logika, a byla poté na dlouho převážně zapomenuta.

1.2.1 Nevýhody formalizace ve výrokové logice

Nevýhodou formalizace našeho problému ve výrokové logice je fakt, že výsledná teorie T_3 je poměrně velká, a navíc byla vytvořena ad hoc pro graf $\mathcal{G}$. Představme si, že potřebujeme graf $\mathcal{G}$ změnit, například přidáním vrcholu 5 spojeného hranami s vrcholy 2 a 3:



Abychom byli schopni formalizovat nový problém, musíme přidat do našeho jazyka tři nové výrokové proměnné: $\mathbb{P}' = \mathbb{P} \cup \{p_5^R, p_5^G, p_5^B\}$, a vytvořit nové teorie T_1', T_2', T_3' přidáním axiomů týkajících se vrcholu 5 a hran $(2,5), (3,5)$. Problémem je, že strukturu grafu $\mathcal{G}$ a přirozené vlastnosti jako "z vrcholu u vede hrana do vrcholu v", nebo "vrchol u je zelený" jsme ('natvrdo', 'nepřirozeně') zakódovali do 0–1 proměnných. Tento nedostatek odstraňuje *predikátová logika*.

1.2.2 Stručné představení predikátové logiky

Modelem v predikátové logice není 0–1 vektor, ale *struktura*. Příkladem struktur jsou naše (orientované) grafy:

$$\mathcal{G} = \langle V^{\mathcal{G}}; E^{\mathcal{G}} \rangle = \langle \{1,2,3,4\}; \{(1,2),(1,3),(1,4),(2,3),(3,4)\} \rangle$$
$$\mathcal{G}' = \langle V^{\mathcal{G}'}; E^{\mathcal{G}'} \rangle = \langle \{1,2,3,4,5\}; \{(1,2),(1,3),(1,4),(2,3),(3,4),(2,5),(3,5)\} \rangle$$

Oba grafy sestávají z množiny vrcholů, a z binární relace na této množině. Jde o struktury v *jazyce teorie grafů* $\mathcal{L} = \langle E \rangle$, kde E je binární *relační symbol*. *Jazyk* predikátové logiky specifikuje jaké relace (kolik a jakých arit — unární, binární, ternární, atd.) mají struktury mít, a jaké symboly pro ně budeme používat. Kromě toho používáme symbol rovnosti $=$ a struktury mohou obsahovat také (jako například funkce $+, -, \cdot$ v *tělese* reálných čísel), ty si ale necháme na později.

Predikátová logika používá tytéž logické spojky jako výroková logika, ale základním stavebním kamenem *predikátových formulí* nejsou výrokové proměnné, nýbrž tzv. *atomické formule*, například: $E(x, y)$ představuje tvrzení, že v grafu vede hrana z vrcholu x do vrcholu y. Zde x, y jsou *proměnné* reprezentující vrcholy daného grafu. Kromě toho ve formulích můžeme používat *kvantifikátory*: $(\forall x)$ "pro všechny vrcholy x" a $(\exists y)$ "existuje vrchol y".[6]

Nyní můžeme formalizovat tvrzení, která dávají smysl pro libovolný graf. Například:

- "V grafu nejsou smyčky": $(\forall x)(\neg E(x, x))$
- "Existuje vrchol výstupního stupně 1": $(\exists x)(\forall y)(\forall z)((E(x, y) \wedge E(x, z)) \rightarrow y = z)$

V daném grafu $\mathcal{G}$ a při dosazení vrcholu u za proměnnou x a vrcholu v za proměnnou y vyhodnotíme $E(x, y)$ jako True, právě když $(u, v) \in E^{\mathcal{G}}$.

1.2.3 Formalizace barvení grafů v predikátové logice

Vraťme se zpět k barvení grafů. Přirozený způsob jak formalizovat náš problém 3-obarvitelnosti je v jazyce $\mathcal{L}' = \langle E, R, G, B \rangle$, kde E je binární a R, G, B jsou unární relační symboly, tedy $R(x)$ znamená "vrchol x je červený". Strukturou pro tento jazyk je (orientovaný) graf spolu s trojicí množin vrcholů, např.

$$\begin{aligned}\mathcal{G}_C &= \langle V^{\mathcal{G}_C}; E^{\mathcal{G}_C}, R^{\mathcal{G}_C}, G^{\mathcal{G}_C}, B^{\mathcal{G}_C} \rangle \\ &= \langle \{1,2,3,4\}; \{(1,2),(1,3),(1,4),(2,3),(3,4)\}, \{1\}, \{2,3\}, \{4\} \rangle\end{aligned}$$

reprezentuje graf $\mathcal{G}$ s validním obarvením z obrázku výše. Budeme říkat, že $\mathcal{G}_C$ je *expanze* $\mathcal{L}$-struktury $\mathcal{G}$ do jazyka $\mathcal{L}'$.

Podobně jako ve výrokové logice musíme nejprve zajistit, aby naše modely reprezentovaly obarvené grafy. Začneme požadavkem, aby každý vrchol byl obarven nejvýše jednou barvou:

$$(\forall x)((\neg R(x) \vee \neg G(x)) \wedge (\neg R(x) \vee \neg B(x)) \wedge (\neg G(x) \vee \neg B(x)))$$

Obarvení alespoň jednou barvou vyjádříme takto:

$$(\forall x)(R(x) \vee G(x) \vee B(x))$$

A hranovou podmínku formalizujeme pomocí predikátu $E(x, y)$ například takto:

$$(\forall x)(\forall y)(E(x, y) \rightarrow ((\neg R(x) \vee \neg R(y)) \wedge (\neg G(x) \vee \neg G(y)) \wedge (\neg B(x) \vee \neg B(y))))$$

Modely takto vzniklé teorie reprezentují orientované grafy s vrcholovým 3-obarvením.

[6] Kvantifikátory si můžeme představit jako "konjunkci" resp. "disjunkci" přes všechny vrcholy grafu.

1.3 Další druhy logických systémů

Predikátové logice, kde proměnné reprezentují jednotlivé vrcholy, říkáme logika *prvního řádu* (anglicky *first-order (FO) logic*). V logice *druhého řádu* (anglicky *second-order (SO) logic*) máme také proměnné reprezentující množiny vrcholů nebo i množiny n-tic vrcholů (tj. relace, funkce). Například tvrzení, že každá neprázdná zdola omezená podmnožina má infimum[7], můžeme formalizovat v logice druhého řádu takto:[8]

$$(\forall S)((\exists x)S(x) \wedge (\exists x)(\forall y)(S(y) \to x \leq y) \to$$
$$(\exists x)((\forall y)(S(y) \to x \leq y) \wedge (\forall z)((\forall y)(S(y) \to z \leq y) \to z \leq x)))$$

V logice třetího řádu máme i proměnné reprezentující množiny množin (což je užitečné např. v topologii), atd.

Kromě výrokové a predikátové logiky existují i další typy logických systémů, například intuicionistická logika (která povoluje jen konstruktivní důkazy), temporální logiky (kde mluvíme o platnosti 'vždy', 'někdy v budoucnosti', 'dokud' apod.), modální logiky ('je možné', 'je nutné'), nebo fuzzy logiky (kde máme výroky '0.35 pravdivé'). Tyto logiky mají důležité aplikace v informatice, např. v umělé inteligenci (modální logiky pro uvažování autonomních agentů o svém okolí), v paralelním programování (temporální logiky), nebo v automatických pračkách (fuzzy logiky). Více viz Příloha C. V tomto kurzu se omezíme na výrokovou logiku a predikátovou logiku prvního řádu.

1.4 O přednášce

Přednášku lze rozdělit do třech částí.

První část se zabývá výrokovou logikou. Nejprve představíme syntaxi a sémantiku, dále problém splnitelnosti CNF formulí (známý NP-úplný problém SAT) a jeho polynomiálně řešitelné fragmenty (2-SAT, Horn-SAT). Budeme pokračovat tablo metodu, u níž si dokážeme korektnost a úplnost, a také několik aplikací, například Větu u kompaktnosti. A na závěr představíme rezoluční metodu ve výrokové logice a její aplikaci v programovacím jazyce Prolog.

Ve druhé části představíme predikátovou logiku. Začneme opět syntaxí a sémantiku,ukážeme si jak lze adaptovat tablo metodu pro predikátovou logiku, opět včetně několika aplikací, a skončíme znovu rezoluční metodou.

Třetí část je úvodem do teorie modelů, definovatelnosti, axiomatizovatelnosti, a algoritmické rozhodnutelnosti. Na závěr si představíme slavné Gödelovy věty o neúplnosti, které ukazují meze formální metody (formální dokazatelnosti v axiomatickém systému).

[7] Což platí v uspořádané množině reálných čísel, ale neplatí v racionálních číslech, např. $\{x \mid x^2 > 2, x > 0\}$.

[8] I když je tato formule velmi složitá, pokuste se porozumět jednotlivým částem.