První přednáška

NAIL062 Výroková a predikátová logika

Jakub Bulín (KTIML MFF UK)

Zimní semestr 2023

Cesta k jistému úspěchu u zkoušky[1]

- Studujte průběžně (každý týden), a průběžně také testujte své znalosti: umíte sami napsat(!) definici, větu, důkaz?
- Před každou přednáškou alespoň zběžně projděte příslušné sekce v **Zápiscích z přednášky**. Ty obsahují vše, co po vás bude vyžadováno. Snažte se pochopit smysl definic a tvrzení.
- Po každé přednášce si skripta podrobně přečtěte. Pokud něčemu nebudete rozumět, využijte konzultačních hodin.
- Snažte se zúčastnit všech přednášek, pokud nemůžete, včas se materiál doučte, případně využijte konzultačních hodin.
- Ujistěte se, že rozumíte nejen myšlenkám, ale umíte pracovat i s formalizmem: ten je neoddělitelnou součástí logiky.
- Stejnou pozornost věnujte i přípravě na cvičení.

[1] Podrobnosti o zkoušce včas upřesníme. Základní formát i většina otázek ale pravděpodobně zůstanou stejné, viz **loňské informace o zkouškách**.

První přednáška

Program

- úvod do logiky
- neformální představení výrokové a predikátové logiky
- syntaxe výrokové logiky

Materiály

Zápisky z přednášky, Kapitola 1 a Sekce 2.1 z Kapitoly 2

Kapitola 1: Úvod do logiky

Co je logika?

Dvě definice:

1. soubor principů, které jsou základem uspořádání prvků nějakého systému (např. počítačového programu, elektronického zařízení, komunikačního protokolu)
2. věda o uvažování prováděném podle striktních pravidel zachovávajících platnost

V informatice obojí: daný systém nejprve *formálně popíšeme*, a poté o něm *formálně uvažujeme* (automaticky!), tj. odvozujeme platné inference za použití nějakého dokazovacího systému

Historie a aplikace logiky

Filozofie $\rightarrow$ Matematika $\rightarrow$ Teoretická informatika $\rightarrow$

Aplikovaná informatika

- logic programming
- discrete optimization (SAT solving, scheduling, planning)
- database theory
- verification (software, hardware, protocol)
- automated reasoning and proving
- knowledge-based representation
- artificial intelligence

1.1 Výroková logika

Příklad ze života: Hledání pokladu

Při hledání pokladu jsme narazili na rozcestí dvou chodeb. Víme, že na konci každé chodby je buď poklad, nebo drak, ale ne obojí. Trpaslík nám řekl, že: *"Alespoň jedna z těch dvou chodeb vede k pokladu"*, a že *"První chodba vede k drakovi."* Je známo, že trpaslíci buď vždy mluví pravdu, nebo vždy lžou. Kterou cestou se máme vydat?

Výroky neformálně

Výrok je tvrzení, kterému lze přiřadit pravdivostní hodnotu:

pravdivý (*True*, 1), nebo lživý (*False*, 0)

Prvovýroky (atomické výroky, výrokové proměnné) zkombinované pomocí logických spojek a závorek do složených výroků:

"(Trpaslík lže,) *právě když* (druhá chodba vede k drakovi.)"

- $\neg$ "neplatí X", *negace*
- $\wedge$ "X a Y", *konjunkce*
- $\vee$ "X nebo Y", *disjunkce* (není exkluzivní)
- $\rightarrow$ "pokud X, potom Y", *implikace* (čistě logická)
- $\leftrightarrow$ "X, právě když Y", *ekvivalence*

Formalizace ve výrokové logice

Volba množiny prvovýroků: bity informace popisující daný systém

$p_1 =$ *"Poklad je v první chodbě."*
$p_2 =$ *"Poklad je ve druhé chodbě."*

Co nejmenší, např. hodnota $t =$ *"Trpaslík mluví pravdu."* je jednoznačně určená hodnotami $\mathbb{P} = \{p_1, p_2\}$.

- *Poklad nebo drak, ale ne obojí*: zakódované do volby $\mathbb{P}$ (přítomnost draka je absence pokladu)
- *"První chodba vede k drakovi."* $\Leftrightarrow \neg p_1$
- *"Alespoň jedna z chodeb vede k pokladu."* $\Leftrightarrow p_1 \vee p_2$
- *Trpaslík buď mluví pravdu, nebo lže*:

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

Teorie $T = \{\varphi\}$ v jazyce $\mathbb{P} = \{p_1, p_2\}$, φ je axiom T.

Modely a důsledky

Lze určit, kde je poklad? Je p_1 nebo p_2 důsledkem φ resp. T?

"Svět", ve kterém je např. v první chodbě poklad a ve druhé drak, popíšeme pomocí pravdivostního ohodnocení $p_1 = 1, p_2 = 0$, neboli modelu $v = (1, 0)$ jazyka $\mathbb{P}$. Celkem máme 4 "světy" a modely:

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

Je "svět" popsaný modelem $v = (1, 0)$ *konzistentní* s tím, co víme, tj. platí v modelu v výrok φ resp. teorie T? Vyhodnotíme podle stromové struktury φ:

$$v(p_1) = 1,\ v(p_2) = 0,\ v(\neg p_1) = 0,\ v(p_1 \vee p_2) = 1,\ \ldots,\ v(\varphi) = 0$$

Množina modelů výroku φ (resp. *modelů teorie* T):

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

V každém modelu teorie T platí výrok p_2, neboli p_2 je důsledek T.

Důkazové systémy

Ověřovat všechny modely je nepraktické, pro $|\mathbb{P}| = n$ máme 2^n modelů, a $\mathbb{P}$ může být i nekonečná.

Důkazový systém

- důkaz výroku ψ z teorie T je formálně definovaný syntaktický objekt, snadno (mechanicky) ověřitelný
- lze hledat algoritmicky čistě na základě struktury ψ a axiomů T ("*syntaxe*"), nemusíme se zabývat modely ("*sémantikou*").

Klíčové vlastnosti:

- korektnost: pokud existuje důkaz ψ z T, potom ψ platí v T
- úplnost, pokud ψ platí v T, potom existuje důkaz ψ z T

Ukážeme si metodu analytického tabla a rezoluční metodu. Obě dokazují *sporem*: předpokládají platnost T a $\neg\psi$, hledají spor.

Metoda analytického tabla

- důkaz je strom olabelovaný předpoklady o platnosti výroků
- v kořeni: neplatí dokazovaný výrok ψ (důkaz sporem)
- připojíme platnost axiomů z T
- dále budujeme tablo zjednodušováním výroků ve vrcholech, podle pravidel zaručujících invariant:

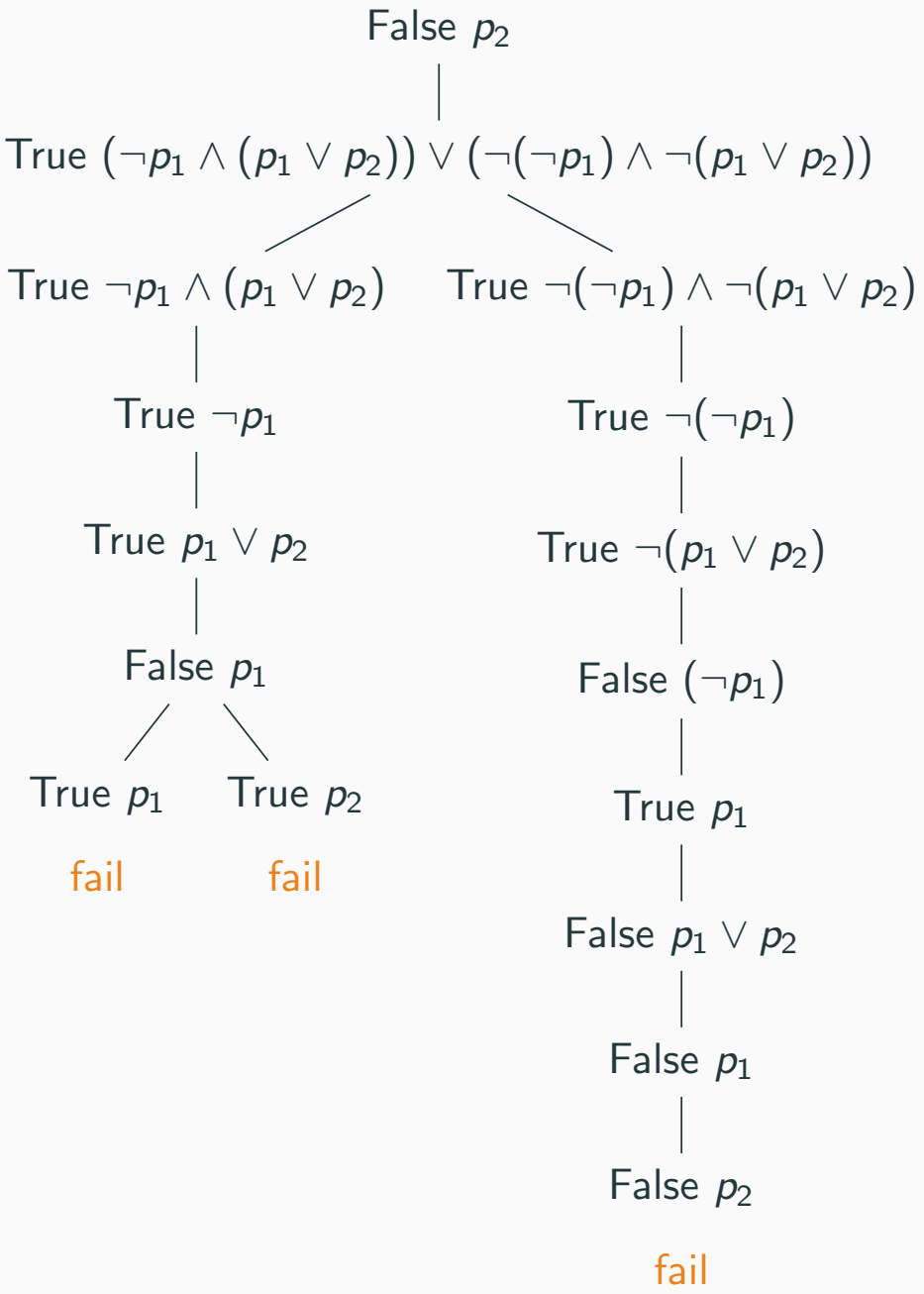
  Každý model teorie T, ve kterém neplatí ψ, se musí *shodovat* s některou z větví tabla.

- například:

  True $(\varphi_1 \to \varphi_2)$ zredukujeme rozvětvením na False φ_1 a True φ_2,
  False $(\varphi_1 \to \varphi_2)$ zredukujeme připojením True φ_1 a False φ_2.

- sporná větev = předpokládá True i False stejného výroku
- důkaz = všechny větve sporné (tj. nemůže existovat model T, ve kterém neplatí ψ)

Příklad tablo důkazu



Konjunktivní normální forma (CNF)

literál p, $\neg p$ klauzule disjunkce literálů CNF konjunkce klauzulí

každý výrok má ekvivalentní CNF ($\psi \sim \psi'$, stejné modely)

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

nahradíme $\neg(\neg p_1) \sim p_1$ a $\neg(p_1 \vee p_2) \sim (\neg p_1 \wedge \neg p_2)$ (*De Morgan*)

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

a dále opakovaně použijeme distributivitu $\vee$ vůči $\wedge$:

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

už je CNF, ještě zjednodušíme: odstraníme duplicitní literály, a klauzule obsahující p_i a zároveň $\neg p_i$ (to jsou tautologie)

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

Rezoluční důkaz

Důkaz sporem: převedeme negaci dokazovaného do CNF a přidáme

p_2 platí v T, právě když je následující CNF výrok nesplnitelný:

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

množinový zápis:

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

rezoluční pravidlo: je-li $p \in C_1$ a $\neg p \in C_2$, potom *rezolventa*

$$C = (C_1 \setminus \{p\}) \cup (C_2 \setminus \{\neg p\})$$

platí v každém modelu, ve kterém platí C_1 i C_2

rezoluční zamítnutí S: posloupnost klauzulí, každá je buď z S nebo rezolventa předchozích, poslední je prázdná klauzule $\square$

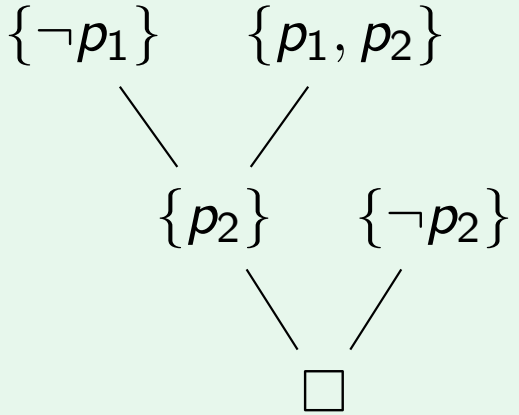
protože $\square$ nemá žádný model, je i S nesplnitelná

Příklad rezolučního důkazu

rezoluční zamítnutí (třetí klauzule je rezolventou 1. a 2., pátá je rezolventou 3. a 4.)

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

rezoluční strom (listy klauzule z S, vnitřní vrcholy rezolventy synů)



1.3 Další druhy logických systémů

Predikátové logiky vyšších řádů

- Predikátová logika, kde proměnné reprezentují jednotlivé vrcholy, je logika prvního řádu (first-order, FO)
- Logika druhého řádu (second-order, SO): proměnné i pro množiny vrcholů a n-tic vrcholů (tj. relace, funkce)

  "Každá neprázdná zdola omezená podmnožina má infimum."

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

- A v logice *třetího řádu* máme i množiny množin (např. v topologii).

Logiky zobecňující pojem pravdy

Kromě toho lze zobecnit pojem platnosti (pravdy):

- temporální logiky (platnost ‘vždy’, ‘někdy v budoucnosti’, ‘dokud’ apod.) – např. v paralelním programování
- modální logiky (‘je možné’, ‘je nutné’) – v umělé inteligenci, uvažování autonomních agentů o svém okolí
- fuzzy logiky (‘je 0.35 pravdivé’) – v automatických pračkách
- intuicionistická logika (povoluje jen konstruktivní důkazy, nemá *zákon vyloučeného třetího*)

1.4 O přednášce

Obsah předmětu

I. Výroková logika (5 přednášek)
   - Syntaxe a sémantika
   - Problém SAT
   - Tablo metoda
   - Rezoluční metoda

II. Predikátová logika (6 přednášek)
   - Syntaxe a sémantika
   - Tablo metoda v predikátové logice
   - Rezoluční metoda v predikátové logice
   - Aplikace: databáze, Prolog, verifikace

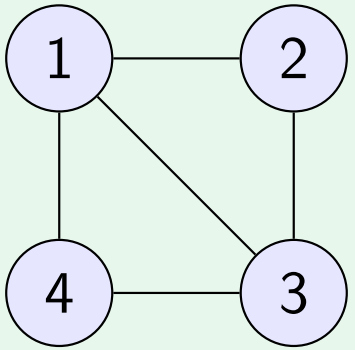
III. Pokročilé partie (2 přednášky)
   - Teorie modelů
   - Nerozhodnutelnost a neúplnost

ČÁST I – VÝROKOVÁ LOGIKA

Kapitola 2: Syntaxe a sémantika výrokové logiky

Příklad: Barvení grafů

Najděte vrcholové obarvení následujícího grafu třemi barvami.



graf: množina vrcholů a množina (libovolně) orientovaných hran

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

jak formalizovat? pro $v \in V$ a $c \in C = \{R, G, B\}$:

$$p_v^c = \text{“vrchol } v \text{ má barvu } c\text{”}$$

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

máme celkem $|\,\mathrm{M}_{\mathbb{P}}\,| = 2^{12} = 4096$ modelů jazyka (12-dim. vektorů)

Formalizace hranového obarvení

- každý vrchol má nejvýše jednu barvu: $4^4 = 2^8 = 256$ modelů

$$T_1 = \{(\neg p_v^R \vee \neg p_v^G) \wedge (\neg p_v^R \vee \neg p_v^B) \wedge (\neg p_v^G \vee \neg p_v^B) \mid v \in V\}$$

- a každý vrchol má alespoň jednu barvu: $3^4 = 81$ modelů

$$T_2 = T_1 \cup \{p_v^R \vee p_v^G \vee p_v^B \mid v \in V\} = T_1 \cup \{\bigvee_{c \in C} p_v^c \mid v \in V\}$$

T_2 je extenze teorie T_1 neboť každý důsledek T_1 platí i v T_2, zde dokonce $\mathrm{M}_{\mathbb{P}}(T_2) \subseteq \mathrm{M}_{\mathbb{P}}(T_1)$

- nakonec přidáme hranovou podmínku:

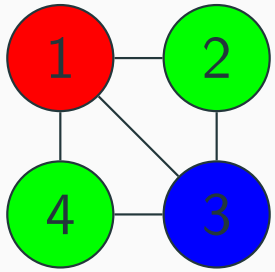
$$T_3 = T_2 \cup \{\bigwedge_{c \in C} (\neg p_u^c \vee \neg p_v^c) \mid (u, v) \in E\}$$

Výsledná teorie T_3 je splnitelná (má model), právě když je graf $\mathcal{G}$ 3-obarvitelný.

Co s ní?

Všechna obarvení?

T_3 má 6 modelů: $v = (1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0)$ a další získané permutací barev



Obarvení, ve kterých je vrchol 1 modrý a vrchol 2 zelený?

Odpovídají modelům teorie $T_3 \cup \{p_1^B, p_2^G\}$

Důkaz, že vrcholy 2 a 4 musí mít stejnou barvu?

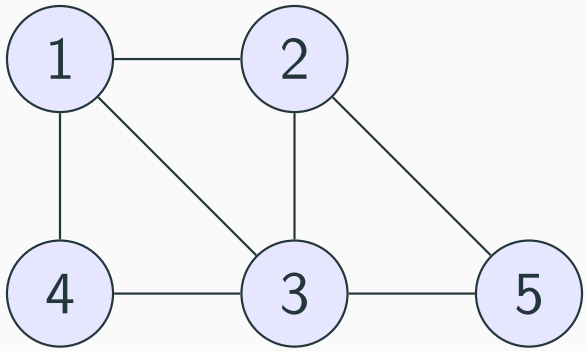
Tablo s kořenem False $(p_2^R \wedge p_4^R) \vee (p_2^G \wedge p_4^G) \vee (p_2^B \wedge p_4^B)$

Nebo rezolucí: přidáme negaci $(p_2^R \wedge p_4^R) \vee (p_2^G \wedge p_4^G) \vee (p_2^B \wedge p_4^B)$, vše převedeme do CNF a zamítneme

1.2 Predikátová logika

Nevýhody formalizace ve výrokové logice

Teorie T_3 je poměrně velká, a 'natvrdo' kóduje graf $\mathcal{G}$.



Obohatit jazyk $\mathbb{P}' = \mathbb{P} \cup \{p_5^R, p_5^G, p_5^B\}$ a vytvořit ještě větší teorii T_3' přidáním axiomů o vrcholu 5 a hranách $(2,5), (3,5)$?

A co vlastnosti obecně platné o všech nebo mnoha grafech?

V predikátové logice můžeme mluvit o vrcholech pomocí proměnných a přirozeně vyjádřit vlastnosti jako:

- *"z vrcholu u vede hrana do vrcholu v"*
- *"vrchol u je zelený"*

Predikátová logika: struktury a jazyk

Modely už nejsou 0–1 vektory, ale struktury, např. naše (orientované) grafy:

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

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

- množina vrcholů, a binární relace na této množině
- jazyk specifikuje kolik relací jakých arit má struktura mít, a symboly pro ně
- např. jazyk grafů $\mathcal{L} = \langle E \rangle$ (kde E je binární relační symbol)
- $\mathcal{G}$ a $\mathcal{G}'$ jsou struktury v jazyce $\mathcal{L}$ ($\mathcal{L}$-struktury)
- můžeme mít také funkce a konstanty, a symbol $=$ pro rovnost

Predikátová logika: syntaxe a sémantika

Syntaxe: místo prvovýroků atomické formule, např. $E(x, y)$, kde x, y jsou proměnné reprezentující vrcholy; stejné logické spojky, ale navíc kvantifikátory:

$(\forall x)$ *"pro všechny vrcholy x"*
$(\exists y)$ *"existuje vrchol y"*

(hrají roli "konjunkce" a "disjunkce" přes všechny prvky)

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

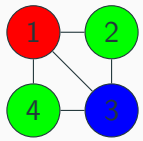
Sémantika: V daném grafu $\mathcal{G}$ a při dosazení vrcholu u za proměnnou x a vrcholu v za proměnnou y vyhodnotíme $E(x, y)$ jako True, právě když $(u, v) \in E^{\mathcal{G}}$.

Barvení grafů v predikátové logice

Jazyk $\mathcal{L}' = \langle E, R, G, B \rangle$, kde E je binární a R, G, B jsou unární relační symboly ($R(x)$ znamená *"vrchol x je červený"*)

$\mathcal{L}'$-struktura: graf s trojicí množin vrcholů

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



$\mathcal{G}_C$ je expanze $\mathcal{L}$-struktury $\mathcal{G}$ do jazyka $\mathcal{L}'$.

Nejvýše jedna barva, alespoň jedna barva, hranová podmínka:

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