

http://math.feld.cvut.cz/demlova/teaching/lgr/text_lgr_2017.pdf
https://math.fel.cvut.cz/en/people/demlova/lgr/predn_lgr.html

Výroková logika

Syntax výrokové logiky. Sémantika výrokové logiky. Důkazový systém - přirozená dedukce.

Významová (sémantická) ekvivalence formulí výrokové logiky. Normální formy formulí. Důsledek ve výrokové logice. Úplné systémy logických spojek. Schopnost formalisace a řešení logických úloh s využitím výrokové logiky.

Výroková logika

Výroky

Máme danou neprázdnou množinu A tzv. *atomických výroků* (též jim říkáme *logické proměnné*). Konečnou posloupnost prvků z množiny A , logických spojek a závorek nazýváme *výrokovou formule* (zkráceně jen *formule*), jestliže vznikla podle následujících pravidel:

1. Každá logická proměnná (atomický výrok) $a \in A$ je výroková formule.
2. Jsou-li α, β výrokové formule, pak $\neg\alpha, (\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \Rightarrow \beta), (\alpha \Leftrightarrow \beta)$ jsou také výrokové formule.
3. Nic jiného než to, co vzniklo pomocí konečně mnoha použití bodů 1 a 2, není výroková formule.

Všechny formule, které vznikly z logických proměnných množiny A , značíme $P(A)$.

Poznámka: Spojka \neg se nazývá *unární*, protože vytváří novou formuli z jedné formule. Ostatní zde zavedené spojky se nazývají *binární*, protože vytvářejí novou formuli ze dvou formulí.

V dalším textu logické proměnné označujeme malými písmeny např. a, b, c, \dots nebo x, y, z, \dots , výrokové formule označujeme malými řeckými písmeny např. $\alpha, \beta, \gamma, \dots$ nebo φ, ψ, \dots . Také většinou nebudeme ve formulích psát ty nejvíc vnější závorky - tj. píšeme $a \vee (b \Rightarrow c)$ místo $(a \vee (b \Rightarrow c))$.

Podformule

Ze syntaktického stromu formule α jednoduše poznáme všechny její podformule: *Podformule* formule α jsou všechny formule odpovídající podstromům syntaktického stromu formule α .

Pravdivostní ohodnocení

Pravdivostní ohodnocení, též pouze *ohodnocení formulí*, je zobrazení $u : P(A) \rightarrow \{0, 1\}$, které splňuje pravidla

1. **Negace** ($\neg\alpha$) je pravdivá právě tehdy, když α je nepravdivá, tj. $u(\neg\alpha) = 1$ právě tehdy, když $u(\alpha) = 0$;
 2. **Konjunkce** ($\alpha \wedge \beta$) je pravdivá právě tehdy, když α a β jsou obě pravdivé, tj. $u(\alpha \wedge \beta) = 1$ právě tehdy, když $u(\alpha) = u(\beta) = 1$;
 3. **Disjunkce** ($\alpha \vee \beta$) je nepravdivá právě tehdy, když α a β jsou obě nepravdivé, tj. $u(\alpha \vee \beta) = 0$ právě tehdy, když $u(\alpha) = u(\beta) = 0$;
 4. **Implikace** ($\alpha \Rightarrow \beta$) je nepravdivá právě tehdy, když α je pravdivá a β nepravdivá, tj. $u(\alpha \Rightarrow \beta) = 0$ právě tehdy, když $u(\alpha) = 1$ a $u(\beta) = 0$;
 5. **Ekvivalence** ($\alpha \Leftrightarrow \beta$) je pravdivá právě tehdy, když buď obě formule α a β jsou pravdivé nebo obě jsou nepravdivé, tj. $u(\alpha \Leftrightarrow \beta) = 1$ právě tehdy, když $u(\alpha) = u(\beta)$.
- Dvě pravdivostní ohodnocení jsou shodná, pokud mají pro všechny logické proměnné shodný výsledek

Pravdivostní tabulky

Vlastnosti, které ohodnocení formulí musí mít, znázorňujeme též pomocí tzv. pravdivostních tabulek logických spojek. Jsou to:

| α | $\neg \alpha$ | | | | |
|----------|---------------|-----------------------|---------------------|----------------------------|--------------------------------|
| 0 | 1 | | | | |
| 1 | 0 | | | | |
| α | β | $\alpha \wedge \beta$ | $\alpha \vee \beta$ | $\alpha \Rightarrow \beta$ | $\alpha \Leftrightarrow \beta$ |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |

| α | β | $\alpha \wedge \beta$ | $\alpha \vee \beta$ | $\alpha \Rightarrow \beta$ | $\alpha \leftrightarrow \beta$ |
|----------|---------|-----------------------|---------------------|----------------------------|--------------------------------|
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |

Tautologie, kontradikce, splnitelné formule

- Formule se nazývá *tautologie*, jestliže je pravdivá ve všech ohodnoceních formulí.
- Formule se nazývá se *kontradikce*, jestliže je nepravdivá ve všech ohodnoceních formulí.
- Formule je *splnitelná*, jestliže existuje aspoň jedno ohodnocení formulí, ve kterém je pravdivá.

Příklady

- Formule $\alpha \vee \neg \alpha$, $\alpha \Rightarrow \alpha$, $\alpha \Rightarrow (\beta \Rightarrow \alpha)$ jsou tautologie.
- Formule $a \vee b$, $(a \Rightarrow b) \Rightarrow a$ jsou splnitelné, ale ne tautologie.
- Formule $\alpha \wedge \neg \alpha$ je kontradikce. Kontradikce je také každá negace tautologie.

Úplné systémy logických spojek

Množiny logických spojek, pomocí, kterých lze popsat všechny možné formule.

například: $\{\vee, \neg\}$, $\{\wedge, \neg\}$, $\{\Rightarrow, \neg\}$

Tautologická ekvivalence formulí

Řekneme, že formule φ a ψ jsou *tautologicky ekvivalentní* (také *sémanticky ekvivalentní*), jestliže pro každé ohodnocení u platí $u(\varphi) = u(\psi)$.

Tautologickou ekvivalenci značíme $\varphi \models \psi$.

- Tautologická ekvivalence je reflexivní, symetrická a tranzitivní (takže ekvivalence)
- Jsou-li α , β , γ , δ formule splňující $\alpha \models \beta$ a $\gamma \models \delta$, pak platí
 - $\neg \alpha \models \neg \beta$;
 - $(\alpha \wedge \gamma) \models (\beta \wedge \delta)$
 - $(\alpha \vee \gamma) \models (\beta \vee \delta)$
 - $(\alpha \Rightarrow \gamma) \models (\beta \Rightarrow \delta)$
 - $(\alpha \leftrightarrow \gamma) \models (\beta \leftrightarrow \delta)$
- Zákony pro formule
 - $\alpha \wedge \alpha \models \alpha$, $\alpha \vee \alpha \models \alpha$ (idempotence \wedge a \vee);
 - $\alpha \wedge \beta \models \beta \wedge \alpha$, $\alpha \vee \beta \models \beta \vee \alpha$ (komutativita \wedge a \vee);
 - $\alpha \wedge (\beta \wedge \gamma) \models (\alpha \wedge \beta) \wedge \gamma$, $\alpha \vee (\beta \vee \gamma) \models (\alpha \vee \beta) \vee \gamma$ (asociativita \wedge a \vee);
 - $\alpha \wedge (\beta \vee \alpha) \models \alpha$, $\alpha \vee (\beta \wedge \alpha) \models \alpha$ (absorpce \wedge a \vee);
 - $\neg \neg \alpha \models \alpha$;
 - $(\alpha \Rightarrow \beta) \models (\neg \alpha \vee \beta)$
- Dvě formule jsou sémanticky ekvivalentní právě když je jejich ekvivalence tautologií

| x | f_1 | f_2 | f_3 | f_4 |
|-----|-------|-------|-------|-------|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |

Funkce f_1 je konstantní 0, jedná se o *kontradikci* a budeme ji značit **F**. Podobně funkce f_4 je *tautologie* (konstantní 1), značíme je **T**. Funkce f_2 je vlastně logická proměnná x a funkce f_3 je $\neg x$.

Normální formy formulí

- Disjunktivní normální forma (DNF)**
 - Formule je v DNF pokud je disjunkcí jedné nebo několika formulí z nichž každá je literálem nebo konjunkcí literálů
- Konjunktivní normální forma (CNF)**
 - Formule je v CNF pokud je konjunkcí jedné nebo několika formulí, z nichž každá je literálem nebo disjunkcí literálů
- Ke každé formuli najdeme ekvivalentní formuli v DNF i v CNF

Důsledek ve výrokové logice

- φ je *sémantickým důsledkem*, též *konsekventem* množiny formulí S právě tehdy, když je pravdivá ve všech ohodnoceních, kdy je pravdivá formule S . Tento fakt značíme $S \models \varphi$

- Dvě formule jsou sémanticky ekvivalentní pokud si jsou vzájemně sémantickým důsledkem
- $\alpha \models \beta$ právě tehdy, když $\alpha \Rightarrow \beta$ je tautologie
- $S \models \varphi$ právě tehdy, když $S \cup \{\neg\varphi\}$ je nespílitelná
- $S \cup \{\varphi\} \models \psi$ právě tehdy, když $S \models (\varphi \Rightarrow \psi)$

Příklady

- Pro každé tři formule α, β, γ platí:
 1. $\{\alpha, \alpha \Rightarrow \beta\} \models \beta$.
 2. $\{\alpha \Rightarrow \beta, \beta \Rightarrow \gamma\} \models (\alpha \Rightarrow \gamma)$.
 3. $\{\alpha \Rightarrow \beta, \neg\beta\} \models \neg\alpha$.
 4. $\{\alpha \vee \beta, \alpha \Rightarrow \gamma, \beta \Rightarrow \gamma\} \models \gamma$.
- Pro konsekventy platí následující tvrzení:
 1. Je-li S množina formulí a $\varphi \in S$, pak φ je konsekvemem S , tj. $S \models \varphi$ pro každou $\varphi \in S$
 2. Tautologie je konsekvemem každé množiny formulí S
 3. Formule φ je tautologie právě tehdy, když je konsekvemem každé množiny formulí
 4. Každá formule je konsekvemem nespílitelné množiny formulí

Rezoluční metoda

- Rezoluční metoda rozhoduje, zda daná množina klausulí je splnitelná nebo je nespílitelná. Tím je také "universální metodou" pro řešení základních problémů ve výrokové logice, neboť:
- Je "univerzální metodou" pro řešení základních problémů ve výrokové logice neboť lze použít k ověření sémantického důsledku i pravdivosti složitých výrokových formulí
- Základní pojmy rezoluční metody:
 - *Klausule* = literál nebo disjunkce konečně mnoha literálů, speciálním případem je prázdná klausule značená F (kontradikce)
 - Pokud C je klausule a p je literál, pak $C \setminus p$ je klausule, která obsahuje všechny literály z C kromě p
 - Klausule D je rezolventa klauzulí C_1 a C_2 právě když existuje literál p , který se vyskytuje v klauzuli C_1 a jehož negace se vyskytuje v klauzuli C_2 , a zároveň platí $D = (C_1 \setminus p) \vee (C_2 \setminus \neg p)$. Značíme $D = res_p(C_1, C_2)$
- Pro resolventu D z klauzulí C_1 a C_2 platí, že je sémantickým důsledkem množiny $\{C_1, C_2\}$

Rezoluční princip

- Značení:
 - A je množina literálů
 - $R(S) = S \cup \{res_p(C_1, C_2) \mid C_1, C_2 \in S, p \in A\}$
 - $R_0(S) = S$
 - $R_{i+1}(S) = R(R_i(S))$ pro $i \in \mathbb{N}$
 - $R^*(S) = \bigcup \{R_i(S) \mid i \geq 0\}$
- Existuje vždy n takové, že $R_n(S) = R_{n+1}(S)$, takže $R_n(S) = R^*(S)$
- Množina klauzulí S je splnitelná právě tehdy, když $R^*(S)$ neobsahuje prázdnou klauzuli F

Postup rezoluční metody

1. Formule množiny M převedeme do CNF a M pak nahradíme množinou S všech klauzulí vyskytujících se v některé formuli v CNF. Klausule, které jsou tautologiemi, vynecháme. Jestliže nám nezbyde žádná klausule, množina M se skládala z tautologií a je pravdivá v každém pravdivostním ohodnocení.
 2. Vytvoříme $R^*(S)$
 3. Obsahuje-li $R^*(S)$ prázdnou klauzuli, je množina M nespílitelná, v opačném případě je M splnitelná.
- Lze použít i výhodnější postup, ve kterém pouze kontrolujeme zda $R^*(S)$ obsahuje F

Výhodnější postup

- Nejprve si zvolíme proměnnou x a rozdělíme klauzule v S do tří skupin, klauzule M_0 které neobsahují x , klauzule M_x které obsahují x a klauzule $M_{\neg x}$ které obsahují negaci x
- Vytvoříme množinu N všech rezolvent z M_x a $M_{\neg x}$, kde vyřadíme všechny tautologie
- Vytvoříme množinu $S_1 = M_0 \cup N$
- Položíme $S = S_1$ a opakujeme pro další proměnné
- Na konci získáme buď prázdnou množinu (S je splnitelná) nebo množinu $\{F\}$, (S není splnitelná)

Přirozená dedukce ve výrokové logice

- Je formálním důkazovým systémem
- Zabývá se odvoditelností, nikoliv pravdivostí
- Nepoužívá žádné axiomy, ale pouze pomocné předpoklady
- Definuje dva druhy pravidel, I-pravidla (zavádějí logické spojky) a E-pravidla (odstraňují logické spojky)
- Při přirozené dedukci se odvozuje odvození, což je posloupnost $\varphi_1, \dots, \varphi_n$, kde každá formule je buď předpoklad, nebo pomocný předpoklad, nebo vznikla z předchozích formulí pomocí některého odvozovacího pravidla

Logický důsledek

- φ je logický důsledek množiny S , pokud $\varphi_n = \varphi$
- Značíme $S \vdash \varphi$
- Platí, že $S \vdash \varphi \Leftrightarrow S \models \varphi$

Odvozovací I-pravidla

- Pro konjunkci:

$$\frac{\varphi, \psi}{\varphi \wedge \psi}, \quad \frac{\varphi, \psi}{\psi \wedge \varphi}$$

- Pro disjunkci:

$$\frac{\varphi}{\varphi \vee \psi}, \quad \frac{\varphi}{\psi \vee \varphi}$$

- Pro implikaci:

$$\frac{\begin{array}{|c|} \hline \varphi \\ \vdots \\ \psi \\ \hline \end{array}}{\varphi \Rightarrow \psi}$$

- Pro ekvivalenci:

$$\frac{\begin{array}{|c|} \hline \varphi \\ \vdots \\ \psi \\ \hline \end{array}, \begin{array}{|c|} \hline \psi \\ \vdots \\ \varphi \\ \hline \end{array}}{\varphi \Leftrightarrow \psi}$$

- Pro negaci (důkaz sporem):

$$\frac{\begin{array}{|c|} \hline \varphi \\ \vdots \\ \perp \\ \hline \end{array}}{\neg \varphi}$$

- Pro spor nelze zavést
- Pro dvojitou negaci:

$$\frac{\varphi}{\neg \neg \varphi}$$

Odvozovací E-pravidla

- Pro konjunkci:

$$\frac{\varphi \wedge \psi}{\varphi}, \quad \frac{\varphi \wedge \psi}{\psi}$$

- Pro disjunkci:

$$\frac{\varphi \vee \psi, \varphi \Rightarrow \alpha, \psi \Rightarrow \alpha}{\alpha}$$

- Pro implikaci (Modus Ponens):

$$\frac{\varphi, \varphi \Rightarrow \psi}{\psi}$$

- Pro ekvivalenci:

$$\frac{\varphi, \varphi \Leftrightarrow \psi}{\psi}, \quad \frac{\psi, \varphi \Leftrightarrow \psi}{\varphi}$$

- Pro negaci:

$$\frac{\varphi \wedge \neg \varphi}{\perp}$$

- Pro spor:

$$\frac{\perp}{\varphi}$$

- Pro dvojitou negaci:

$$\frac{\neg \neg \varphi}{\varphi}$$

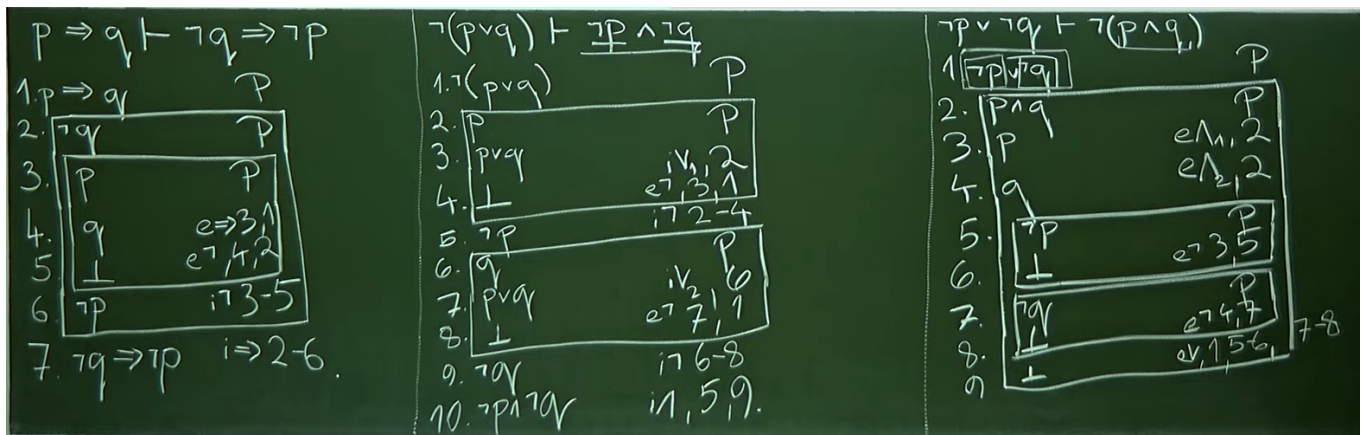
Speciální pravidla

- Obrácená implikace:

$$\frac{\neg \psi, \varphi \Rightarrow \psi}{\neg \varphi}$$

- Vyloučení třetího:

$$\frac{}{\varphi \vee \neg \varphi}$$



Predikátová logika

Syntax predikátové logiky. Sémantika predikátové logiky. Významová (sémantická) ekvivalence formulí predikátové logiky. Normální formy formulí. Důsledek v predikátové logice. Schopnost formalisace a řešení logických úloh s využitím predikátové logiky.

Syntaxe predikátové logiky

- Jazyk predikátové logiky \mathcal{L}
- Jazyk predikátové logiky se skládá z
 1. *logických symbolů*, tj.:
 1. spočetné množiny individuálních proměnných: $\text{Var} = \{x, y, \dots, x_1, x_2, \dots\}$
 2. výrokových logických spojek: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
 3. obecného kvantifikátoru \forall a existenčního kvantifikátoru \exists
 2. *speciálních symbolů*, tj.:
 1. množiny Pred predikátových symbolů (nesmí být prázdná) - velká písmena
 2. množiny Kons konstantních symbolů (může být prázdná) - malá písmena na začátku abecedy
 3. množiny Func funkčních symbolů (může být prázdná) - například f, g atp.
 3. pomocných symbolů, jako jsou závorky a čárka

Arita

- Popisuje kolika objektů se daný predikát nebo funkce týká

Pojmy

- **Term** - Množina termů je definována těmito pravidly:
 1. Každá proměnná a každý konstantní symbol je term
 2. Jestliže f je funkční symbol arity n a t_1, t_2, \dots, t_n jsou termy, pak $f(t_1, t_2, \dots, t_n)$ je také term
- **Atomická formule** - predikátový symbol P aplikovaný na tolik termů, kolik je jeho arita
- **Formule** - množina formulí je definována těmito pravidly:
 1. Každá atomická formule je formule.
 2. Jsou-li φ a ψ dvě formule, pak $(\neg\varphi)$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \Rightarrow \psi)$, $(\varphi \Leftrightarrow \psi)$ jsou opět formule.
 3. Je-li φ formule a x proměnná, pak $(\forall x\varphi)$ a $(\exists x\varphi)$ jsou opět formule.
- **Podformule** - libovolný podřetězec φ , který je sám formulí

Volný a vázaný výskyt proměnné

- Máme proměnnou x , která je v listu syntaktického stromu formule φ
 - Jde o **vázaný výskyt** proměnné x pokud jsme při postupu od listu ke kořeni narazili na kvantifikátor s touto proměnnou
 - V opačném případě jde o **volný výskyt** proměnné x
- Formule, která má pouze vázané výskyty proměnných se nazývá **uzavřená formule** (též **sentence**)
- Formule, která má pouze volné výskyty se nazývá **otevřená formule**
- Vázané výskyty nějaké proměnné můžeme přejmenovat, pokud jsou přejmenovány všechny výskyty vázané daným kvantifikátorem a při přejmenování se nestal žádný volný výskyt vázaným, tomu říkáme **legální přejmenování proměnné**
- Dvě formule jsou **stejné** pokud se liší pouze legálním přejmenováním

Sémantika predikátové logiky

Interpretace jazyka predikátové logiky

- Dvojice $\langle U, \llbracket - \rrbracket \rangle$, kde
 - U je neprázdná množina nazývaná *universum*;
 - $\llbracket - \rrbracket$ je přiřazení, které
 1. Každému predikátovému symbolu $P \in \text{Pred}$ arity $n > 0$ přiřazuje podmnožinu U^n , tj. n -ární relaci na množině U . Když $n = 0$, tak přiřazuje P nulu nebo jedničku.
 2. Každému konstantnímu symbolu $a \in \text{Kons}$ přiřazuje prvek z U , značíme jej $\llbracket a \rrbracket$,
 3. každému funkčnímu symbolu $f \in \text{Func}$ arity n přiřazuje zobrazení množiny U^n do U , značíme je $\llbracket f \rrbracket$,
- Kontext proměnných ρ je zobrazení, které každé proměnné $x \in \text{Var}$ přiřadí prvek $\rho(x) \in U$
 - Update kontextu ρ o hodnotu d v x $\rho[x := d]$ je nový kontext proměnných, který se od ρ liší jenom v proměnné x , která v něm má hodnotu d
- Termy jsou interpretovány následujícím způsobem:
 1. Je-li term konstantní symbol $a \in \text{Kons}$, pak jeho hodnota je prvek $\llbracket a \rrbracket_\rho = \llbracket a \rrbracket$. Je-li term proměnná x , pak jeho hodnota je $\llbracket x \rrbracket_\rho = \rho(x)$
 2. Je-li $f(t_1, t_2, \dots, t_n)$ term, pak jeho hodnota je

$$\llbracket f(t_1, t_2, \dots, t_n) \rrbracket_\rho = \llbracket f \rrbracket(\llbracket t_1 \rrbracket_\rho, \dots, \llbracket t_n \rrbracket_\rho)$$

Pravdivostní hodnoty

- Formule v dané interpretaci a daném kontextu
 1. Nechť φ je atomická formule. Tj. $\varphi = P(t_1, t_2, \dots, t_n)$, kde P je predikátový symbol arity n a t_1, t_2, \dots, t_n jsou termy. Pak φ je pravdivá v interpretaci $\langle U, \llbracket - \rrbracket \rangle$ a kontextu ρ právě tehdy, když
$$(\llbracket t_1 \rrbracket_\rho, \dots, \llbracket t_n \rrbracket_\rho) \in \llbracket P \rrbracket$$
 2. Jsou-li φ a ψ formule, jejichž pravdivost v interpretaci $\langle U, \llbracket - \rrbracket \rangle$ a kontextu ρ již známe, pak se řídíme tím co již známe z výrokové logiky
 3. Je-li φ formule a x proměnná, pak
 - $\forall x\varphi(x)$ je pravdivá právě tehdy, když formule φ je pravdivá v každém kontextu $\rho[x := d]$, kde d je prvek U
 - $\exists x\varphi(x)$ je pravdivá právě tehdy, když formule φ je pravdivá v aspoň jednom kontextu $\rho[x := d]$, kde d je prvek U
- Sentence φ je pravdivá v interpretaci $\langle U, \llbracket - \rrbracket \rangle$ právě když je pravdivá v každém kontextu ρ

Pojmy

- *Model sentence* je interpretace ve které je sentence φ pravdivá
- *Tautologie* je sentence, která je pravdivá v každé interpretaci (například $\forall x P(x) \Rightarrow P(a)$)
- *Kontradikce* je sentence, která je nepravdivá v každé interpretaci (například $\forall x P(x) \wedge \exists x \neg P(x)$)
- Sentence je *splnitelná*, pokud je pravdivá v alespoň jedné interpretaci
- *Splnitelná množina sentencí* je taková množina sentencí M , pro kterou existuje interpretace v níž jsou všechny sentence z M pravdivé (i prázdná množina)
- *Nesplnitelná množina sentencí* je taková, kde v každé interpretaci je nepravdivá některá sentence z množiny M
- Sentence φ a ψ jsou *tautologicky (sémanticky) ekvivalentní*, když jsou pravdivé ve stejných interpretacích

Tvrzení o tautologické ekvivalenci

1. $\neg(\forall x P(x)) \models (\exists x \neg P(x))$
2. $\neg(\exists x P(x)) \models (\forall x \neg P(x))$
3. $(\forall x P(x)) \wedge (\forall x Q(x)) \models \forall x (P(x) \wedge Q(x))$

Sémantický důsledek

- Sentence φ je sémantickým důsledkem (nebo *konsekventem*) množiny sentencí S právě tehdy, když každý model množiny S je také modelem sentence φ
- Jestliže existuje interpretace $\langle U, [-] \rangle$ v níž je pravdivá každá sentence z množiny S , ale není pravdivá formule φ , pak φ není sémantickým důsledkem S

Rezoluční metoda v predikátové logice

Základní pojmy

- *Literál* je atomická formule, nebo negace atomické formule
- *Komplementární literály* jsou dva literály z nichž jeden je negací druhého
- *Klausule* je sentence, která obsahuje pouze obecné kvantifikátory, které leží na jejím začátku. Za kvantifikátory následuje literál nebo disjunkce literálů
- *Rezolventy* - analogické k rezolventům ve výrokové logice, ale postup k jejich nalezení je složitější

Vytváření rezolvent v predikátové logice

- Rezolventy v predikátové logice nemusejí vždy existovat
- Příklad - vytvoříme z klauzulí K_1 a K_2 rezolventu K

$$K_1 = \forall x \forall y (P(x) \vee \neg Q(x, y))$$

$$K_2 = \forall x \forall y (Q(x, y) \vee R(y))$$

$$K = \forall x \forall y (P(x) \vee R(y))$$

- Unifikační algoritmus
 - **Vstup:** Dva pozitivní literály L_1, L_2 , které nemají společné proměnné.
 - **Výstup:** Hlášení neexistuje v případě, že hledaná substituce neexistuje, v opačném případě substituce ve tvaru množiny prvků tvaru x/t , kde x je proměnná, za kterou se dosazuje, a t je term, který se za proměnnou x dosazuje
- 1. Položme $E_1 := L_1, E_2 := L_2, \theta := \emptyset$
- 2. Jsou-li E_1, E_2 prázdné řetězce, stop. Množina θ určuje hledanou substituci. V opačném případě položíme $E_1 := E_1\theta, E_2 := E_2\theta$
- 3. Označíme X první symbol řetězce E_1 , Y první symbol řetězce E_2
- 4. Je-li $X = Y$, odstraníme X a Y z počátku E_1 a E_2 . Jsou-li X a Y predikátové nebo funkční symboly, odstraníme i jim příslušné závorky a jdeme na krok 2
- 5. Je-li X proměnná, neděláme nic.
- 6. Je-li Y proměnná (a X nikoli), přehodíme E_1, E_2 a X, Y
- 7. Není-li ani X ani Y proměnná, stop. Výstup neexistuje
- 8. Je-li Y proměnná nebo konstanta, položíme $\theta := \theta \cup \{X/Y\}$. Odstraníme X a Y ze začátků řetězců E_1 a E_2 (spolu s čárkami, je-li třeba) a jdeme na krok 2
- 9. Je-li Y funkční symbol, označíme Z výraz skládající se z Y a všech jeho argumentů (včetně závorek a čárek). Jestliže Z obsahuje X , stop, výstup neexistuje
- 10. V opačném případě položíme $\theta := \theta \cup \{X/Z\}$, odstraníme X a Z ze začátků E_1 a E_2 (odstraníme čárky, je-li třeba) a jdeme na krok 2

Rezoluční princip

- Stejný jako ve výrokové logice

Skolemizace

- Odstranění existenčních kvantifikátorů s zachováním ekvivalence
- $\exists x P(x) \approx P(c)$, kde c je nový konstantní symbol
- $\forall x \exists y R(x, y) \approx \forall x R(x, f(x))$, kde f je nový funkční symbol arity 1.

Teorie grafů



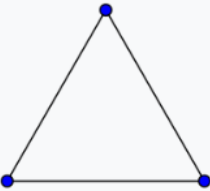
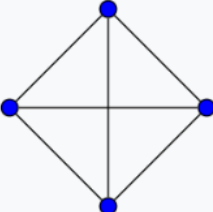
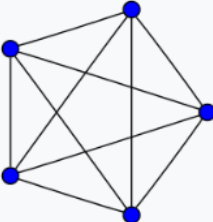
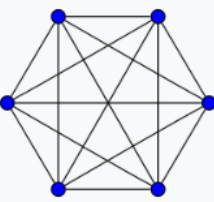
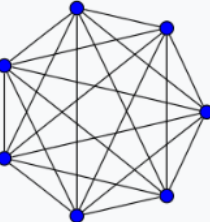
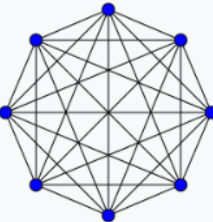
Základní pojmy a definice teorie grafů; schopnost formální práce s těmito pojmy. Stromy a jejich vlastnosti. Minimální kostry a algoritmy na jejich hledání. Komponenty silné souvislosti a algoritmus na jejich hledání. Schopnost modelování praktických problémů s využitím grafů.

Definice orientovaného grafu

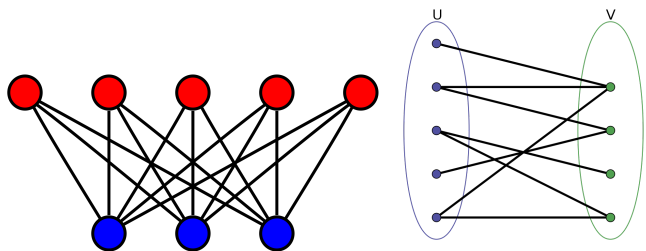
- Orientovaný graf je trojice $G = (V, E, \varepsilon)$, kde V je konečná množina vrcholů (též zvaných uzlů), E je konečná množina jmen hran (též nazývaných orientovaných hran) a ε je přiřazení, které každé hraně $e \in E$ přiřazuje uspořádanou dvojici vrcholů a nazývá se vztah incidence
- Jestliže $\varepsilon(e) = (u, v)$ pro $u, v \in V$, říkáme, že vrchol u je *počáteční vrchol* hrany e a vrchol v je *koncový vrchol* hrany e . O vrcholech u, v říkáme, že jsou krajní vrcholy hrany e , též že jsou incidentní s hranou e . Jestliže počáteční vrchol a koncový vrchol jsou stejné, říkáme, že hrana e je orientovaná smyčka.

Speciální případy grafů

- Graf o n vrcholech, ve kterém $E = \binom{V}{2}$ je množina všech dvouprvkových podmnožin množiny V se nazývá *úplný graf*, značí se K_n .

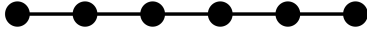
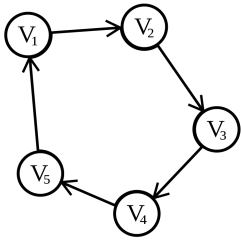
| $K_1: 0$ | $K_2: 1$ | $K_3: 3$ | $K_4: 6$ |
|---|---|---|--|
|  |  |  |  |
| $K_5: 10$ | $K_6: 15$ | $K_7: 21$ | $K_8: 28$ |
|  |  |  |  |

- Vlastnosti úplného grafu: $|E| = n \cdot \frac{n-1}{2}$, je regulární a celý graf je svojí největší klikou
- *Bipartitní graf* je graf, jehož množina vrcholů se dá rozdělit na dvě disjunktní podmnožiny V_1, V_2 tak, že každá hrana grafu má jeden koncový vrchol ve V_1 a druhý ve V_2



- Vlastnosti bipartitního grafu:
 - Neobsahuje kružnici liché délky.
 - Bipartitní graf, který obsahuje všechny možné hrany se nazývá *úplný bipartitní graf*. Značí se $K_{m,n}$, kde $|V_1| = m$, $|V_2| = n$ jsou velikosti partit. Na obrázku s červenými a modrými vrcholy je např. $K_{5,3}$.
 - $|E| = mn$.
 - Je regulární, právě když $m = n$.

- Cesta délky n je graf isomorfní grafu $V = \{0, 1, \dots, n\}$ a s množinou hran $E = \{\{0, 1\}, \{1, 2\}, \dots, \{n-1, n\}\}$. Značíme ji P_n .



- Kružnice délky $n \geq 3$ je graf isomorfní grafu s množinou vrcholů $V = \{1, \dots, n\}$ a s množinou hran $E = \{\{1, 2\}, \{2, 3\}, \dots, \{n-1, n\}\}$. Značíme ji C_n .

Izomorfní graf

- Izomorfní grafy jsou grafy u kterých jsou shodné všechny důležité vlastnosti (obsahují stejné podgrafy, cesty, kružnice,...)
- Formálně řekneme, že grafy G a G' jsou izomorfní, pokud

$$\exists F : V(G) \rightarrow V(G') : \{x, y\} \in E(G) \Leftrightarrow \{f(x), f(y)\} \in E(G')$$

| Regulární graf | Všechny vrcholy stejný stupeň |
|----------------------------------|--|
| Podgraf | $G' = (V', E')$, kde $V' \subseteq V$, $E' \subseteq E$ |
| Faktor grafu | Podgraf $G' = (V', E')$, kde $V' = V$ |
| Podgraf indukovaný množinou V' | Podgraf $G' = (V', E')$, kde E' obsahuje všechny hrany, které mají oba krajní vrcholy v množině V' |
| Sled | Sled délky k je posloupnost vrcholů a hran $v_0, e_1, v_1, e_2, \dots, v_{k-1}, e_k, v_k$ taková, že hrana e_i je incidentní s vrcholy v_{i-1} a v_i . |
| Uzavřený sled | Sled, kde $v_0 = v_k$ |
| Tah (uzavřený) | Sled (uzavřený), ve kterém se neopakují hrany. |
| Eulerovský tah (uzavřený) | Tah (uzavřený), který obsahuje všechny hrany a všechny vrcholy grafu. |
| Cesta (uzavřená) | Tah (uzavřený), ve kterém se neopakují vrcholy (s výjimkou $v_0 = v_k$). |
| Cyklus | Uzavřená orientovaná cesta, ve které se neopakují vrcholy. |
| Kružnice | Uzavřená cesta, která má aspoň jednu hranu |
| Souvislý graf | Mezi každými jeho vrcholy existuje cesta. |
| Strom | Souvislý graf, který neobsahuje kružnici. |
| Kořenový strom | Orientovaný graf, který je strom a má kořen. |
| Les | Graf, který neobsahuje kružnice. |
| Prostý graf | Graf bez paralelních hran. |
| Obyčejný graf | Prostý graf bez smyček. |
| Eulerovský graf | Graf, kde existuje eulerovský tah. |
| Silně souvislý graf | Orientovaný graf, kde mezi každými dvěma vrcholy existuje orientovaná cesta. |
| Doplňek grafu | $G^{\text{dop}} = (V, \binom{V}{2} - E)$, tedy s hranami, které nejsou v původním grafu. |
| Vzdálenost vrcholů | Délka nejkratší cesty mezi dvěma vrcholy. |
| Komponenta souvislosti | Maximální souvislý podgraf. |
| Skóre grafu | Sestupně seřazená n -tice stupňů vrcholů. |
| Kostra grafu | Faktor, který je stromem. |
| Komponenta silné souvislosti | Maximální podgraf, který je souvislý. |
| Kořen | Vrchol u orientovaného grafu, z něhož vede orientovaná cesta do všech ostatních vrcholů. |
| Jádro | Jádro orientovaného grafu $G = (V, E)$ je množina $J \subseteq V$ jeho vrcholů taková, že mezi libovolnými dvěma nevede hrana a z každého vrcholu mimo J vede aspoň jedna hrana do J . |
| Obarvení vrcholů | Zobrazení $b : V \rightarrow B$ takové, že žádné dva vrcholy spojené hranou nemají stejnou barvu. |
| k -barevný graf | Graf, který se dá obarvit k barvami. |

| Regulární graf | Všechny vrcholy stejný stupeň |
|---------------------------|--|
| Barevnost grafu | Nejmenší počet barev, které jsou potřeba k obarvení jeho vrcholů. |
| Klika | Maximální podmnožina vrcholů s vlastností, že každé dva různé vrcholy jsou spojeny hranou. |
| Klikovost grafu | Počet vrcholů v nejpočetnější klice. |
| Nezávislá množina v grafu | Množina vrcholů, jejíž žádné dva vrcholy nejsou spojeny hranou. |
| Nezávislost grafu | Počet vrcholů v maximální nezávislé množině. |

Definice neorientovaného grafu

- Neorientovaný graf je trojice $G = (V, E, \varepsilon)$, kde V je konečná množina vrcholů (též zvaných uzlů), E je konečná množina jmen hran a ε je přiřazení, které každé hraně $e \in E$ přiřazuje množinu $\{u, v\}$ pro vrcholy $u, v \in V$ a nazývá se vztah *incidence*.
- Jestliže $\varepsilon(e) = \{u, v\}$ pro $u, v \in V$, říkáme, že u, v jsou krajní vrcholy hrany e , též že jsou incidentní s hranou e .
- Je-li $u = v$, říkáme že e je (neorientovaná) smyčka.

Stromy

- Orientovaný nebo neorientovaný graf se nazývá strom, je-li souvislý a neobsahuje-li kružnici
- V každém stromu s alespoň dvěma vrcholy existuje vrchol stupně 1.
- Každý strom o n vrcholech má $n - 1$ hran.
- Strom je graf, který má nejmenší počet hran aby mohl být souvislý a současně největší počet hran aby v něm neexistovala kružnice.
- Je dán graf G , pak následující je ekvivalentní
 - G je strom
 - Graf G nemá kružnice a přidáme-li ke grafu libovolnou hranu uzavřeme přesně jednu kružnici.
 - Graf G je souvislý a odebráním libovolné hrany přestane být souvislý.

Souvislost

- Souvislý graf* je takový, kde $\forall u, v \in V$ existuje cesta z u do v
- Komponenta souvislosti* je maximální množina vrcholů grafu G taková, že podgraf indukovaný množinou A je souvislý
- Maximální množina* je taková množina vrcholů, pro kterou nebude indukovaný podgraf souvislý ve chvíli kdy přidáme libovolný další vrchol z grafu G
- Graf je *silně souvislý*

Silná souvislost

- Orientovaný graf je *silně souvislý*, pokud $\forall u, v \in V$ existuje orientovaná cesta z u do v
- Souvislý graf je silně souvislý, pokud každá jeho hrana leží v nějakém cyklu

Kosaraju, Sharir algoritmus

- Algoritmus na hledání silně souvislých komponent
- Prohledáme graf do hloubky a vypíšeme vrcholy v pořadí ve kterém jsme je opouštěli
 - V grafu G obrátíme hrany, dostaneme graf G'
 - Prohledáme graf G' do hloubky, a to v pořadí opačném pořadí v kroku 1
 - Vrcholy stromů druhého prohledávání jsou pak vrcholy jednotlivých silně souvislých komponent

Tarjanův algoritmus

- Rozšíření algoritmu DFS
- Kromě pořadových čísel jsou vrcholům přidělována i zpětná čísla
- Dělíme vrcholy do tří skupin:
 - Ještě nenavštívené vrcholy (ani pořadové, ani zpětné číslo)
 - Navštívené, ale nezařazené vrcholy
 - Vrcholy zařazené do nějaké komponenty souvislosti
- Značení:
 - Vrchol x
 - Pořadové číslo - $P(x)$
 - Zpětné číslo - $Z(x)$
 - Zásobník - ZAS

- Algoritmus

1. Všechny vrcholy jsou nenavštívené a zásobník je prázdný
2. Vybereme libovolný nenavštívený vrchol x , označíme v , přiřadíme $P(v)$, $Z(v) := P(v)$ a vložíme v na vrchol ZAS . Jestliže nenajdeme nenavštívený vrchol, výpočet končí
3. Vezmeme nějakou nepoužitou hranu e začínající v e a koncový vrchol označíme w . Pokud taková hrana neexistuje, jdeme na krok 7
4. Je-li w stále nenavštívený, přiřadíme mu $P(w)$, položíme $Z(w) := P(w)$, zařadíme w na vrchol ZAS a pokračujeme krokem 3
5. Je-li vrchol w nezařazený, zkontrolujeme zda $Z(v) > P(w)$, pokud ano tak položíme $Z(v) := P(w)$
6. Pokud byl vrchol w už zařazen, pokračujeme krokem 3
7. Pokud $Z(v) < P(v)$, pokračujeme rovnou na krok 8. Pokud $Z(v) = P(v)$ utvoříme novou komponentu silné souvislosti z dosud nezařazených vrcholů w , pro které platí $P(w) \geq P(v)$. Tyto vrcholy pak odstraníme ze zásobníku a pokračujeme na krok 8
8. Pokud je ZAS prázdný, pokračujeme na krok 2, pokud ne tak označíme x vrchol ZAS . Pokud nebyla uzavřena komponenta a $Z(x) > Z(v)$, položíme $Z(x) := Z(v)$, $v := x$ a pokračujeme krokem 3.

Minimální kostra

- Faktor grafu G je jeho podgraf, který má stejnou množinu vrcholů
- Kostra grafu G je jeho faktor, který je zároveň stromem
- Máme-li graf G s ohodnocením hran c , pak minimální kostra grafu G je taková kostra $K = (V, L)$, která minimalizuje

$$\sum_{e \in L} c(e)$$

- V každém souvislém ohodnoceném grafu existuje minimální kostra, ale nemusí být jediná

Obecný postup pro hledání minimální kostry

- Je dán souvislý graf $G = (V, E)$ a ohodnocení hran c
1. Na začátku máme $L = \emptyset$. Označíme S množinu všech komponent souvislosti grafu $K = (V, L)$; tj. na začátku je $s = v; v \in V$
 2. Dokud není graf $K = (V, L)$ souvislý (tj. dokud S se neskládá z jediné množiny), vybereme hranu e podle následujících pravidel:
 1. e spojuje dvě různé komponenty souvislosti S, S' grafu K (tj. dvě množiny z S)
 2. A pro S nebo S' je nejlevnější hranou která vede z komponenty ven
 3. Hranu e přidáme do množiny L a množiny S a S' nahradíme jejich sjednocením.
 4. Postup ukončíme, jestliže jsme přidali $n - 1$ hran (tj. jestliže se S skládá z jediné množiny).

Borůvkův-Kruskalův algoritmus

- Modifikace obecného postupu pro hledání minimální kostry:
1. Seřídíme hrany podle ceny do neklesající posloupnosti, tj. $c(e_1) \leq c(e_2) \leq \dots \leq c(e_m)$. Položíme $L = \emptyset$, $S = \{\{v\}; v \in V\}$
 2. Probíráme hrany v daném pořadí. Hranu e_i přidáme do L , jestliže má oba krajní vrcholy v různých množinách $S, S' \in S$. V S množiny S a S' nahradíme jejich sjednocením. V opačném případě hranu přeskočíme.
 3. Algoritmus končí, jestliže jsme přidali $n - 1$ hran (tj. S se skládá z jediné množiny).

Jarníkův-Primův algoritmus

- Modifikace obecného postupu pro hledání minimální kostry:
1. Vybereme libovolný vrchol v . Položíme $L = \emptyset$, $S = v$
 2. Vybereme nejlevnější hranu e , která spojuje některý vrchol x z množiny S s vrcholem y , který v S neleží. Vrchol y přidáme do množiny S a hranu e přidáme do L .
 3. Opakujeme krok 2 dokud nejsou všechny vrcholy v množině S .

Eulerovy grafy

- Tah je sled, ve kterém se neopakují hrany
- Eulerovský tah je takový, který prochází každou hranou grafu
- Uzavřený eulerovský tah je takový tah, kde se poslední vrchol shoduje s prvním (výjimka pravidla, že se vrcholy nesmí opakovat), jinak mluvíme o otevřeném eulerovském tahu
- Eulerovský graf je graf, ve kterém existuje eulerovský tah