Důkazové systémy a rezoluce

Jindřich Matuška

Faculty of Informatics, Masaryk University

14. listopadu 2024

Čas na odpovědníky

Obsah

Obecná rezoluce ve výrokové logice

SLD rezoluce ve výrokové logice

Obecná rezoluce v predikátové logice

Dopředné a zpětné řetězení

Obsah

Obecná rezoluce ve výrokové logice

SLD rezoluce ve výrokové logice

Obecná rezoluce v predikátové logice

Dopředné a zpětné řetězení

S čím pracujeme

Pracujeme s klauzulemi

- Množiny literálů $\{l_1, \ldots, l_n\}$
  - Ekvivalentní zápis $\bigvee_{i \in \{1,\ldots,n\}} l_i$
- Každý literál je výroková proměnná (p) nebo její negace ($\neg p$)
- Speciálně symbol $\square$ je prázdná klauzule (kontradikce)
- Množinu klauzulí chápeme jako jejich konjunkci

Rezoluční pravidlo

Uvažujme klauzule $C_1 = \{p, l_1, \ldots, l_n\}$ a $C_2 = \{\neg p, l'_1, \ldots, l'_m\}$, kde l_i, l'_j jsou literály. Jejich rezolventou je klauzule $C = \{l_1, \ldots, l_n, l'_1, \ldots, l'_m\}$.

Rezolventa dvou klauzulí zachovává pravdivost v interpretaci.

$$I(C_1) = 1 \wedge I(C_2) = 1 \implies I(C) = 1$$

Obrácená implikace nutně neplatí.

Rezoluční pravidlo

Uvažujme klauzule $C_1 = \{p, l_1, \dots, l_n\}$ a $C_2 = \{\neg p, l'_1, \dots, l'_m\}$, kde l_i, l'_j jsou literály. Jejich rezolventou je klauzule $C = \{l_1, \dots, l_n, l'_1, \dots, l'_m\}$.

Rezolventa dvou klauzulí zachovává pravdivost v interpretaci.

$$I(C_1) = 1 \wedge I(C_2) = 1 \implies I(C) = 1$$

Obrácená implikace nutně neplatí.

Proč? (Příklad 7.1.4)

Rezoluční důkaz C z množiny klauzulí S

Konečná posloupnost klauzulí $C_1, \ldots, C_n$, kde $C_n = C$ a pro každé C_i:

- $C_i \in S$ nebo
- C_i vzniklo aplikací rezolučního pravidla na předcházející klauzule

Pokud existuje takový rezoluční důkaz, řekneme že C je rezolučně dokazatelná z S, zapisujeme $S \vdash_R C$.
Pokud existuje důkaz $S \vdash_R \square$, nazveme tento důkaz vyvrácením S.
Rezoluční důkazy zapisujeme často pomocí stromu.

Důkaz klauzule $C = \{\neg q\}$ z množiny
$S = \{\{p, r\}, \{\neg s, \neg r\}, \{\neg q, s\}, \{\neg p\}\}$

Příklad 7.1.1

Obecnou rezolucí ukažte, že

a) formule $(p \vee q) \wedge (\neg q \vee \neg r) \wedge (\neg p \vee s) \wedge (\neg s) \wedge (r \vee s)$ je nesplnitelná,

b) platí logické vyplývání $\{(\neg s \wedge t) \Rightarrow r, \neg s, \neg s \Rightarrow t\} \models r$.

Rezoluční vyvrácení

Pro množinu formulí $\mathcal{T}$ a formuli φ platí $\mathcal{T} \models \varphi$, právě když $\mathcal{T} \cup \{\neg\varphi\}$ je nesplnitelná.

Obsah

Obecná rezoluce ve výrokové logice

SLD rezoluce ve výrokové logice

Obecná rezoluce v predikátové logice

Dopředné a zpětné řetězení

SLD rezoluce

Lineární rezoluce na uspořádaných klauzulích s výběrovým pravidlem.

Rezoluční strom SLD rezoluce je ve tvaru jedné větve.
SLD strom zachycuje všechna možná vyhodnocení.

Výběové pravidlo určuje na kterém literálu rezolvujeme.

- V tomto předmětu **vždy první literál** v klauzuli.

SLD rezoluce je úplná pro Hornovy klauzule

- každá klauzule má nejvýše 1 pozitivní literál
- *fakta* – pouze pozitivní literál, $[p]$
- *pravidla* – pozitivní literál a alespoň jeden negativní, $[p, \neg q, \neg r]$
- *cíle* – pouze negativní literály, $[\neg q, \neg r]$

Příklad

Pomocí SLD rezoluce vyvraťte množinu

$$\{[\neg s, \neg u], [q, \neg r], [q, \neg t], [q], [r], [u, \neg q], [s]\}$$

Nakreslete SLD strom systematického hledání řešení. Nakreslete rezoluční strom pro nejlevější úspěšnou větev SLD stromu.

Příklad 7.2.1

Vysvětlete intuitivní význam jednotlivých typů Hornových klauzulí, tedy *faktů*, *pravidel* a *cílů*.

Příklad 7.2.2

SLD rezolucí ukažte nesplnitelnost množiny uspořádaných klauzulí $\{[q, \neg s, \neg r], [p, \neg q], [s], [r]\} \cup \{[\neg p, \neg s]\}$.

Obsah

Obecná rezoluce ve výrokové logice

SLD rezoluce ve výrokové logice

Obecná rezoluce v predikátové logice

Dopředné a zpětné řetězení

Obecná rezoluce v predikátové logice

Na formulích ve skolemově normální formě.
Klauzule opět zapisujeme jako množiny literálů.
Při rezoluci používáme substituci (nahrazení) za termy

- Nahradit můžeme obecnější term za specifičtější
- Proměnnou za libovolný term

Substituce

- Množina $\{x_1/t_1, \ldots, x_n/t_n\}$
- x_i jsou proměnné, t_i jsou termy
- Přejmenování proměnných, pokud všechny t_i jsou proměnné
- Unifikátor S, pokud $|S\Phi| = 1$

$$S_1 = \{P(x,y), \neg Q(y, f(x))\}, S_2 = \{P(f(x)), P(y), P(f(f(c)))\}$$

$$\Phi_1 = \{x/c, y/b\}, \Phi_2 = \{x/f(c), y/f(f(c))\}$$

Rezoluční pravidlo pro predikátovou logiku

Pro dvě klauzule **bez společných proměnných**

$$\{P(\vec{t_1}), \dots, P(\vec{t_k}), L_1, \dots, L_n\} \text{ a } \{\neg P(\vec{t'_1}), \dots, \neg P(\vec{t'_l}), L'_1, \dots, L'_m\}$$

a substituci Φ, která je unifikátorem množiny

$$\{P(\vec{t_1}), \dots P(\vec{t_k}), P(\vec{t'_1}), \dots P(\vec{t'_l})\}$$

je jejich rezolventou

$$\{L_1, \dots, L_n, L'_1, \dots, L'_m\}\Phi$$

Příklad 7.3.1

Nalezněte $S\Phi$ pro zadaná S a Φ.

a) $S = \{P(x), Q(y)\}, \Phi = \{x/y, y/x\}$
b) $S = \{P(c, x), \neg P(x, f(z))\}, \Phi = \{f(z)/x, x/d\}$
c) $S = \{P(x, y), P(c, z), P(c, c)\}, \Phi = \{x/y, y/z, z/c\}$

Příklad 7.3.2

Nalezněte (nejobecnější) unifikátory následujících množin literálů (a, b, c, d jsou konstanty).

a) $S = \{P(x), P(y)\}$

b) $S = \{P(x), Q(y)\}$

c) $S = \{Q(x, x), Q(y, c)\}$

d) $S = \{Q(h(x, y), w), Q(h(g(v), a), f(v)), Q(h(g(v), a), f(b))\}$

Příklad 7.3.5

Uvažujte následující tvrzení:

- „Existuje drak.“
- „Draci spí nebo loví.“
- „Když mají draci hlad, nespí.“

Rezolucí dokažte, že z uvedených tvrzení plyne „Když má drak hlad, loví,“ a to dvěma způsoby: (1) přímým odvozením závěru a (2) vyvrácením množiny obsahující předpoklady a negaci závěru. Zejména v řešení

a) formalizujte uvedená tvrzení (předpoklady a závěr) v jazyce predikátové logiky,
b) určete význam (interpretaci) jednotlivých predikátů,
c) proveďte skolemizaci tvrzení a zapište je v klauzulárním tvaru a
d) nakonec proveďte samotnou rezoluci.

Obsah

Obecná rezoluce ve výrokové logice

SLD rezoluce ve výrokové logice

Obecná rezoluce v predikátové logice

Dopředné a zpětné řetězení

Dopředné a zpětné řetězení

Báze znalostí – Hornovy klauzule bez omezení

- Lze zapsat jako množina implikací a faktů

Dopředné řetězení – začínáme u faktů, pomocí implikací odvozujeme další fakty

Zpětné řetězení – začínáme u dotazu, koukáme na možná pravidla odvození

Příklad 7.4.1

Proveďte algoritmus dopředného řetězení pro dotaz E, je-li zadána následující báze znalostí:

- $(B \wedge D) \Rightarrow E$
- $A \Rightarrow B$
- $(A \wedge F) \Rightarrow D$
- $F \Rightarrow A$
- $C \Rightarrow B$
- F