AD4M33AU
Automatické uvažování

Unifikace, optimalizace rezolučního kalkulu

Petr Pudlák

Unifikace

- Problém: Mějme množinu neuspořádaných dvojic výrazů

  $$S = \{s_1 \approx t_1, ..., s_n \approx t_n\}$$

  Hledáme nejobecnější substituci θ takovou, že pro všechna i, $1 \leq i \leq n$, platí $\theta s_i \equiv \theta t_i$.
- Použití:
  - Rezoluční kalkulus.
  - Prolog.
  - Odvozování typů ve funkcionálních jazycích.

Unifikační algoritmus

- $U(\{\})$ $\Rightarrow$ identická substituce
- $U(\mathbf{S'} \cup \{t \approx t\})$ $\Rightarrow U(\mathbf{S'})$
- $U(\mathbf{S'} \cup \{f(s_{1,} ..., s_n) \approx f(t_1, ..., t_n)\})$

  $\Rightarrow U(\mathbf{S'} \cup \{s_1 \approx t_1, ..., s_n \approx t_n\})$
- $U(\mathbf{S'} \cup \{f(s_{1,} ..., s_n) \approx g(t_1, ..., t_m)\})$ $\Rightarrow$ <u>nelze</u>

  pokud $f \not\equiv g$ nebo $n \neq m$
- $U(\mathbf{S'} \cup \{X \approx t\})$ $\Rightarrow$ <u>nelze</u>

  pokud $X \in Vars(t)$
- $U(\mathbf{S'} \cup \{X \approx t\})$ $\Rightarrow \sigma \circ \theta$ což je $\sigma \cup \{X \mapsto \sigma t\}$

  pokud $X \notin Vars(t)$,

  kde $\sigma = U(\theta \mathbf{S'})$ a $\theta = \{X \mapsto t\}$

Unifikační algoritmus vždy skončí

- Ke každému kroku algoritmu přiřaďme trojici čísel:

1. Počet různých proměnných v **S**.
2. Celkový počet funkčních symbolů v **S**.
3. Počet dvojic v **S**.

- Tato trojice čísel se v každém kroku algoritmu zmenšuje vzhledem k lexikografickému uspořádání.

Věta o nalezení MGU unifikačním algoritmem

- Věta: Popsaný unifikační algoritmus najde MGU množiny **S**.
- Důkaz:
  - Buď σ unifikátor **S**. Pak platí $\sigma = \sigma \circ U(\mathbf{S})$.
  - Dokážeme indukcí podle počtu kroků algoritmu.
  - Zkoumejme zejména poslední krok: σ unifikuje $\mathbf{S'} \cup \{X \approx t\}$, takže $X \notin Vars(t)$ [proč?] a $\sigma X \equiv \sigma t$. Buď $\theta = \{X \mapsto t\}$. Pak σ unifikuje $\theta\mathbf{S'}$, takže podle indukčního předpokladu $\sigma = \sigma \circ U(\theta\mathbf{S'})$. Odtud:
    $\sigma \circ U(\mathbf{S'} \cup \{X \approx t\}) = \sigma \circ U(\theta S') \circ \theta = \sigma \circ \theta = \sigma$

Rezoluce s rovností

- Ve velké části případů se v zadání vyskytuje rovnost.
- Jak formalizovat rovnost v logice 1. řádu?

Pro zajímavost: definice „=“ v logice 2. řádu

$$X = Y \quad \Leftrightarrow \quad (\forall p)\,(p(X) \Leftrightarrow p(Y))$$

- Zde kvantifikujeme *p* přes všechny možné predikáty!

Klasické zavedení „=“

- Přidáme axiomy pro:
  - reflexivitu
  - symetrii
  - tranzitivitu
- Přidáme pro všechny predikátové a funkční symboly axiomy upravující záměnu dvou navzájem rovných termů:
  - $(X = Y) \Rightarrow (f(X_1, \ldots, X_{i-1}, \underline{X}, X_{i+1}, \ldots, X_n) = $
    $= f(X_1, \ldots, X_{i-1}, \underline{Y}, X_{i+1}, \ldots, X_n))$
  - $(X = Y) \Rightarrow (p(X_1, \ldots, X_{i-1}, \underline{X}, X_{i+1}, \ldots, X_n) \Rightarrow$
    $\Rightarrow p(X_1, \ldots, X_{i-1}, \underline{Y}, X_{i+1}, \ldots, X_n))$

Problém klasického zavedení

- Čím více máme symbolů a čím je větší jejich arita, tím více dodatečných klauzulí přidáváme.
- Prohledávací strom pro dokazovač velmi rychle narůstá.
- Důkaz jednoduchého tvrzení je komplikovaný, například i

$$(X = Y) \Rightarrow (p(f(g(X))) \Rightarrow p(f(g(Y))))$$

Pravidlo paramodulace

- **Paramodulace:**

$$\frac{A \mid (s = t) \qquad B[u]}{\theta A \mid \theta B[t]}$$

  kde:
  - *θ* je MGU termů *s* a *u*,
  - *B[t]* je formule vzniklá z *B[u]* nahrazením jednoho výskytu *u* pomocí *t*,
- Výslednou klauzuli nazýváme **paramodulant.**

Pravidlo rezoluce reflexivity

- **Rezoluce reflexivity:**

$$\frac{A \mid \sim(s = t)}{\theta A}$$

kde θ je MGU termů s a t.

Úplnost paramodulace

Věta: Pravidla

- rezoluce,
- faktorizace,
- paramodulace,
- rezoluce reflexivity

dohromady tvoří kalkulus, který vždy odvodí spor ze sporné množiny klauzulí s rovností.

Další pokročilejší mechanismy pro práci s rovností

- Existují složitější mechanismy pro efektivnější práci s rovností, například **superpozice**.
- Používají vhodná kvazi-uspořádání na termech.

Základní optimalizace rezoluce

1. Zkrácení výsledných klauzulí při konverzi na CNF (viz. výroková rezoluce).
2. Snížení arity funkčních symbolů při skolemizaci (viz. příklad na minulé přednášce).
3. Subsumpce.
4. Omezit rezoluci jen na některé klauzule.

Subsumpce v logice 1. řádu

- Jestliže existuje substituce θ taková, že $\theta\varphi\subseteq\psi$ ve smyslu množin literálů, říkáme, že klauzule φ **subsumuje** klauzuli ψ.
  Značme $\varphi\sqsubseteq\psi$.
- Jestliže $\varphi\sqsubseteq\psi$ pak $\varphi \models \psi$.
  Naopak to neplatí.
- Pro množiny klauzulí **A** a **B** řekneme, že **A** subsumuje **B** (**A**$\sqsubseteq$**B**), pokud každou formuli z **B** subsumuje nějaká formule z **A**.

Subsumpce v logice 1. řádu – příklady

- $p(X) \sqsubseteq p(c)$
- $p(X) \sqsubseteq p(X)|q(X,Y)$
- $p(X) \sqsubseteq p(f(c))|r(X,f(Y))$
- $p(X) \mid q(Z) \sqsubseteq p(X) \mid q(X)$

  ale neplatí $p(X) \mid q(X) \sqsubseteq p(X) \mid q(Z)$!

Subsumpce v rezoluci

- Pokud existuje rezoluční důkaz sporu z **B** a pokud **A**$\sqsubseteq$**B**, pak existuje rezoluční důkaz sporu z **A**, který není delší než důkaz z **B**.
- Myšlenka: Formule z **A** jsou „stejně silné nebo silnější“ než formule v **B**, takže důkaz z **A** „není těžší“ než z **B**.
- <u>Použití:</u> Pokud jsme dovodili dvě formule φ a ψ, a pokud $\varphi \sqsubseteq \psi$, ponecháme si pouze φ a ψ můžeme zahodit.

Subsumpce v logice 1. řádu – příklad

- Mějme rezoluční důkaz:

```
p(f(X)) | q(X,Y) | r(X)    ~p(f(f(c))
----------------------------------------
        q(f(c), Y) | r(f(c))              ~q(U,V)
        ---------------------------------------------
                r(f(c))                              ~r(f(c))
                ---------------------------------------------
                                  □
```

Subsumpce v logice 1. řádu – příklad (pokračování)

- Odvodíme-li mezitím klauzuli *p(Y) | r(Z)*, můžeme klauzuli *p(f(X)) | q(X,Y) | r(X)* nahradit klauzulí *p(Y) | r(Z)* [protože *p(Y) | r(Z)* $\sqsubseteq$ *p(f(X)) | q(X,Y) | r(X)*], a pak mechanicky upravit důkaz, který bude v tomto případě i o jeden krok kratší – nebude třeba provádět rezoluci s *~q(U,V)*:
- p(Y) | r(Z)   ~p(f(f(c))

```
p(Y) | r(Z)     ~p(f(f(c))
----------------------------------------
            r(Z)                              ~r(f(c))
            ------------------------------------------------
                                  □
```

Zpětná a dopředná subsumpce

Stejná myšlenka jako ve výrokové rezoluci:

- **Dopředná (forward) subsumpce:** Pokud odvodíme klauzuli ψ, a ψ je subsumována nějakou z již dříve odvozených klauzulí φ (tedy $\varphi \sqsubseteq \psi$), klauzuli ψ zahodíme.
  (Cokoliv bychom dokázali z ψ můžeme dokázat i z φ.)
- **Zpětná (backward) subsumpce:** Pokud odvodíme klauzuli φ, klauzulí φ nahradíme všechny doposud odvozené klauzule ψ_i, které jsou subsumovány φ (tedy $\varphi \sqsubseteq \psi_i$).