AD4M33AU

Automatické uvažování

Metody aplikace nástrojů automatického uvažování

Petr Pudlák

Často lze jednu úloh

- Často lze úlohu formalizovat různými způsoby.
- Obtížnost řešení různých způsobů bývá pro dokazovač často velmi odlišná.

Rozdělení domněnky na co nejjednodušší části

- Čím jednodušší domněnka, tím snáze ji dokazovač dokáže.
- Chceme-li v teorii ***A*** dokázat domněnku $\overline{\varphi}$ tvaru

  φ_1 & ... & φ_n,

  bude dokazovač úspěšnější, pokud budeme postupně dokazovat jednotlivé φ_i než pokud se budeme snažit dokázat celou $\overline{\varphi}$ najednou.
- (Příklad.)

Použití už dokázaných tvrzení jako lemmat

- Navíc můžeme už dokázaná tvrzení použít jako lemmata pro další důkazy.
- Pokud jednou dokážeme, že $\boldsymbol{A} \models \varphi_1$, můžeme pro důkaz φ_2 použít množinu $\boldsymbol{A} \cup \{\varphi_1\}$ atd.

Použít pouze axiomy nutné k důkazu

- Pokud víme, které axiomy jsou k důkazu nutné a které ne, dáme dokazovači pouze ty nutné.
- Výrazně se tím zmenší velikost prohledávacího prostoru.

Jaké předpoklady použít a jaké ne?

- Shrnutí – čím lze zefektivnit hledáním důkazu:
  - <u>Odebráním</u> nadbytečných axiomů, které nejsou potřebné pro důkaz. Nutno vědět, které axiomy jsou nadbytečné!
  - <u>Přidáním</u> vhodných pomocných tvrzení (lemmat). Rozpoznat vhodné lemma není jednoduché, u většiny tvrzení jejich přidání nepomůže.
- Co přidávat nebo odebírat?
- Závisí na typu problému, axiomech atp.

Vhodná reprezentace

- Objekty, které nepotřebujeme kvantifikovat, nemusíme přidávat jako prvky univerza.
- Místo toho je můžeme reprezentovat pomocí vhodných predikátů.

Převod problému do výrokové logiky

- Pokud je doména problému konečná a malá, může být často efektivnější převést problém do výrokové logiky.
- (Příklad.)

Využití dokazovačů pro nestandardní logiky

- Často je potřeba pracovat v nestandardních logikách, například:
  - modální logiky,
  - temporální logiky

Příklad: Lineární temporální logika

- V jazyce jsou navíc <u>logické</u> operátory:
  - X*φ* (také ○*φ*): *φ* musí platit v dalším stavu.
  - G*φ* (také □*φ*): *φ* musí platit ve všech následujících stavech (**g**lobálně).
  - F*φ* (také ◊φ): *φ* jednou (**f**inally) musí platit.
- Binary operators:
  - *ψ* U *φ*: *ψ* musí platit odteď do nějakého stavu v budoucnu, od kdy platí φ (**u**ntil).
  - *ψ* R *φ*: *φ* musí platit do stavu (včetně), ve kterém poprvé platí *ψ*; pokud *ψ* nikdy v budoucnu neplatí, *φ* musí platit navždy.

LTL – vztahy

- Operátory lze definovat pouze pomocí U a X:
  - F φ = true U φ
  - G φ = false R φ = ~F~φ
  - ψ R φ = ~(~ψ U ~φ)

Vnoření LTL do logiky 1. řádu

- Prvky univerza jsou časové okamžiky.
- Přidáme axiomy lineárního uspořádání.
- Přidáme axiomy pro pojem následného stavu:
  - ![X]: (less(X,succ(X)))
  - ![Y]: (less(Y,X) | less(succ(X), Y))
- Každá výroková proměnná z LTL se zakóduje jako unární predikát.
- Temporální operátory se přepíší pomocí operací s časem, např. GF*φ* se přepíše jako
  - ![X]: (less(T, X) => ?[Y]: (less(X,Y) & φ(Y)))

Příklad vnoření LTL

- Co je silnější:

  FGp ?? GFp

Příklad vnoření LTL

- Dokažme, že:

  FGp => Gfp
- ![T]: (

```
        ?[X]: (less(T, X)
                & ![Y]: (less(X, Y) => p(Y))
            )
      =>
        ![X]: (less(T, X)
                => ?[Y]: (less(X, Y) & p(Y))
            )
  )
```

Vnoření modálních logik do predikátové logiky

- LTL má danou sémantiku na časové ose, proto ji lze snadno vnořit do logiky 1. řádu.
- Jiné (například obecné modální) logiky takovou sémantiku nemají, a jsou pouze zadávány.
- Nelze je proto takto přímočaře převést do logiky 1. řádu.

Vnoření modálních logik do predikátové logiky

- Do logiky 1. řádu nemůžeme přidávat logické operátory, ale můžeme přidávat funkční symboly.
- Zakódujeme formule modální výrokové logiky pomocí termů logiky 1. řádu.
- Každý logický operátor zakódujeme funkčním symbolem.
- Výrokové proměnné zakódujeme proměnnými logiky 1. řádu.
- Zavedeme predikát zachycující pravdivost formule.
- (Příklad.)