Engenharia de Computação

ECM253 – Linguagens Formais, Autômatos e Compiladores

Métodos de prova em lógica de predicados

Slides da disciplina ECM253 – Linguagens Formais, Autômatos e Compiladores
Curso de Engenharia de Computação
Instituto Mauá de Tecnologia
Prof. Marco Antonio Furlan de Souza

Valididade

- Uma **fbf proposicional sempre** tem **valor-verdade** (depende dos valores-verdade atribuídos aos símbolos), enquanto que uma **fbf predicativa pode ter ou não** ter um **valor-verdade** (depende da interpretação);
- Pode haver um **número infinito de interpretações possíveis** para as **fbfs predicativas** e apenas 2^n linhas para **fbfs proposicionais** com n símbolos proposicionais;
- O análogo à **tautologia** para as **fbfs predicativas** é a **validade** – uma **fbf predicativa é válida se** for **verdadeira** para **qualquer interpretação possível**;
- **Denota-se** por $\vDash A$ quando uma **expressão** A é **valida**;
- Se existe uma **interpretação** que faz uma **expressão** A ser **verdadeira**, então diz-se que tal interpretação **satisfaz** A ou que A é **satisfazível**;
- Como o **número de interpretações pode ser infinito**, **não há um algoritmo** em **lógica de predicados** para se **determinar** se uma **fbf é válida**.

Exemplos de fbfs da lógica de predicados e validade

- $(\forall x)P(x) \rightarrow (\exists x)P(x)$ é **válida**, pois se é verdade que o predicado é verdadeiro para todo x então é também para pelo menos um;
- $(\forall x)P(x) \rightarrow P(a)$ é **válida**, pois se é verdade que o predicado é verdadeiro para todo x então é também para o caso particular de um indivíduo a;
- $P(x) \rightarrow (Q(x) \rightarrow P(x))$ é **válida**, mesmo contendo a **variável livre** x;
- $(\exists x)P(x) \rightarrow (\forall x)P(x)$ **não é válida**, pois não se pode generalizar que se o predicado é válido para pelo menos um valor, então ele então será para todos.

- **Exercícios**
  - Decidir se cada uma das fbs a seguir é válida ou não, justificando.
    (a) $(\exists x)A(x) \leftrightarrow \neg((\forall x)\neg A(x))$
    (b) $(\forall x)A(x) \leftrightarrow \neg((\exists x)(\neg A(x)))$
    (c) $(\forall x)(P(x) \vee Q(x)) \leftrightarrow (\forall x)P(x) \vee (\exists y)Q(y)$

- **Regras de equivalência**
  - São as **mesmas da lógica proposicional**, mas se **acrescentam** as seguintes **equivalências** (provadas anteriormente):
    - $\neg(\exists x)A(x) \leftrightarrow (\forall x)\neg A(x)$ (negação do quantificador existencial, ne)
    - $\neg(\forall x)A(x) \leftrightarrow (\exists x)(\neg A(x))$ (negação do quantificador universal, nu)
  - **Exemplo de aplicação** (trecho de prova)

1. $(\forall x)R(x)$	(hipótese)
2. $(\forall x)R(x) \rightarrow (\forall x)S(x)$	(hipótese)
3. $(\forall x)S(x)$	1,2, mp
4. $\neg\neg(\forall x)S(x)$	3, dn
5. $\neg((\exists x)(\neg S(x)))$	4, nu

Regras de inferência

- Utilizam-se as mesmas da lógica proposicional, e acrescentam-se as regras a seguir:

Regras de inferência			
De	**Pode derivar**	**Nome/abreviação**	**Restrição no uso**
$(\forall x)P(x)$	$P(t)$ onde t é símbolo de variável ou constante.	Instanciação universal/ ui	Se t é uma variável, não pode aparecer no escopo de um quantificador para t.
$(\exists x)P(x)$	$P(\mathsf{a})$ onde a é um símbolo constante não utilizado anteriormente na sequência de prova.	Instanciação existencial/ ei	Deve ser a primeira regra empregada para introduzir a na sequência.
$P(x)$	$(\forall x)P(x)$	Generalização universal/ ug	$P(x)$ não deve ter sido deduzido de qualquer hipótese onde x é livre e nem ter sido deduzido por instanciação existencial de qualquer fbf no qual x é livre.
$P(x)$ ou $P(\mathsf{a})$, a constante	$(\exists x)P(x)$	Generalização existencial/ eg	Para sair de $P(a)$ para $(\exists x)P(x)$, x não deve aparecer em $P(a)$.

Regras de inferência

- **Exemplo**. Provar o argumento: *Todos os humanos são mortais. Sócrates é humano. Portanto Sócrates é mortal*.
- Se $H(x)$ representa x é humano, $M(x)$ representa x é mortal, e s é um símbolo constante a que foi atribuído Sócrates, então pode-se reescrever o argumento assim: $(\forall x)(H(x) \to M(x)) \land H(s) \to M(s)$.
- A partir daí, elabora-se a sequência de prova:

1. $(\forall x)(H(x) \to M(x))$	(hipótese)
2. $H(s)$	(hipótese)
3. $H(s) \to M(s)$	1,ui
4. $M(s)$	2,3,mp

Regras de inferência

- **Exemplo.** Provar o argumento: $(\forall x)(P(x) \to Q(x)) \land (\forall x)P(x) \to (\forall x)Q(x)$.
- Sequência de prova:

1. $(\forall x)(P(x) \to Q(x))$	(hipótese)
2. $(\forall x)P(x)$	(hipótese)
3. $P(x) \to Q(x)$	1,ui
4. $P(x)$	2,ui
5. $Q(x)$	3,4,mp
6. $(\forall x)Q(x)$	5,ug

Exercícios

- Provar os argumentos a seguir:

  (a) $(\exists x)P(x) \wedge (\forall x)(P(x) \to Q(x)) \to (\exists x)Q(x)$

  (b) $(\exists x)R(x) \wedge \neg((\exists x)(R(x) \wedge S(x))) \to (\exists x)(\neg S(x))$

Referências bibliográficas

[1] GERSTING, J.L. **Fundamentos matemáticos para a ciência da computação**. 4.ed. Rio de Janeiro, RJ: LTC, 2001. 538 p. ISBN 85-216-1263-X.