Lógica Computacional 1
Segunda Prova

Tópicos: Lógica de Predicados - dedução e semântica

Instituto de Ciências Exatas, Universidade de Brasília
Departamento de Ciência da Computação

28 de novembro de 2011

Profs. Mauricio Ayala Rincón e Flávio L. C. de Moura

Duração: 100 min
Início: 16:00h - Término: 17:45h

1. (4 pontos) A lógica de primeira ordem diferencia-se da lógica proposicional principalmente devido à presença dos predicados (e não mais proposições) e dos quantificadores. Tais elementos dão um poder de expressividade maior a esse tipo de linguagem. Isso acaba gerando, como em toda linguagem expressiva, diversas maneiras equivalentes de se representar um determinado fato. Por exemplo, a sentença

   Nem todo programa de computador é correto.

   é equivalente a

   Existe um programa de computador que não é correto.

   Seja p um predicado unário interpretado como: "x é um programa de computador correto", podemos representar as duas expressões equivalentes como $\neg\forall x(p(x)) \equiv \exists x(\neg p(x))$. Construa uma prova em dedução natural para os sequentes

   (a) (2 pontos) $\exists x(\neg p(x)) \vdash \neg\forall x(p(x))$

   (b) (2 pontos) $\neg\forall x(p(x)) \vdash \exists x(\neg p(x))$

   indicando a regra utilizada em cada passo da derivação.

   Solução:

   (a)

$$\dfrac{\exists x(\neg p(x)) \qquad \dfrac{\dfrac{[\neg p(x_0)]^a \qquad \dfrac{[\forall x(p(x))]^b}{p(x_0)}\,(\forall e)}{\bot}\,(\neg e)}{\neg\forall x(p(x))}\,(\neg i)\,b}{\neg\forall x(p(x))}\,(\exists e)\,a$$

(b)

$$
\begin{prooftree}
\AxiomC{$[\neg\exists x(\neg p(x))]^a$}
\AxiomC{$[\neg p(x_0)]^b$}
\RightLabel{$(\exists i)$}
\UnaryInfC{$\exists x(\neg p(x))$}
\RightLabel{$(\neg e)$}
\BinaryInfC{$\bot$}
\RightLabel{$PBC\ b$}
\UnaryInfC{$p(x_0)$}
\RightLabel{$(\forall i)$}
\UnaryInfC{$\forall x(p(x))$}
\AxiomC{$\neg\forall x(p(x))$}
\RightLabel{$(\neg e)$}
\BinaryInfC{$\bot$}
\RightLabel{$PPC\ a$}
\UnaryInfC{$\exists x(\neg p(x))$}
\end{prooftree}
$$

2. (4.0 pontos) Em uma aula de lógica, o professor pediu aos alunos que representassem a sentença

Todo pássaro voa.

como uma fórmula da lógica de predicados utilizando os predicados unários p e v com a seguinte semântica:

$p(x) : x$ é pássaro
$v(x) : x$ voa.

Surgiram duas respostas $\forall x(p(x) \to v(x))$ e $\forall x(p(x) \wedge v(x))$. Estas respostas são equivalentes? Em outras palavras, a equivalência $\forall x(p(x) \to v(x)) \equiv \forall x(p(x) \wedge v(x))$ é válida? Sabendo que a lógica de predicados é correta e completa, isto é,

$$\Gamma \vdash \varphi \text{ se, e somente se, } \Gamma \models \varphi \text{ para qualquer fórmula } \varphi \text{ e conjunto de fórmulas } \Gamma,$$

responda esta pergunta analisando os sequentes a seguir da seguinte forma: se o sequente for válido construa uma prova em dedução natural, caso contrário construa uma estrutura que não seja modelo do sequente, isto é, uma estrutura que satisfaça as hipóteses, mas não satisfaça o consequente.

(a) (2 pontos) $\forall x(p(x) \wedge v(x)) \vdash \forall x(p(x) \to v(x))$ e

(b) (2 pontos) $\forall x(p(x) \to v(x)) \vdash \forall x(p(x) \wedge v(x))$

Solução:

(a)

$$
\begin{prooftree}
\AxiomC{$[p(x_0)]^a$}
\AxiomC{$\forall x(p(x) \wedge v(x))$}
\RightLabel{$(\forall e)$}
\UnaryInfC{$p(x_0) \wedge v(x_0)$}
\RightLabel{$(\wedge e_2)$}
\UnaryInfC{$v(x_0)$}
\RightLabel{$(\wedge i)$}
\BinaryInfC{$p(x_0) \wedge v(x_0)$}
\RightLabel{$(\wedge e_2)$}
\UnaryInfC{v_0}
\RightLabel{$(\to i)\ a$}
\UnaryInfC{$p(x_0) \to v(x_0)$}
\RightLabel{$(\forall i)$}
\UnaryInfC{$\forall x(p(x) \to v(x))$}
\end{prooftree}
$$

(b) Várias estruturas são possíveis. Considere por exemplo, a estrutura M com domínio $D = \{a, b\}$ e $p^D = \{a\}$ e $v^D = D$. Neste caso, temos que $M \models \forall x(p(x) \to v(x))$, mas $M \not\models \forall x(p(x) \wedge v(x))$ uma vez que a conjunção $p(x) \wedge v(x)$ é falsa se interpretamos x como b. Portanto o sequente não é válido, e as fórmulas $\forall x(p(x) \wedge v(x))$ e $\forall x(p(x) \to v(x))$ não são equivalentes.

3. (2 pontos) A lógica de predicados é mais expressiva que a lógica proposicional, mas também tem suas limitações. Neste sentido, os teoremas de Löwenheim-Skolem e de compacidade têm um papel importante e caraterizam a semântica das linguagens de primeira ordem. Em particular, não é possível expressar a noção de finito e contável infinito na linguagem desta lógica. Enuncie e demonstre o teorema de Löwenheim-Skolem.

   Ajuda: O teorema estabelece que para qualquer sentença da lógica de predicados que tenha modelos de cardinalidade pelo menos n, para qualquer $n \in \mathbb{N}$, existem modelos infinitos. Considere o conjunto de fórmulas

$$\Gamma := \{\psi\} \cup \{\phi_n \mid n \in \mathbb{N}\}$$

   onde $\phi_n := \exists x_1 \ldots \exists x_n \bigwedge_{1 \leq i < j \leq n} \neg(x_i = x_j)$ e utilize o teorema da compacidade.

   Solução: A fórmula $\phi_n := \exists x_1 \ldots \exists x_n \bigwedge_{1 \leq i < j \leq n} \neg(x_i = x_j)$ especifica a existência de pelo menos n elementos. Observe primeiro que qualquer subconjunto finito $\Gamma_0 \subset \Gamma$ é satisfatível: Seja k maior que o índice de qualquer ϕ_n em Γ_0. São dois casos a considerar.

   - $\psi \notin \Gamma_0$, selecione qualquer modelo de cardinalidade k. Este satisfará Γ_0.
   - $\psi \in \Gamma_0$, selecione um modelo de ψ de cardinalidade maior ou igual que k. Este satisfará Γ_0.

   Dessa forma, conclui-se que qualquer subconjunto finito de Γ é satisfatível, o que implica, pelo teorema de compacidade, que o é. Mas um modelo de Γ deve ter cardinalidade infinita, uma vez que todas as fórmulas ϕ_n para $n \in \mathbb{N}$, valem nesse modelo. Dessa forma (como ψ também está em Γ), esse modelo infinito é também modelo da fórmula ψ.