Recuperatorio de Lógica

Lógica y Computabilidad

2do cuatrimestre de 2010

Este examen se aprueba obteniendo al menos **50 puntos**. El parcial es a libro abierto y se puede suponer demostrado todo lo que se dio en clase, colocando referencias claras. En el caso de usar resultados de las guías de ejercicios, deben incluirse las demostraciones.

Ejercicio 1. Considere los conectivos proposicionales binarios $\leftrightarrow$ y $\dot\vee$, que llamaremos informalmente "si y sólo si" y "ó-exclusivo", cuyas cláusulas semánticas están dadas por:

$$\begin{array}{lll} v \models \alpha \leftrightarrow \beta & \text{sii} & \text{o bien } v \models \alpha \text{ y } v \models \beta, \text{ o bien } v \not\models \alpha \text{ y } v \not\models \beta \\ v \models \alpha \,\dot\vee\, \beta & \text{sii} & \text{o bien } v \models \alpha \text{ y } v \not\models \beta, \text{ o bien } v \not\models \alpha \text{ y } v \models \beta \end{array}$$

a) (**15 p.**) Demostrar que existen dos fórmulas α, β de *una* variable proposicional con conectivos $\{\neg, \rightarrow\}$ tal que α que no es expresable sólo con $\{\leftrightarrow\}$ y β no es expresable sólo con $\{\dot\vee\}$.

b) (**10 p.**) Demostrar que toda fórmula de *una* variable proposicional con conectivos $\{\neg, \rightarrow\}$ es expresable con el conjunto de conectivos $\{\leftrightarrow, \dot\vee\}$.

Ejercicio 2. (**15 p.**) Dado una fórmula β fija y Γ un conjunto consistente. Mostrar que si $\Gamma \nvdash \beta$ y $\Gamma \nvdash \neg\beta$, entonces existen Γ_1 y Γ_2 maximales consistentes, tales que $\mathbf{Con}(\Gamma) \subseteq \mathbf{Con}(\Gamma_1)$, $\mathbf{Con}(\Gamma) \subseteq \mathbf{Con}(\Gamma_2)$, $\Gamma_1 \vdash \beta$ y $\Gamma_2 \vdash \neg\beta$.

Ejercicio 3. Sea $\mathcal{L} = \{=, \rightsquigarrow\}$ el lenguaje de primer orden con igualdad y un símbolo de relación binaria (infijo) $\rightsquigarrow$. Leemos $x \rightsquigarrow y$ como "x se reduce en un paso a y". Llamamos "Sistema de Reducción Abstracto' (ARS, por sus siglas en inglés) a cualquier $\mathcal{L}$-estructura.

Sea $\mathrm{suc}(x) = \{e : x \rightsquigarrow e\}$ el conjunto de posibles reducciones directas a partir de x.[1] Decimos que un ARS tiene *branching finito* cuando para todo elemento x, $\mathrm{suc}(x)$ es finito.

Ejemplos: El ARS $\mathcal{A}_1 = (\mathbb{N}, \overset{\mathcal{A}_1}{\rightsquigarrow})$ donde $a \overset{\mathcal{A}_1}{\rightsquigarrow} b$ sii $b \leq a$, *tiene* branching finito, ya que cada elemento tiene un solo finitos elementos menores o iguales (por más que haya 'rulos'). En cambio el ARS $\mathcal{A}_2 = (\mathbb{N}, \overset{\mathcal{A}_2}{\rightsquigarrow})$ donde $\overset{\mathcal{A}_2}{\rightsquigarrow} = \{(0, i) : i \geq 1\}$, *no* tiene branching finito ya que $\mathrm{suc}(0) = \mathbb{N}^+$.

a) (**10 p.**) Demostrar que, para todo $n > 0$, existe una $\mathcal{L}$-fórmula $\psi_n(x)$ con sólo una variable libre x tal que, para todo ARS $\mathcal{A}$ y toda valuación v, $\mathcal{A} \models \psi_n[v]$ sii $|\mathrm{suc}(v(x))| \geq n$.

b) (**25 p.**) Demostrar que no es posible dar una $\mathcal{L}$-fórmula φ tal que para todo ARS $\mathcal{A}$ y valuación v, $\mathcal{A} \models \varphi[v]$ sii $\mathcal{A}$ tiene branching finito.

Ejercicio 4. Sea Γ un conjunto *consistente* de sentencias de primer orden. Decimos que la teoría Γ es "completa respecto a la negación" si para toda *sentencia* φ, $\Gamma \vdash \varphi$ ó $\Gamma \vdash \neg\varphi$. Decidir en cada caso si la afirmación es verdadera ó falsa. Demostrar *detalladamente* o proveer un contraejemplo.

a) (**10 p.**) Si Γ es completa respecto a un modelo $\mathcal{M}$, Γ es completa respecto a la negación (recordar que Γ es completa respecto a $\mathcal{M}$ sii Γ es completa respecto a la clase que sólo contiene a $\mathcal{M}$).

b) (**15 p.**) Sean Σ, Δ conjuntos consistentes tal que Δ es completo respecto a la negación y $\Sigma \models \Delta$ entonces para toda sentencia φ, $\Sigma \models \varphi$ sii $\Delta \models \varphi$.

[1] Notar que '$\mathrm{suc}(x)$' *no* está en el lenguaje y *no* es una fórmula.