El sistema de Hilbert: Lógica de Primer Orden

El sistema de deducción de Hilbert consta de los siguientes elementos:

- Esquemas para generar fórmulas válidas:

  (a) $\varphi \to (\psi \to \varphi)$.

  (b) $(\varphi \to (\psi \to \theta)) \to ((\varphi \to \psi) \to (\varphi \to \theta))$.

  (c) $(\neg\varphi \to \neg\psi) \to ((\neg\varphi \to \psi) \to \varphi)$.

  (d) $(\forall x\, \varphi(x)) \to \varphi(t)$, donde t es un término cualquiera que no menciona a ninguna variable en φ.

  (e) $\varphi(t) \to (\exists x\, \varphi(x))$, donde t es un término cualquiera y x no es mencionado en φ.

  (f) $(\exists x\, \varphi) \leftrightarrow (\neg\forall x\, \neg\varphi)$.

El sistema de Hilbert: Lógica de Primer Orden

- Axiomas para la igualdad:

  (a) $\forall x\,(x = x)$.

  (b) $\forall x \forall y\,(x = y \rightarrow y = x)$.

  (c) $\forall x \forall y \forall z\,((x = y \wedge y = z) \rightarrow x = z)$.

  (d) Para todo predicado m-ario P:

$$\forall x_1 \cdots \forall x_m \forall y_1 \cdots \forall y_m\,((P(x_1, \ldots, x_m) \wedge x_1 = y_1 \wedge \cdots \wedge x_m = y_m) \rightarrow P(y_1, \ldots, y_m)).$$

  (e) Para toda función n-aria f:

$$\forall x_1 \cdots \forall x_n \forall y_1 \cdots \forall y_n\,((x_1 = y_1 \wedge \cdots \wedge x_n = y_n) \rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)).$$

El sistema de Hilbert: Lógica de Primer Orden

- Reglas de inferencia:

  (a) Modus Ponens:

$$\frac{\varphi \to \psi \qquad \varphi}{\psi}$$

  (b) Generalización: Si y no aparece libre en φ, entonces

$$\frac{\varphi \to \psi(y)}{\varphi \to \forall x \psi(x)}$$

El sistema de Hilbert: Lógica de Primer Orden

Dado un conjunto de fórmulas $\Sigma \cup \{\varphi\}$, una deducción formal de φ desde Σ es una secuencia de fórmulas $\varphi_1, \varphi_2, \ldots, \varphi_n$ tal que:

- Para cada $i \leq n$:
  - $\varphi_i \in \Sigma$ o
  - φ_i es un axioma lógico o
  - existen $j, k < i$ tales que φ_i es obtenido desde φ_j y φ_k usando modus ponens o
  - existe $j < i$ tal que φ_i es obtenido desde φ_j usando la regla de generalización.
- $\varphi_n = \varphi$.

Notación: $\Sigma \vdash_H \varphi$.

El sistema de Hilbert: Lógica de Primer Orden

Ejemplo: Demostraremos que $\{\forall x\,\phi,\ \forall x\,(\phi \to \psi)\} \vdash_H \forall x\,\psi$ (asuma que y es una variable no mencionada en ninguna de estas fórmulas):

(1)	$\forall x\,\phi \to \phi(y)$	por esquema (d)
(2)	$\forall x\,(\phi \to \psi) \to (\phi(y) \to \psi(y))$	por lo mismo
(3)	$\forall x\,\phi$	pertenece a Σ
(4)	$\forall x\,(\phi \to \psi)$	pertenece a Σ
(5)	$\phi(y)$	modus ponens de (1) y (3)
(6)	$\phi(y) \to \psi(y)$	modus ponens de (2) y (4)
(7)	$\psi(y)$	modus ponens de (5) y (6)
(7)	$\forall x\,\psi(x)$	por regla de generalización aplicada a (7)

El sistema de Hilbert: Propiedades

Teorema (corrección): Dado un conjunto de fórmulas $\Sigma \cup \{\varphi\}$, si $\Sigma \vdash_H \varphi$, entonces $\Sigma \models \varphi$.

Ejercicio: Demuestre el teorema.

Teorema (completidad de Gödel): Dado un conjunto de fórmulas $\Sigma \cup \{\varphi\}$, si $\Sigma \models \varphi$, entonces $\Sigma \vdash_H \varphi$.

Corolario (compacidad): Un conjunto de fórmulas Σ es satisfacible si y sólo si Σ es finitamente satisfacible.

Ejercicio: Demuestre el corolario.

Una aplicación del teorema de compacidad

Usaremos el teorema de compacidad para demostrar que el conjunto de los grafos conexos no es definible en la lógica de primer orden.

Es decir, dado $\mathcal{L} = \{E(x, y)\}$, queremos usar compacidad para demostrar que no existe $\mathcal{L}$-oración α tal que para todo grafo G,

$$G \models \alpha \quad \Leftrightarrow \quad G \text{ es conexo.}$$

Asuma por contradicción que tal $\mathcal{L}$-fórmula α existe.

Sea $\mathcal{L}'$ la expansión de $\mathcal{L}$ con dos constantes c_1 y c_2, y considere $\mathcal{L}'$-formulas

$$\phi_n \equiv \neg\big(\exists x_1 \ldots \exists x_n (E(c_1, x_1) \wedge E(x_1, x_2) \wedge \ldots \wedge E(x_n, c_2))\big).$$

¿Cuándo una $\mathcal{L}'$-estructura satisface ϕ_n?

Una aplicación del teorema de compacidad

Defina un conjunto Σ de $\mathcal{L}'$-oraciones como:

$$\{\alpha\} \cup \{\phi_n \mid n \geq 0\} \cup \{\neg(c_1 = c_2)\}.$$

Por teorema de compacidad, Σ es satisfacible por una $\mathcal{L}'$-estructura $\mathfrak{A}$.

Contradicción: $\mathfrak{A}$ es un grafo conexo y no existe camino de largo n entre c_1 y c_2, para ningún $n \geq 0$.

Ejercicio: Demuestre usando compacidad que no es posible expresar en lógica de primer orden que un grafo contiene un ciclo de largo finito.

Una forma normal para la lógica de primer orden

A veces para demostrar una propiedad de la lógica de primer orden es útil asumir que las fórmulas están en la siguiente forma normal:

Una fórmula está en forma normal prenex (FNP) si es de la forma $Q_1 x_1 \ldots Q_n x_n \alpha$, donde cada Q_i es $\forall$ o $\exists$, y α no contiene cuantificadores.

Ejemplo: La fórmula $\forall x(A(x) \rightarrow \forall y B(x,y))$ no está en FNP, pero la fórmula $\forall x \forall y(A(x) \rightarrow B(x,y))$ sí lo está. Sin embargo, ambas fórmulas son equivalentes.

Una forma normal para la lógica de primer orden

Teorema: Toda fórmula es equivalente a una fórmula en FNP.

Para demostrarlo es suficiente usar inducción y las siguientes equivalencias:

$$
\begin{array}{rcll}
\neg\forall x\alpha & \equiv & \exists x\neg\alpha & \\
\neg\exists x\alpha & \equiv & \forall x\neg\alpha & \\
\alpha \rightarrow \forall x\beta & \equiv & \forall x(\alpha \rightarrow \beta) & \text{si } x \text{ no está libre en } \alpha. \\
\alpha \rightarrow \exists x\beta & \equiv & \exists x(\alpha \rightarrow \beta) & \text{si } x \text{ no está libre en } \alpha. \\
\forall x\alpha \rightarrow \beta & \equiv & \exists x(\alpha \rightarrow \beta) & \text{si } x \text{ no está libre en } \beta. \\
\exists x\alpha \rightarrow \beta & \equiv & \forall x(\alpha \rightarrow \beta) & \text{si } x \text{ no está libre en } \beta.
\end{array}
$$