Tema 5: Lógica primer orden

Lógica de predicados (LP)

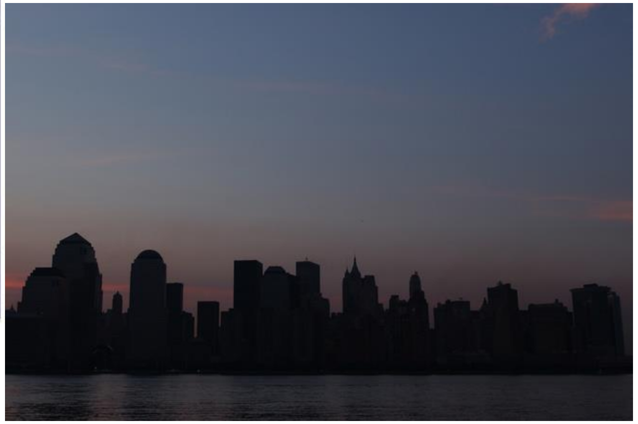
Rosa María Maza Quiroga
Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga

Contenido

1. Introducción
2. Sintaxis y Semántica
3. Utilización
4. Conclusiones

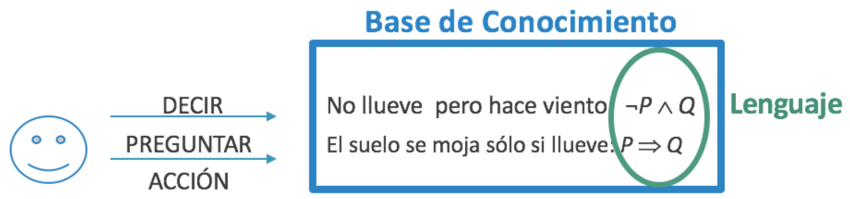
1. Introducción

Motivación



1. Introducción: Motivación I

- Tema anterior el modelo **Lógica proposicional**:
  - Sintaxis: fórmulas bien formadas.
  - Semántica: reglas para determinar verdad de una sentencia respecto a un modelo.
  - Base de Conocimiento: conjunto de sentencias, incluyen reglas y hechos que el agente basado en conocimiento conoce.
  - Regla de inferencia: vimos la Regla de Resolución (correcta y completa).
  - Algoritmo de Resolución: conversión CNF, aplicar regla de resolución repetidamente y se obtiene el resultado (se infiere o no una sentencia de una BC).



➢ Demasiado sencilla para representar el conocimiento de entornos complejos.

1. Introducción: Motivación II

- Tema presente la **Lógica primer orden**:
  - Toma ideas del lenguaje natural:
    - Se construye sobre objetos y las relaciones entre ellos
    - ❖ Ej. objetos: números, colores, personas.
    - ❖ EJ. relaciones: es mayor que, es más oscuro que, es hermano de.

    - Supone que dichas relaciones o se cumplen o no se cumplen.
    - ❖ Ej. relación: El negro <u>es más oscuro que</u> el blanco. Esto se cumple.

    - Considera la posibilidad de objetos genéricos: variables.
    - ❖ Ej. variable color = {negro, blanco, rojo, amarillo...}
    - ❖ Ej. relación que tienen todos los objetos: todos los colores son bonitos.

  - Evita ambigüedades del lenguaje natural.

2. Sintaxis y Semántica

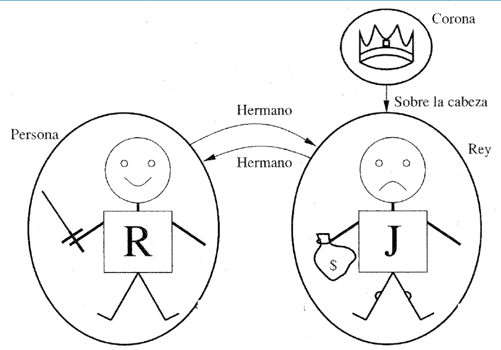
Modelos

Símbolos e interpretaciones

2. Sintaxis y Semántica. Modelo

- Un modelo es un mundo posible que podría considerarse:

LÓGICA PROPOSICIONAL	LÓGICA DE PRIMER ORDEN
BC: *Sol* ⇔ (*Despejado*)	
Modelo: *m* = {*Sol*= Verdad, *Despejado* = Verdad}	

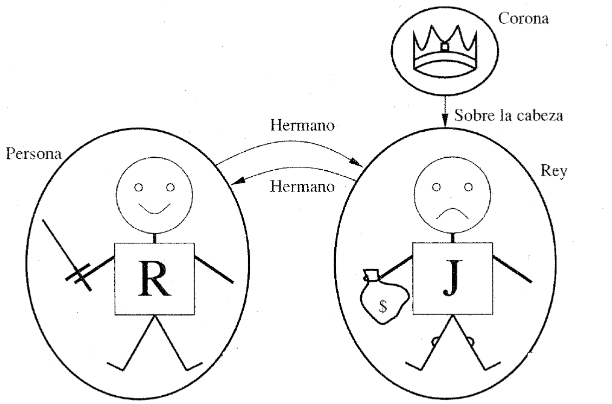


- Componentes del modelo:
  - **Dominio**: conjunto de objetos que contiene.
  - Un conjunto de **relaciones** entre objetos.
    - Relación: conjunto de tuplas (lista ordenada de elementos) de objetos que están relacionados.

2. Sintaxis y Semántica. Modelo

❖ Ejemplo de un modelo:

- Dominio: El rey Ricardo Corazón de León; su hermano el rey Juan y una corona. Total: 3 objetos.
- Relaciones:
  - Hermandad: {<Ricardo Corazón de León, Rey Juan>, <Rey Juan , Ricardo Corazón de León >}
  - Sobre la cabeza: {<La corona, Rey Juan>}



2. Sintaxis y Semántica. Símbolos e interpretaciones

- Elementos sintácticos:
  - Símbolos de **constante**: representan objetos.
  - Símbolos de **predicado**: representan relaciones.
- Todos los símbolos empiezan por mayúscula y se escriben en cursiva.
- Cada símbolo de predicado tiene su **aridad** que fija el número de argumentos.
- Cada modelo incluye una **interpretación** que dice qué objetos y relaciones se corresponden con los símbolos de constante y predicado.
- Representaremos a las variables por cualquier cadena alfanumérica que empiece por minúscula.

2. Sintaxis y Semántica. Símbolos e interpretaciones

- Ejemplo de símbolos e interpretaciones:

- Símbolos de constante (objetos): *Ricardo*, *Juan, Corona.*
- Símbolos de predicado (relaciones): *Hermano, SobreCabeza, Persona, Rey, EsCorona.*
- Una posible interpretación (entre otras muchas):
  - *Ricardo* se refiere a Ricardo Corazón de León y *Juan* se refiere al malvado rey Juan. *Corona* se refiere a la corona de oro que estaba en esa época para representar al rey oficial.
  - *Hermano* se refiere a la relación de hermandad, *SobreCabeza* se refiere a la relación sobre la cabeza, y *Persona*, *Rey* y *EsCorona* se refieren a los conjuntos de objetos que son personas, reyes y coronas, respectivamente.

2. Sintaxis y Semántica. Símbolos e interpretaciones

5 objetos

2 relaciones binarias:

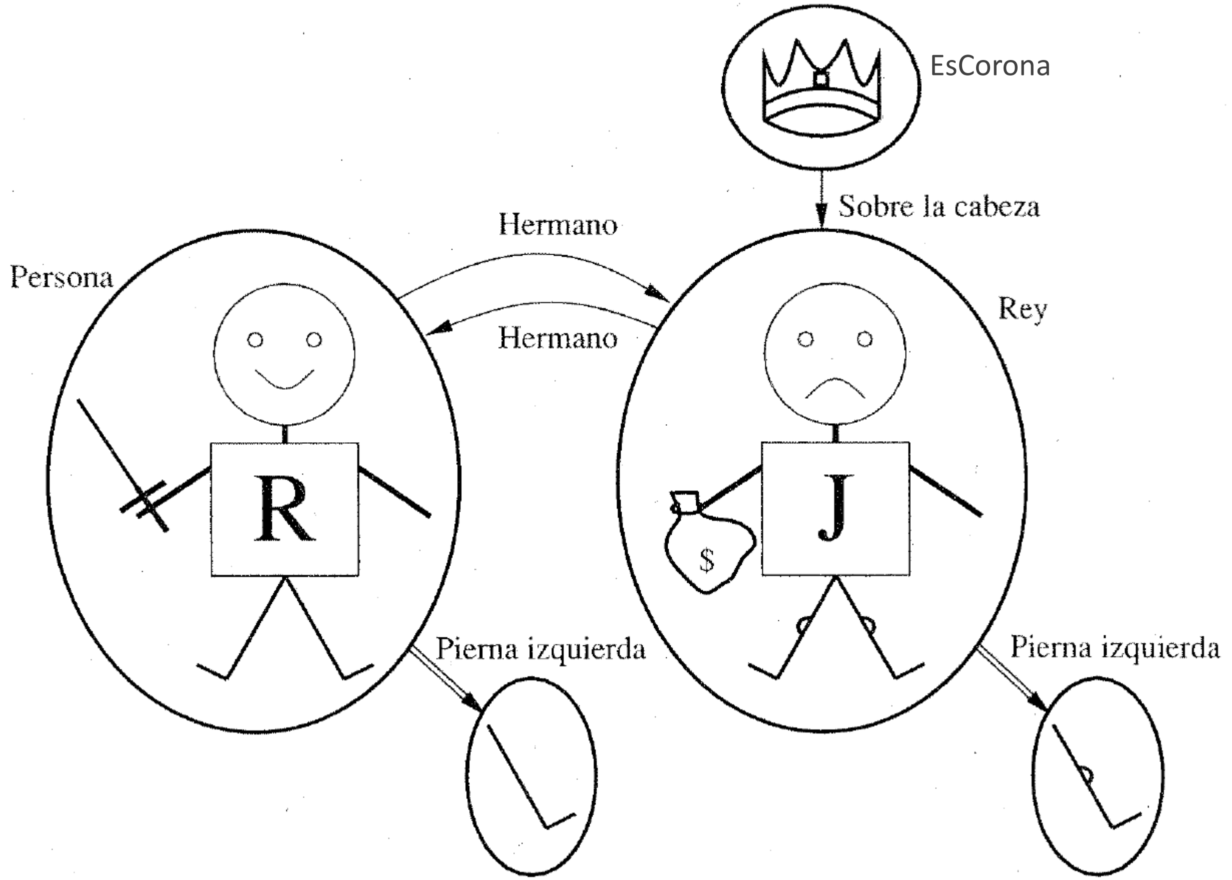
Hermano, SobreCabeza

3 relaciones unitarias:

Persona, EsCorona, Rey

1 función unitaria:

PiernaIzquierda



2. Sintaxis y Semántica. Símbolos e interpretaciones

LÓGICA PROPOSICIONAL

- Una sentencia puede ser verdadera o falsa.
- La interpretación asigna valores de verdad a los átomos.
- El valor de verdad de las sentencias compuestas se compone a partir de los valores de verdad de los átomos, siguiendo las reglas semánticas de las conectivas lógicas (tablas de verdad).

LÓGICA DE PRIMER ORDEN

- Se establece un dominio.
- Se determina una interpretación que establece una correspondencia entre:
  - Las constantes y los objetos del dominio.
  - Los predicados y las relaciones entre dichos objetos.

2. Sintaxis y Semántica.
Términos y sentencias atómicas y compuestas

- Cada **término** se refiere a un objeto del dominio, y su semántica viene determinada por la interpretación.

❖ Ej. *Ricardo*, *PiernaIzquierda*(*Juan*)

- Una **expresión atómica** es verdadera si la relación a la que hace referencia el predicado, es cierta entre los objetos del dominio a los que hacen referencia sus argumentos.

❖ *Hermano*(*Ricardo*, *Juan*)

- El valor de verdad de las **expresiones compuestas** mediante conectivas lógicas, se compone a partir de los valores de las expresiones atómicas (como en lógica proposicional).

2. Sintaxis y Semántica. Cuantificadores

- Los cuantificadores nos permiten expresar propiedades de colecciones de objetos:

  - Cuantificador universal ∀ (para todo): va seguido de una o más variables en minúscula.

    $\forall$ *x P: P* es verdadero para todo objeto x

  - Cuantificador existencial ∃ (existe): va seguido de una o más variables en minúscula.

    $\exists$ *x P: P* es verdadero para al menos un objeto x

2. Sintaxis y Semántica. Cuantificadores

- Se pueden anidar y el orden es muy importante:
  - Ej. Todo el mundo ama a alguien: $\forall x\ \exists y\ Amar(x,y)$
  - Ej. Existe alguien que es amado por todo el mundo: $\exists y\ \forall x\ Amar(x,y)$
- Reglas de Morgan para fórmulas cuantificadas:

$$\boxed{\forall\, x\, P \equiv \neg\exists\, x \neg\, P}$$

- *Ej.* $\forall\, x\ Gusta(x,\ Helado)$ es equivalente a $\neg\exists\, x \neg\ Gusta(x,\ Helado)$

Se leería: ‘A todos les gusta el helado, es lo mismo que no existe alguien a quien no le guste.’

$$\boxed{\exists\, x\, P \equiv \neg\, \forall\, x \neg\, P}$$

- *Ej.* $\exists\, x\ Matrícula(x)$ es equivalente a $\neg\, \forall\, x \neg\ Matrícula(x)$

Se leería: ‘Hay alguien que sacará matrícula, es lo mismo que no todos no sacaron matrícula.’

☞ Ejercicio 1. Uso de cuantificadores.

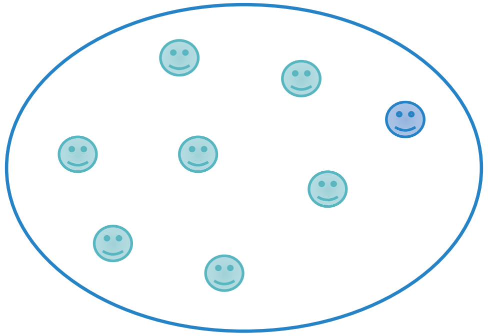
Ejercicio 1. Enunciado y ejemplos

Ejercicio 1 (uso de los cuantificadores):

Traduce estas fórmulas del lenguaje natural al lenguaje de la lógica de primer orden. Usa el predicado *Loves*, donde *Loves*(x,y) quiere decir "x ama a y".

a) "Hay alguien que ama a todo el mundo".
b) "Hay alguien que ama a al menos una persona".
c) "Hay alguien que ama a algún otro".
d) "Todos se aman mutuamente".
e) "Hay alguien que es amado por todos".
f) "Hay alguien a quien todos aman".
g) "Todo el mundo tiene a alguien que lo ama".

- Aclaración con ejemplos:



a) **Hay alguien que ama a todo el mundo**: $\exists x\ \forall y\ Amar(x,y)$.

g) **Todo el mundo tiene a alguien que lo ama**: $\forall x\ \exists y\ Amar(y,x)$

Equivalente a: **todo el mundo es amado por alguien**.

Explicación: todos los objetos persona tienen al menos a otro objeto persona que lo ama.

Diapositiva cuantificadores : **Existe alguien que es amado por todo el mundo**: $\exists y\ \forall x\ Amar(x,y)$.

Explicación: existe un objeto persona al quien todos los objetos persona quieren.

Hacemos un paréntesis:

Propósito: las equivalencias en el lenguaje natural (decir lo mismo de dos formas diferentes) dan lugar a equivalencias en Lógica de Primer Orden (dos sentencias que serán equivalentes).

Vamos a ver algunos ejemplos para entender cómo son las equivalencias y su demostración.

1. Equivalencias en el lenguaje natural: $\forall\, x\, P \equiv \neg\exists\, x \neg P$

❖ **Formaliza a Lógica de Primer Orden: Todos los andaluces son españoles.**

$\forall\, x\, (\mathit{Andaluz}(x) \Rightarrow \mathit{Español}(x))$

❖ **¿Cómo se puede decir esta oración en lenguaje natural de otra forma?**

No existe un andaluz que no sea español.

❖ **Formaliza a Lógica de Primer Orden la frase del apartado anterior.**

$\neg\exists\, x\, (\mathit{Andaluz}(x) \land \neg\, \mathit{Español}\,(x))$

❖ **Observa la equivalencia entre las dos formalizaciones en Lógica de Primer Orden.**

Al decir lo mismo las dos oraciones, las dos sentencias son equivalentes.

$\forall\, x\, (\mathit{Andaluz}(x) \Rightarrow \mathit{Español}(x)) \equiv \neg\exists\, x\, (\mathit{Andaluz}(x) \land \neg\, \mathit{Español}\,(x))$

❖ **Señala las diferencias en la equivalencia.**

$\forall\, x\, (\mathit{Andaluz}(x) \Rightarrow \mathit{Español}(x)) \equiv \neg\exists\, x\, (\mathit{Andaluz}(x) \land \neg\, \mathit{Español}\,(x))$

1. Explicación teórica de cómo se llega a la equivalencia: $\forall x\, P \equiv \neg\exists x \neg P$

❖ **Ejercicio 1.1. Dado el siguiente enunciado, formalízalo a Lógica de Primer Orden y encuentra la equivalencia usando las Leyes de Morgan con cuantificadores: Todos los andaluces son españoles.**

$\forall x\, (Andaluz(x) \Rightarrow Español(x))$

Aplicamos la primera Ley de Morgan para cuantificadores que vimos en clase $\boxed{\forall x\, P \equiv \neg\exists x \neg P}$

$\forall x\, (Andaluz(x) \Rightarrow Español\,(x)) \equiv \{\forall x\, P \equiv \neg\exists x \neg P\} \equiv \neg\exists x \neg (Andaluz(x) \Rightarrow Español\,(x))$

❖ **Ejercicio 1.2. Hecho el apartado anterior, encontrar la sentencia disyuntiva equivalente:**

Entendido esto, vamos a hacer unas equivalencias más:

Probemos a aplicar la equivalencia $\alpha \Rightarrow \beta$ por $\neg\alpha \vee \beta$:

$\neg\exists x \neg (Andaluz(x) \Rightarrow Español\,(x)) \equiv \neg\exists x \neg (\neg Andaluz\,(x) \vee Español\,(x))$

Probemos a mover hacia afuera la negación usando la Ley de Morgan $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$:

$\neg\exists x\, (Andaluz(x) \wedge \neg Español\,(x))$

$\boxed{\text{Conclusión: } \forall x\, (Andaluz(x) \Rightarrow Español\,(x)) \equiv \neg\exists x\, (Andaluz(x) \wedge \neg Español\,(x))}$ Muy interesante ver la equivalencia entre estas sentencias.

Todos los andaluces son españoles ≡ No existe andaluz que no sea español

1. Ejemplo I

- Ej. Dado el siguiente enunciado, formalízalo a Lógica de Primer Orden y encuentra la equivalencia usando las Leyes de Morgan (con cuantificador): **Todos los que viven en el pueblo son extranjeros.**

$\forall\, x$ (Vive(x) $\Rightarrow$ Extranjero(x))

Aplicamos la primera Ley de Morgan para cuantificadores que vimos en clase $\boxed{\forall\, x\, P \equiv \neg\exists\, x\, \neg\, P}$

$\forall\, x$ (Vive(x) $\Rightarrow$ Extranjero(x)) ≡ {$\forall\, x\, P \equiv \neg\exists\, x\, \neg\, P$ } ≡ $\neg\exists\, x\, \neg$ (Vive(x) $\Rightarrow$ Extranjero(x))

- Entendido esto, vamos a hacer unas equivalencias más:

  Probemos a aplicar la equivalencia $\alpha \Rightarrow \beta$ por $\neg\alpha \vee \beta$:

  $\neg\exists\, x\, \neg$ (Vive(x) $\Rightarrow$ Extranjero(x)) ≡ $\neg\exists\, x\, \neg$ ($\neg$ Vive(x) $\vee$ Extranjero(x))

  Probemos a mover hacia afuera la negación usando la Ley de Morgan $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$:

  $\neg\exists\, x$ (Vive(x) $\wedge$ $\neg$ Extranjero(x))

  Conclusión: $\forall\, x$ (Vive(x) $\Rightarrow$ Extranjero(x)) ≡ $\neg\exists\, x$ (Vive(x) $\wedge$ $\neg$ Extranjero(x)) Muy interesante ver la equivalencia entre estas sentencias.

  Todos los que viven en el pueblo son extranjeros ≡ No existe alguien que viva en el pueblo y no sea extranjero

1. Ejemplo II

- Ej. Dado el siguiente enunciado, formalízalo a Lógica de Primer Orden y encuentra la equivalencia usando las Leyes de Morgan (con cuantificador): **Todos los que viven en el pueblo no son extranjeros.**

$\forall x$ (Vive(x) $\Rightarrow \neg$ Extranjero(x))

Aplicamos la primera Ley de Morgan para cuantificadores que vimos en clase $\boxed{\forall x\, P \equiv \neg\exists x \neg P}$

$\forall x$ (Vive(x) $\Rightarrow \neg$ Extranjero(x)) ≡ {$\forall x\, P \equiv \neg\exists x \neg P$ } ≡ $\neg\exists x \neg$ (Vive(x) $\Rightarrow \neg$ Extranjero(x))

- Entendido esto, vamos a hacer unas equivalencias más:

  Probemos a aplicar la equivalencia $\alpha \Rightarrow \beta$ por $\neg\alpha \vee \beta$:

  $\neg\exists x \neg$ (Vive(x) $\Rightarrow \neg$ Extranjero(x)) ≡ $\neg\exists x \neg$ ($\neg$ Vive(x) $\vee \neg$ Extranjero(x))

  Probemos a mover hacia afuera la negación usando la Ley de Morgan $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$:

  $\neg\exists x$ (Vive(x) $\wedge$ Extranjero(x))

  Conclusión: $\forall x$ (Vive(x) $\Rightarrow \neg$ Extranjero(x)) ≡ $\neg\exists x$ (Vive(x) $\wedge$ Extranjero(x)) Muy interesante ver la equivalencia entre estas sentencias.

  Todos los que viven en el pueblo no son extranjeros ≡ No existe alguien que viva en el pueblo y sea extranjero

2. Equivalencias en el lenguaje natural: $\exists\, x\, P \equiv \neg \forall\, x \neg P$

❖ **Formaliza a Lógica de Primer Orden: Hay un superhéroe que es verde.**

$\exists\, x\, (Superhéroe(x) \wedge Verde(x))$

❖ **¿Cómo se puede decir esta oración en lenguaje natural de otra forma?**

No todos los superhéroes ‘no’ son verdes (**si** considero a todos los superhéroes, no todos serán verdes).

❖ **Formaliza a Lógica de Primer Orden la frase del apartado anterior.**

$\neg \forall\, x\, (Superhéroe(x) \Rightarrow \neg Verde(x))$

❖ **Observa la equivalencia entre las dos formalizaciones en Lógica de Primer Orden.**

Al decir lo mismo las dos oraciones, las dos sentencias son equivalentes.

$\exists\, x\, (Superhéroe(x) \wedge Verde(x)) \equiv \neg \forall\, x\, (Superhéroe(x) \Rightarrow \neg Verde(x))$

❖ **Señala las diferencias en la equivalencia.**

$\exists\, x\, (Superhéroe(x) \wedge Verde(x)) \equiv \neg \forall\, x\, (Superhéroe(x) \Rightarrow \neg Verde(x))$

2. Explicación teórica de cómo se llega a la equivalencia: $\exists\, x\, P \equiv \neg \forall\, x \neg P$

- **Ejercicio 1.1. Dado el siguiente enunciado, formalízalo a Lógica de Primer Orden y encuentra la equivalencia usando las Leyes de Morgan con cuantificadores: Hay un superhéroe que es verde.**

$\exists\, x\, (\mathit{Superhéroe}(x) \wedge \mathit{Verde}(x))$

Aplicamos la primera Ley de Morgan para cuantificadores que vimos en clase $\boxed{\exists\, x\, P \equiv \neg \forall\, x \neg P}$

$\exists\, x\, (\mathit{Superhéroe}(x) \wedge \mathit{Verde}(x)) \equiv \{\exists\, x\, P \equiv \neg \forall\, x \neg P\} \equiv \neg \forall\, x \neg (\mathit{Superhéroe}(x) \wedge \mathit{Verde}(x))$

- **Ejercicio 1.2. Hecho el apartado anterior, encontrar la sentencia disyuntiva equivalente:**

Entendido esto, vamos a hacer unas equivalencias más:

Probemos a mover hacia dentro la negación usando la Ley de Morgan $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$:

$\neg \forall\, x \neg (\mathit{Superhéroe}(x) \wedge \mathit{Verde}(x)) \equiv \neg \forall\, x\, (\neg \mathit{Superhéroe}(x) \vee \neg \mathit{Verde}(x))$

Probemos a aplicar la equivalencia $\alpha \Rightarrow \beta$ por $\neg\alpha \vee \beta$:

$\neg \forall\, x\, (\mathit{Superhéroe}(x) \Rightarrow \neg \mathit{Verde}(x))$

$\boxed{\text{Conclusión: } \exists\, x\, (\mathit{Superhéroe}(x) \wedge \mathit{Verde}(x)) \equiv \neg \forall\, x\, (\mathit{Superhéroe}(x) \Rightarrow \neg \mathit{Verde}(x))}$ Muy interesante ver la equivalencia entre estas sentencias.

Hay un superhéroe que es verde ≡ No todos los superhéroes ‘no’ son verdes (si considero a todos los superhéroes, no todos serán verdes).

Cerramos el paréntesis:

Seguimos con la teoría

2. Sintaxis y Semántica.

- El símbolo **igualdad** = se utiliza para indicar que dos términos se refieren al mismo objeto.
  - Ej. El padre de Juan y Henry son el mismo objeto: *Padre*(*Juan*) = *Henry*
  - Ej. Ricardo tiene al menos dos hermanos: ∃x, y *Hermano*(x, *Ricardo*) ∧ *Hermano*(y, *Ricardo*) ∧ ¬ (x,y)
- Se pueden usar conectivas lógicas para formar **fórmulas compuestas** a partir de los átomos, con la misma sintaxis y semántica que en la lógica proposicional.
  - Ej. *Hermano*(*Juan*, *Ricardo*) ∧ *Hermano*(*Ricardo*, *Juan*)
  - Ej. *Rey*(*Ricardo*) ∨ *Rey*(*Juan*)
  - Ej. ¬ *Rey*(*Ricardo*) ⇒ *Rey*(*Juan*)

2. Sintaxis y Semántica. Ejemplos

- Ej. Todos los reyes son personas:

  $\forall x\ Rey(x) \Rightarrow Persona(x)$ (correcto) [**implicación cuantificada universalmente**]

  $\forall x\ Rey(x) \wedge Persona(x)$ (incorrecto)

La segunda expresión sería ”Todos los objetos son reyes y personas”. Esto es demasiado ‘estricto’.

- Ej. El rey Juan tiene una corona sobre su cabeza:

  $\exists\ x\ Corona(x) \wedge SobreCabeza(x, Juan)$ (correcto)

  $\exists\ x\ Corona(x) \Rightarrow SobreCabeza(x, Juan)$ (incorrecto) [**implicación cuantificada existencialmente**]

La segunda expresión sería ”Existe un objeto que o no es una corona o está sobre la cabeza del rey Juan’. Esto es demasiado ‘débil’ y no sirve de nada.

☞ Ejercicio 6. Lectura de fórmulas de primer orden.

Ejercicio 6. Enunciado y ejemplos

Ejercicio 6 (lectura de fórmulas de primer orden):

Traduce al español las siguientes fórmulas de la lógica de primer orden, y determina cuáles de ellas representan afirmaciones verdaderas cuando se interpretan en el conjunto de los números reales, **R**.

a) $\neg \forall x\, x \neq 0$

b) $\exists\, x\, x = x^2 \Rightarrow x < 0$

c) $\forall x\ \forall y\, x > y \vee y > x$

d) $\forall x\ \exists y\, x > y \Rightarrow x > y^2$

e) $\exists x\ \forall y\, x + y = x$

f) $\exists x\ \forall \mathrm{y}\, x + \mathrm{y} = \mathrm{y}$

g) $\exists x\ \forall y\, x > y \vee -x > y$

h) $\exists x \forall y\, x > y \vee \neg(x > y)$

i) $\exists x \forall y\, y > x \Rightarrow y^2 > x$

j) $\exists x \forall y\, x > y \Rightarrow x > y^2$

k) $\forall x\ \exists y\ \forall z\, xy = yz$

l) $\exists x\ \forall y\ \exists z\, (x + y)z = 1$

m) $\forall x\ \forall y\, x > y \Rightarrow\ \exists z\, (x > z \wedge z > y)$

n) $\forall x\ \exists z\ \forall y\, x > z \Rightarrow z > y$

Ejercicio 6. Implicación cuantificada existencialmente

b) $\exists x \quad x = x^2 \Rightarrow x < 0$
<u>Significado</u>: existe un número real tal que si ese número es igual a su cuadrado, entonces es negativo.
<u>Implicación</u>: si la premisa es falsa, da igual como sea la conclusión, la sentencia es verdadera.
<u>Explicación</u>: con que exista un sólo número real que sea diferente a su cuadrado, ya es un objeto que cumple esta sentencia y la hace verdadera en el dominio de los números reales (no dice mucho de los números reales).
Caso: $2 \neq 2^2$ por tanto esta sentencia es verdadera para los números reales.

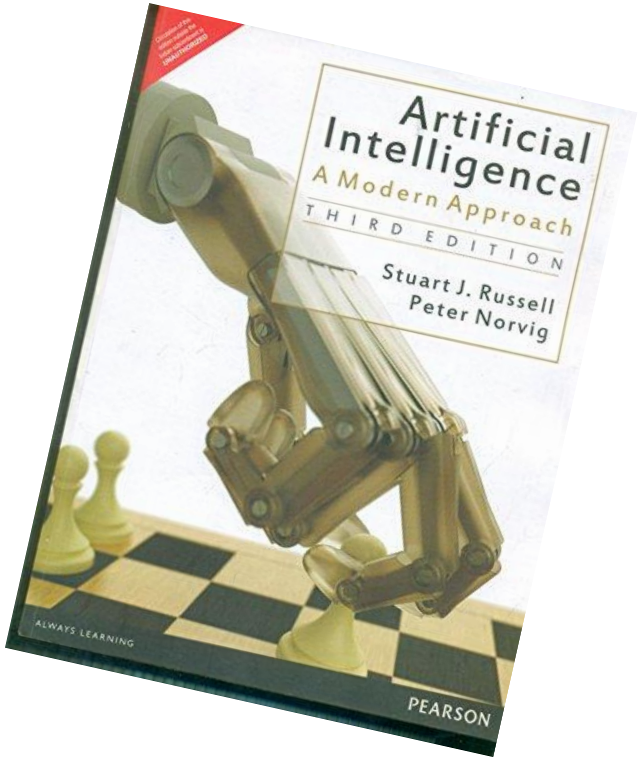
d) $\forall x\ \exists y\ x > y \Rightarrow x > y2$
<u>Significado</u>: para todo real *x* existe un *y* tal que si *x* es mayor que *y* también es mayor al cuadrado de *y*.
<u>Explicación</u>: *x* considera a todos los números, entonces *x* puede ser *y* (no da información relevante).
<u>Caso</u>: en ese caso (*x*=*y*) la premisa es falsa y la sentencia es verdadera.

Conclusión: truco, con un solo caso que neguemos la premisa la sentencia es afirmativa.

Ejercicio 8.6. Representar en LP. Enunciado

8.6 Represente las siguientes sentencias en lógica de primer orden, utilizando un vocabulario consistente (que usted debe definir):

a) Algunos estudiantes estudian francés en la primavera de 2001.
b) Cada estudiante que estudia francés lo aprueba.
c) Sólo un estudiante estudia griego en la primavera de 2001.
d) La mejor puntuación en griego siempre es mayor que la mejor puntuación en francés.
e) Todas las personas que compran una póliza son inteligentes.
f) Nadie compra una póliza cara.
g) Hay un agente que vende pólizas sólo a la gente que no está asegurada.
h) Hay un barbero que afeita a todos los hombres de la ciudad que no se afeitan ellos mismos.
i) Una persona nacida en Reino Unido, cuyos padres sean ciudadanos de Reino Unido o residentes en Reino Unido, es un ciudadano de Reino Unido.
j) Una persona nacida fuera de Reino Unido, que tenga uno de los padres ciudadano de Reino Unido o residente en Reino Unido, es ciudadano de Reino Unido por ascendencia.
k) Los políticos pueden mentir a algunos todo el tiempo, y pueden mentir a todos algún tiempo, pero no pueden mentir a todos todo el tiempo.



Ejercicio 8.6. Representar en LP. Solución

a) Algunos estudiantes cursaron francés en la primavera de 2001.

Cursar(*x, a, s*)*:* Estudiante *x* cursa asignatura *a* en el sementre *s.* Relación ternaria (aridad 3).

Ejercicio 8.6. Representar en LP. Solución

a) Algunos estudiantes cursaron francés en la primavera de 2001.

Cursar(*x, a, s*)*:* Estudiante *x* cursa asignatura *a* en el sementre *s.* Relación ternaria (aridad 3).

∃ *x Estudiante*(*x*) ∧ *Cursar*(*x, Francés, Primavera2001*)

Ejercicio 8.6. Representar en LP. Solución

a) Algunos estudiantes cursaron francés en la primavera de 2001.

Cursar(*x, a, s):* Estudiante *x* cursa asignatura *a* en el sementre *s.* Relación ternaria (aridad 3).

∃ *x Estudiante*(*x*) ∧ *Cursar*(*x, Francés, Primavera2001)*

b) Cada estudiante que estudia francés lo aprueba.

∀ *x, s Estudiante*(*x*) ∧ *Cursar*(*x*, *Francés*, *s*) ⇒ *Aprobar(x*, *Francés*, *s*)

Ejercicio 8.6. Representar en LP. Solución

a) Algunos estudiantes cursaron francés en la primavera de 2001.

Cursar(*x, a, s):* Estudiante *x* cursa asignatura *a* en el sementre *s.* Relación ternaria (aridad 3).

∃ *x Estudiante*(*x*) ∧ *Cursar*(*x, Francés, Primavera2001)*

b) Cada estudiante que estudia francés lo aprueba.

∀ *x, s Estudiante*(*x*) ∧ *Cursar*(*x*, *Francés*, *s*) ⇒ *Aprobar(x*, *Francés*, *s*)

c) Sólo un estudiante estudia griego en la primavera de 2001.

Ejercicio 8.6. Representar en LP. Solución

a) Algunos estudiantes cursaron francés en la primavera de 2001.

Cursar(*x, a, s):* Estudiante *x* cursa asignatura *a* en el sementre *s.* Relación ternaria (aridad 3).

∃ *x Estudiante*(*x*) ∧ *Cursar*(*x, Francés, Primavera2001)*

b) Cada estudiante que estudia francés lo aprueba.

∀ *x, s Estudiante*(*x*) ∧ *Cursar*(*x*, *Francés*, *s*) ⇒ *Aprobar(x*, *Francés*, *s*)

c) Sólo un estudiante estudia griego en la primavera de 2001.

∃ x *Estudiante*(*x*) ∧ *Cursar*(*x*, *Griego*, *Primavera2001*) ∧∀*y Estudiante*(*y*) ∧ *y*!=*x* ⇒ ¬ *Cursar*(*y*, *Griego* , *Primavera2001*)

Ejercicio 8.6. Representar en LP. Solución

a) Algunos estudiantes cursaron francés en la primavera de 2001.

Cursar(*x, a, s*)*:* Estudiante *x* cursa asignatura *a* en el sementre *s*. Relación ternaria (aridad 3).

$\exists$ *x Estudiante*(*x*) $\wedge$ *Cursar*(*x, Francés, Primavera2001*)

b) Cada estudiante que estudia francés lo aprueba.

$\forall$ *x, s Estudiante*(*x*) $\wedge$ *Cursar*(*x*, *Francés*, *s*) $\Rightarrow$ *Aprobar*(*x*, *Francés*, *s*)

Recordando orden de precedencia de conectivas:
$\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
Ambas sentencias indican lo mismo.

c) Sólo un estudiante estudia griego en la primavera de 2001.

$\exists$ x *Estudiante*(*x*) $\wedge$ *Cursar*(*x*, *Griego*, *Primavera2001*) $\wedge\forall$*y Estudiante*(*y*) $\wedge$ *y*!=*x* $\Rightarrow \neg$ *Cursar*(*y*, *Griego* , *Primavera2001*)

($\exists$ x *Estudiante*(*x*) $\wedge$ *Cursar*(*x*, *Griego*, *Primavera2001*) $\wedge\forall$*y Estudiante*(*y*) $\wedge$ *y*!=*x* **)** $\Rightarrow \neg$ *Cursar*(*y*, *Griego* , *Primavera2001*)

Ejercicio 8.6. Representar en LP. Solución

a) Algunos estudiantes cursaron francés en la primavera de 2001.

Cursar(*x, a, s):* Estudiante *x* cursa asignatura *a* en el sementre *s.* Relación ternaria (aridad 3).

∃ *x Estudiante*(*x*) ∧ *Cursar*(*x, Francés, Primavera2001)*

b) Cada estudiante que estudia francés lo aprueba.

∀ *x, s Estudiante*(*x*) ∧ *Cursar*(*x*, *Francés*, *s*) ⇒ *Aprobar(x*, *Francés*, *s*)

Recordando orden de precedencia de conectivas:
¬,∧,∨,⇒,⇔
Ambas sentencias indican lo mismo.

c) Sólo un estudiante estudia griego en la primavera de 2001.

∃ x *Estudiante*(*x*) ∧ *Cursar*(*x*, *Griego*, *Primavera2001*) ∧∀*y Estudiante*(*y*) ∧ *y*!=*x* ⇒ ¬ *Cursar*(*y*, *Griego* , *Primavera2001*)

(∃ x *Estudiante*(*x*) ∧ *Cursar*(*x*, *Griego*, *Primavera2001*) ∧∀*y Estudiante*(*y*) ∧ *y*!=*x* **)** ⇒ ¬ *Cursar*(*y*, *Griego* , *Primavera2001*)

d) La mejor puntuación de griego siempre es mayor que la mejor puntuación en francés.

Puntuación(x,a, s): puntuación obtenida por un estudiante *x* en una asignatura *a* en un semestre *s.*

∀ *s* ∃ *x* ∀ *y Puntuación(x,Griego, s*) > *Puntuación*(*y, Francés, s*).

Ejercicio 8.6. Representar en LP. Enunciado

8.6 Represente las siguientes sentencias en lógica de primer orden, utilizando un vocabulario consistente (que usted debe definir):

***a*)** Algunos estudiantes estudian francés en la primavera de 2001.
***b*)** Cada estudiante que estudia francés lo aprueba.
***c*)** Sólo un estudiante estudia griego en la primavera de 2001.
***d*)** La mejor puntuación en griego siempre es mayor que la mejor puntuación en francés.
***e*)** Todas las personas que compran una póliza son inteligentes.
***f*)** Nadie compra una póliza cara.
***g*)** Hay un agente que vende pólizas sólo a la gente que no está asegurada.
***h*)** Hay un barbero que afeita a todos los hombres de la ciudad que no se afeitan ellos mismos.
***i*)** Una persona nacida en Reino Unido, cuyos padres sean ciudadanos de Reino Unido o residentes en Reino Unido, es un ciudadano de Reino Unido.
***j*)** Una persona nacida fuera de Reino Unido, que tenga uno de los padres ciudadano de Reino Unido o residente en Reino Unido, es ciudadano de Reino Unido por ascendencia.
***k*)** Los políticos pueden mentir a algunos todo el tiempo, y pueden mentir a todos algún tiempo, pero no pueden mentir a todos todo el tiempo.

Ejercicio 8.6. Representar en LP. Solución

e) Todas las personas que compran una póliza son inteligentes.

$\forall\ x\ Persona(x) \land (\exists\ y, Poliza(y) \land Comprar(x, y)) \Rightarrow Inteligente(x).$

f) Nadie compra una póliza cara.

$\forall\ x, y\ Persona(x) \land Poliza(y) \land Cara(y) \Rightarrow \neg\ Comprar(x,y)$

g) Hay un agente que vende pólizas sólo a la gente que no está asegurada.

$\exists\ \mathrm{x}\ Agente(x) \land \forall\ y, z\ Poliza(y) \land Vender(x, y, z) \Rightarrow (Persona(z) \land \neg\ Asegurado(z))$

h) Hay un barbero que afeita a todos los hombres de la ciudad que no se afeitan a sí mismos.

$\exists\ x\ Barbero(x) \land \forall\ y\ Hombre(y) \land \neg\ Afeitar(y, y) \Rightarrow Afeitar(x, y).$

Ejercicio 8.6. Representar en LP. Solución

i) Una persona nacida en Reino Unido, cuyos padres sean ciudadanos de Reino Unido o residentes en Reino Unido, es un ciudadano de Reino Unido.

Ejercicio 8.6. Representar en LP. Solución

i) Una persona nacida en Reino Unido, cuyos padres sean ciudadanos de Reino Unido o residentes en Reino Unido, es un ciudadano de Reino Unido.

Relación unaria: *Persona*

Relaciones binarias: Nacer(persona, lugar), Ser*Padre*(progenitor, hijo)*, Residir*(persona, lugar)

Relaciones ternarias: Ciudadano(persona, lugar, razón)

Ejercicio 8.6. Representar en LP. Solución

i) Una persona nacida en Reino Unido, cuyos padres sean ciudadanos de Reino Unido o residentes en Reino Unido, es un ciudadano de Reino Unido.

Relación unaria: *Persona*

Relaciones binarias: Nacer(persona, lugar), Ser*Padre*(progenitor, hijo)*, Residir*(persona, lugar)

Relaciones ternarias: Ciudadano(persona, lugar, razón)

$\forall$ x *Persona*(*x*) $\wedge$ *Nacer*(*x*,*UK*)$\wedge$($\forall$ *y* Ser*Padre*(*y*, *x*) $\Rightarrow$ (($\exists$ *r Ciudadano*(*y*,*UK*, *r*))$\vee$*Residir*(*y*,*UK*)))
$\Rightarrow$ *Ciudadano*(*x*, *UK*, *Nacimiento*)

Ejercicio 8.6. Representar en LP. Enunciado

8.6 Represente las siguientes sentencias en lógica de primer orden, utilizando un vocabulario consistente (que usted debe definir):

***a*)** Algunos estudiantes estudian francés en la primavera de 2001.
***b*)** Cada estudiante que estudia francés lo aprueba.
***c*)** Sólo un estudiante estudia griego en la primavera de 2001.
***d*)** La mejor puntuación en griego siempre es mayor que la mejor puntuación en francés.
***e*)** Todas las personas que compran una póliza son inteligentes.
***f*)** Nadie compra una póliza cara.
***g*)** Hay un agente que vende pólizas sólo a la gente que no está asegurada.
***h*)** Hay un barbero que afeita a todos los hombres de la ciudad que no se afeitan ellos mismos.
***i*)** Una persona nacida en Reino Unido, cuyos padres sean ciudadanos de Reino Unido o residentes en Reino Unido, es un ciudadano de Reino Unido.
***j*)** Una persona nacida fuera de Reino Unido, que tenga uno de los padres ciudadano de Reino Unido o residente en Reino Unido, es ciudadano de Reino Unido por ascendencia.
***k*)** Los políticos pueden mentir a algunos todo el tiempo, y pueden mentir a todos algún tiempo, pero no pueden mentir a todos todo el tiempo.

Ejercicio 8.6. Representar en LP. Solución

i) Una persona nacida en Reino Unido, cuyos padres sean ciudadanos de Reino Unido o residentes en Reino Unido, es un ciudadano de Reino Unido.

$\forall$ x *Persona*(*x*) $\wedge$ *Nacer*(*x*,*UK*)$\wedge$($\forall$ *y* Ser*Padre*(*y*, *x*) $\Rightarrow$ (($\exists$ *r Ciudadano*(*y*,*UK*, *r*))$\vee$*Residir*(*y*,*UK*)))
$\Rightarrow$ *Ciudadano*(*x*, *UK*, *Nacimiento*)

j) Una persona nacida fuera de Reino Unido ,que tenga uno de los padres ciudadano de Reino Unido o residente en Reino Unido, es ciudadano de Reino Unido por ascendencia.

$\forall$ x *Persona*(*x*) $\wedge$ $\neg$ *Nacer*(*x*,*UK*) $\wedge$ ($\exists$ *y* Ser*Padre*(*y*, *x*) $\wedge$ *Ciudadano*(*y*, *UK*, *Nacimiento*))
$\Rightarrow$ *Ciudadano*(*x*, *UK*, *Descendiente*)

k) Los políticos pueden mentir a algunos todo el tiempo, y pueden mentir a todos algún tiempo, pero no pueden mentir a todos todo el tiempo.

$\forall$ *x Políticos*(*x*) $\Rightarrow$ ($\exists$ *y* $\forall$ *t Persona*(*y*) $\wedge$ *Mentir*(*x*, *y*, *t*)) $\wedge$
($\exists$ *t* $\forall$ *y Persona*(*y*) $\Rightarrow$ *Mentir*(*x*, *y*, *t*)) $\wedge$
$\neg$ ($\forall$ *t* $\forall$ *y Persona*(*y*) $\Rightarrow$ *Mentir*(*x*, *y*, *t*))

3. Utilización

- Componentes del modelo:
  - **Dominio**: conjunto de objetos que contiene.
  - Un conjunto de **relaciones** entre objetos.
    - Relación: conjunto de tuplas (lista ordenada de elementos) de objetos que están relacionados.

- Elementos sintácticos:
  - Símbolos de **constante**: representan objetos.
  - Símbolos de **predicado**: representan relaciones.

- Cada modelo incluye una **interpretación** que dice qué objetos y relaciones se corresponden con los símbolos de constante y predicado.

- Los cuantificadores nos permiten expresar propiedades de colecciones de objetos $\forall$, $\exists$

- Reglas de Morgan para fórmulas cuantificadas:

$$\forall x\, P \equiv \neg \exists x \neg P$$

$$\exists x\, P \equiv \neg \forall x \neg P$$

3. Utilización. Asertos

- Las sentencias se añaden a una Base de Conocimiento (BC, como en lógica proposicional) con **DECIR**. Estas sentencias se denominan **asertos**.

- Ej. Juan es un rey: DECIR(BC, *Rey*(*Juan*))
- Ej. Ricardo es persona: DECIR(BC(*Persona*(*Ricardo*))
- Ej. Todos los reyes son personas: DECIR(BC, $\forall x$ *Rey*(x) $\Rightarrow$ *Persona*(x))

3. Utilización. Consultas a la BC

- Se puede preguntar a la Base de Conocimiento (BC, como en lógica proposicional) con **PREGUNTAR**. Estas sentencias se denominan **peticiones**.

Las siguientes peticiones son verdad:

- Ej. Pregunta si Juan es rey: PREGUNTAR(BC, *Rey*(*Juan*))
- Ej. Pregunta si Ricardo es una persona: PREGUNTAR(BC(*Persona*(*Ricardo*))

- Si queremos valores de una variable que hacen verdadera una sentencia se utiliza **PREGUNTARVAR**. La **respuesta** es una sustitución o **lista de ligaduras** (conjunto de parejas variable/término):

- Ej. : PREGUNTARVAR(BC, *Persona*(x))

respuesta: {x/*Ricardo*}, {x/*Juan*}

☞ Ejercicio 2. Ontología de listas.

Ejercicio 2. Ontología de listas. Enunciado

Ejercicio 2 (ontología de listas):

Considera la siguiente ontología para las listas:

–La función *Insert*(x,y) inserta el elemento x al principio de la lista y. Por ejemplo, la lista (A, B, C) se modela mediante este término:

Insert(A, *Insert*(B, *Insert*(C, *EmptyList*)))

donde *EmptyList* es un símbolo de constante que representa a la lista vacía.

–La función *Last*(x) devuelve el último elemento de una lista vacía. Devuelve la constante *ListError* para una lista vacía.

–Los axiomas de la ontología son los siguientes:

i) La lista vacía no tiene último elemento.

ii) El último elemento de una lista con un solo elemento es ese elemento.

iii) El último elemento de una lista y es también el último elemento de cualquier lista construida insertando un elemento al principio de y.

–Tu tarea es traducir los axiomas de la ontología al lenguaje de la lógica de primer orden, ~~y a continuación explicar cómo encontrarías el último elemento de la lista (*A*, *B*, *C*, *D*) con la ayuda de un demostrador de teoremas.~~

- Guía para resolverlo:

i) El último elemento de la constante *EmptyList* dará un error, siendo error la constatnte *ListError.* 'El último elemento' es la interpretación del predicado *Last.*

Ejercicio 2. Ontología de listas. Enunciado

Ejercicio 2 (ontología de listas):

Considera la siguiente ontología para las listas:

–La función *Insert*(x,y) inserta el elemento x al principio de la lista y. Por ejemplo, la lista (A, B, C) se modela mediante este término:

Insert(A, *Insert*(B, *Insert*(C, *EmptyList*)))

donde *EmptyList* es un símbolo de constante que representa a la lista vacía.

–La función *Last*(x) devuelve el último elemento de una lista vacía. Devuelve la constante *ListError* para una lista vacía.

–Los axiomas de la ontología son los siguientes:

i) La lista vacía no tiene último elemento.

ii) El último elemento de una lista con un solo elemento es ese elemento.

iii) El último elemento de una lista y es también el último elemento de cualquier lista construida insertando un elemento al principio de y.

–Tu tarea es traducir los axiomas de la ontología al lenguaje de la lógica de primer orden, ~~y a continuación explicar cómo encontrarías el último elemento de la lista (A, B, C, D) con la ayuda de un demostrador de teoremas.~~

- Guía para resolverlo:

ii) Debemos crear una lista de un sólo elemento (ej. x): añadiendo a la constante *EmptyList* el elemento, que se hace con el predicado *Insert.* 'El último elemento' es la interpretación del predicado *Last.* Esto se cumple para todo elemento x.

Ejercicio 2. Ontología de listas. Enunciado

Ejercicio 2 (ontología de listas):

Considera la siguiente ontología para las listas:

–La función *Insert*(*x*,*y*) inserta el elemento *x* al principio de la lista *y*. Por ejemplo, la lista (*A*, *B*, *C*) se modela mediante este término:

Insert(*A*, *Insert*(*B*, *Insert*(*C*, *EmptyList*)))

donde *EmptyList* es un símbolo de constante que representa a la lista vacía.

–La función *Last*(*x*) devuelve el último elemento de una lista vacía. Devuelve la constante *ListError* para una lista vacía.

–Los axiomas de la ontología son los siguientes:

i) La lista vacía no tiene último elemento.

ii) El último elemento de una lista con un solo elemento es ese elemento.

iii) El último elemento de una lista *y* es también el último elemento de cualquier lista construida insertando un elemento al principio de *y*.

–Tu tarea es traducir los axiomas de la ontología al lenguaje de la lógica de primer orden, ~~y a continuación explicar cómo encontrarías el último elemento de la lista (*A*, *B*, *C*, *D*) con la ayuda de un demostrador de teoremas.~~

❖ Guía para resolverlo:

iii) Tenemos la consante *y,* cuya interpretación es una lista no vacía, por tanto debemos indicar que este objeto es **cualquier** lista del dominio, y no es igual al objeto que representa la lista vacía. El predicado *Insert* inserta al principio, por tanto, **cualquier** elemento del domino que sea insertado con no modifica su elemento final. El último elemento de la lista *y* es **igual**, con y sin inserción de cualquier objeto al principio.

3. Utilización. Ejemplo dominio de parentesco I

- Dominio: los objetos son personas.
- Predicados unarios: Masculino, Femenino.
- Predicados binarios: Paternidad, Hermandad, Hijo, Hija, Cónyuge, Esposa, Marido, Abuelo, Nieto, Primo, Tía, Tío...
- Funciones (todos lo tienen) : Madre, Padre. Cada persona tiene un padre y una madre biológico.
- Se escribe el conocimiento del dominio y da lugar a los axiomas, que son las reglas básicas a partir de las cuales se derivan conclusiones útiles:
  - *Femenino*(*Ana*)
  - $\forall$ m, h *Madre*(h)=m $\Leftrightarrow$ Femenino(m) $\wedge$ *Paternidad*(m, h)
  - $\forall$ *x, y Hermandad*(*x,y*) $\Leftrightarrow$ (*x* ¬= *y*) $\wedge$ [$\exists$ *p Paternidad*(*p*, *x*) $\wedge$ *Paternidad*(*p*, *y*) $\wedge$ Masculino (*p*)]

❖ Ejercicio:

- Padre e hijo son relaciones inversas: $\forall$ *x, y Padre*(*x,y*) $\Leftrightarrow$ *Hijo*(*y,x*)
- Un abuelo es el padre del padre: $\forall$ *x, y Abuelo*(*x,y*) $\Leftrightarrow$ $\exists$ *p Paternidad*(*x, p*) $\wedge$ *Paternidad*(*p*, *y*)]

3. Utilización. Ejemplo dominio de parentesco II

- Los **teoremas** son fórmulas que se infieren a partir de los axiomas. Desde un punto de vista lógico, una BC sólo necesita contener a los axiomas porque los teoremas no aumentan el número de conclusiones que se obtienen de la BC.

$$\forall\, x, y\ Hermano(x,y) \Leftrightarrow Hermano(y,x) \text{ (teorema)}$$

- Desde un punto de vista práctico, los teoremas son esenciales para reducir el coste computacional al derivar sentencias nuevas. Sin los teoremas un sistema de razonamiento tiene que empezar desde el principio cada vez.

☞ Ejercicio 3. Axiomas de Peano

☞ Ejercicio 5. Paradoja del barbero

Ejercicio 5. Paradoja del barbero. Enunciado

Ejercicio 5 (la paradoja del barbero):

Traduce las afirmaciones siguientes a fórmulas de la lógica de primer orden. Se trata de la paradoja del barbero de Bertrand Russell; se da el caso de que ambas afirmaciones no pueden ser ciertas a la vez en ninguna interpretación. ~~¿Cómo demostrarías esto con ayuda de un demostrador de teoremas?~~

i) Cualquiera que no se afeite a si mismo debe ser afeitado por el barbero (se supone qu solamente hay un barbero).

ii) Aquel a quien el barbero afeite, no se afeita a si mismo.

Ejercicio 5. Paradoja del barbero. Enunciado

Ejercicio 5 (la paradoja del barbero):

Traduce las afirmaciones siguientes a fórmulas de la lógica de primer orden. Se trata de la paradoja del barbero de Bertrand Russell; se da el caso de que ambas afirmaciones no pueden ser ciertas a la vez en ninguna interpretación. ~~¿Cómo demostrarías esto con ayuda de un demostrador de teoremas?~~

i) Cualquiera que no se afeite a si mismo debe ser afeitado por el barbero (se supone qu solamente hay un barbero).

ii) Aquel a quien el barbero afeite, no se afeita a si mismo.

Afeitar(*x*,*y*): *x* afeita a *y*.

i) $\forall x \neg Afeitar(x,x) \Rightarrow Afeitar(Barbero,x)$

ii) $\forall x\, Afeitar(Barbero,x) \Rightarrow \neg Afeitar(x,x)$

Por tanto: $\forall x \neg Afeitar(x,x) \Leftrightarrow Afeitar(Barbero,x)$

Pero esta fórmula es una contradicción, porque para el caso de x=Barbero, se obtiene:

$\neg Afeitar(Barbero,Barbero) \Leftrightarrow Afeitar(Barbero,Barbero)$

que es insatisfacible. Así que las afirmaciones i) y ii) no pueden ser verdaderas al mismo tiempo.

Ejercicio 3. Axiomas de Peano. Enunciado

Ejercicio 3 (axiomas de Peano)

Los axiomas de Peano, también conocidos como los axiomas de Dedekind–Peano o los postulados de Peano postulates, son un conjunto de axiomas para los números naturales presentados por el matemático italiano del siglo XIX Giuseppe Peano. Han sido empleados prácticamente sin cambios hasta hoy. Tu tarea es traducir un subconjunto de ellos y la definición de la suma al lenguaje de la lógica de primer orden. ~~A continuación explica cómo demostrar que la suma es conmutativa con la ayuda de un demostrador de teoremas. Por último, explica cómo encontrar el resultado de la suma 2+3 con un demostrador de teoremas~~.

i) 0 es un número natural.

ii) Para cualquier número natural n, su sucesor $Suc(n)$ también es un número natural.

iii) Para cualquier número natural n, $Suc(n) = 0$ es falso. Es decir, no hay ningún número natural cuyo sucesor sea 0.

iv) Para cualesquiera números naturales m y n, si $Suc(m)=Suc(n)$, entonces $m=n$. O sea, Suc es una inyección.

v) La suma se define mediante las siguientes ecuaciones: $a+0=a$; $a+Suc(b)=Suc(a+b)$.

Ejercicio 5.14. Traducir a LP. Enunciado

Ejercicio 5.14 Traduce este argumento del lenguaje natural al lenguaje de la lógica de primer orden:

O Poirot es un genio o es un fraude.
Si alguien sabe como resolver un caso difícil, entonces es un genio.
Poirot sabe como resolver el caso del Orient Express.

Por consiguiente, si el caso del Orient Express es difícil, entonces Poirot no es un fraude.

Ejercicio 5.14. Traducir a LP. Solución

Ejercicio 5.14 Traduce este argumento del lenguaje natural al lenguaje de la lógica de primer orden:

O Poirot es un genio o es un fraude.
Si alguien sabe como resolver un caso difícil, entonces es un genio.
Poirot sabe como resolver el caso del Orient Express.

Por consiguiente, si el caso del Orient Express es difícil, entonces Poirot no es un fraude.

Objetos: Constantes: *Poirot, OrientExpress*

Relaciones: Predicados:
- unarios: *Genio*, *Fraude*, *Caso*, *Difícil*
- binarios: *Resuelve*, objeto alguien resuelve un objeto caso.

Ejercicio 5.14. Traducir a LP. Solución

Ejercicio 5.14 Traduce este argumento del lenguaje natural al lenguaje de la lógica de primer orden:

O Poirot es un genio o es un fraude.
Si alguien sabe como resolver un caso difícil, entonces es un genio.
Poirot sabe como resolver el caso del Orient Express.

Por consiguiente, si el caso del Orient Express es difícil, entonces Poirot no es un fraude.

Constantes: *Poirot, OrientExpress*

Predicados: unarios: *Genio, Fraude, Caso, Difícil*; binarios: *Resuelve*

Solución

$(Genio(Poirot) \lor Fraude(Poirot)) \land (\neg Genio(Poirot) \lor \neg Fraude(Poirot))$
$\forall x, y\ Caso(x) \land Dificil(x) \land Resuelve(y, x) \Rightarrow Genio(y)$
$Caso(OrientExpress) \land Resuelve(Poirot, OrientExpress)$
$\models$
$Dificil(OrientExpress) \Rightarrow \neg Fraude(Poirot)$

Ejercicio 5.18. Traducir a LP. Enunciado

Ejercicio 5.18 Formaliza el siguiente argumento expresado en lenguaje natural mediante un lenguaje de predicados de primer orden:

Todos los estudiantes son felices si aprueban una asignatura.
Nadie aprueba una asignatura sin hacer los ejercicios de la asignatura.
Yo soy estudiante y he hecho los ejercicios de Sistemas Inteligentes.

Por consiguiente, Yo soy feliz.

Ejercicio 5.18. Traducir a LP. Solución

Ejercicio 5.18 Formaliza el siguiente argumento expresado en lenguaje natural mediante un lenguaje de predicados de primer orden:

Todos los estudiantes son felices si aprueban una asignatura.
Nadie aprueba una asignatura sin hacer los ejercicios de la asignatura.
Yo soy estudiante y he hecho los ejercicios de Sistemas Inteligentes.

Por consiguiente, Yo soy feliz.

Constantes: *Yo, SistemasInteligentes*

Predicados:
- unarios: *Estudiante*, *Feliz*, *Asignatura*.
- binarios: *Aprobar*, *HacerEjercicios*.

Ejercicio 5.18. Traducir a LP. Solución

Ejercicio 5.18 Formaliza el siguiente argumento expresado en lenguaje natural mediante un lenguaje de predicados de primer orden:

Todos los estudiantes son felices si aprueban una asignatura.
Nadie aprueba una asignatura sin hacer los ejercicios de la asignatura.
Yo soy estudiante y he hecho los ejercicios de Sistemas Inteligentes.

Por consiguiente, Yo soy feliz.

Constantes: *Yo, SistemasInteligentes*

Predicados:
- unarios: *Estudiante*, *Feliz*, *Asignatura*.
- binarios: *Aprobar*, *HacerEjercicios*.

Solución

$\forall x\ Student(x) \wedge [\exists y\ Course(y) \wedge Passes(x,y)] \Rightarrow Happy(x)$

$\forall x, y\ Student(x) \wedge Course(y) Passes(x,y) \Rightarrow DoesHomework(x,y)$

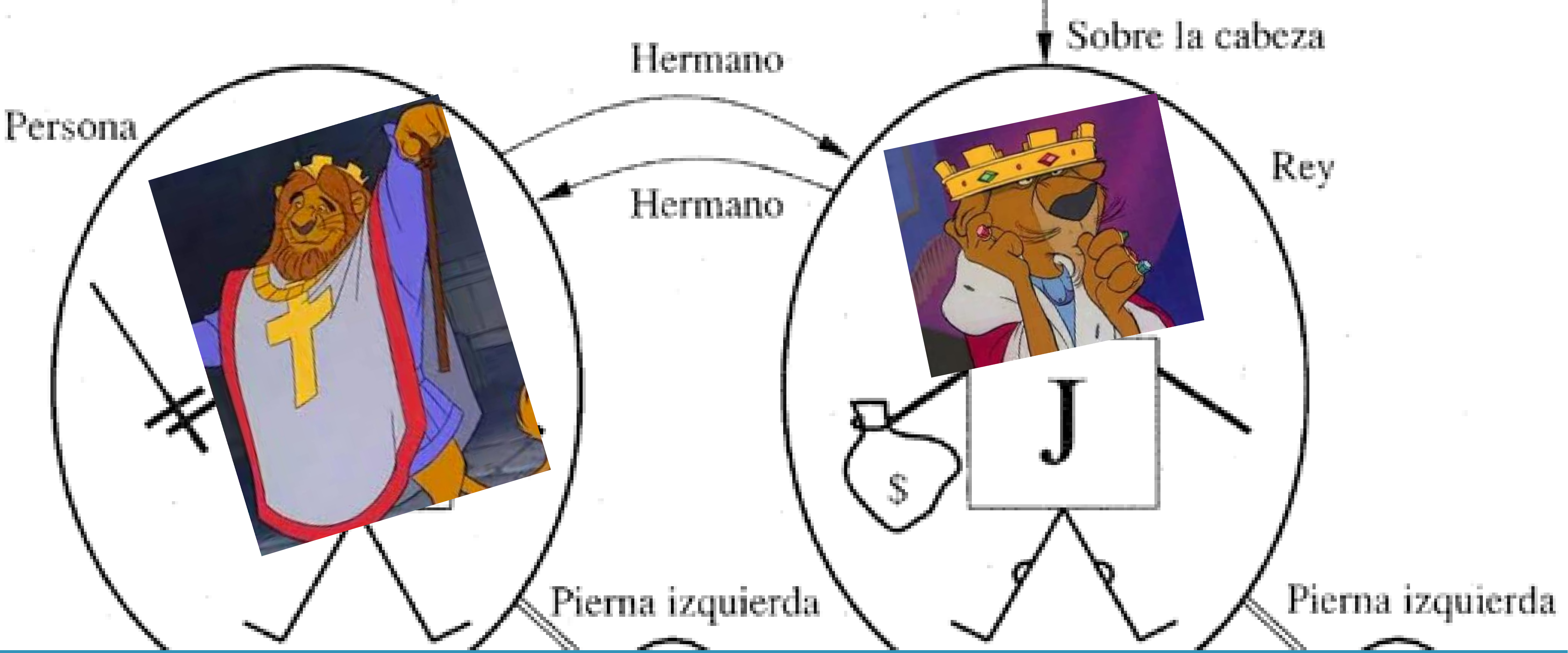
$Student(Me)\ \wedge\ Course(IS)\ \wedge\ DoesHomework(Me, IS)$

$\models$

$Happy(Me)$

4. Conclusiones

- La lógica de primer orden considera la existencia de objetos y relaciones entre éstos. Eso da lugar a un dominio.
- Un mundo posible, modelo, incluye un conjunto de objetos y una interpretación que asigna un significado a los símbolos de constante, predicado y función.
- La lógica de primer orden se está utilizando para certificar la seguridad del software crítico. Las pruebas encontradas por los demostradores de teoremas automatizados garantizan que un programa satisface ciertas propiedades de corrección. Queda mucho trabajo por hacer.



Gracias, Rosa ☺