Tema 1:
Sintaxis y Semántica

Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

Lógica Informática
(Tecnologías Informáticas)
Curso 2017–18

Contenido

Problema básico

Dados un conjunto de **afirmaciones** (hechos, hipótesis,...), $\mathcal{BC}$, y una afirmación, A, decidir si A ha de ser necesariamente cierta, suponiendo que todos las afirmaciones de $\mathcal{BC}$ lo son.

La **Lógica** proporciona formulaciones precisas de este problema y diferentes soluciones.
Algunos aspectos esenciales de este problema son:

1. El lenguaje para expresar (representar) las afirmaciones.
2. La definición precisa de “afirmación cierta”.
3. Mecanismos efectivos para garantizar la corrección de las deducciones.

A lo largo de este curso estudiaremos estas cuestiones en los dos casos más comunes: la **Lógica Proposicional**, y la **Lógica de Primer Orden**.

Lógica Proposicional

Características generales de la Lógica Proposicional:

- Sus expresiones (fórmulas) representan *afirmaciones* que pueden considerarse *verdaderas* o *falsas*.
- Se construyen a partir de expresiones básicas usando operadores (*conectivas*).
- Las conectivas se corresponden con formas sencillas de construir afirmaciones complejas en el lenguaje natural partiendo de otras más sencillas:
  - Conjunción: "... tal ... **y**... cual..."
  - Disyunción: "... tal ... **o**... cual ..."
  - Implicación "**Si** ... tal ... **entonces**... cual ..."
  - Negación: "**No** es cierto que tal ..."
- Sólo permite analizar las formas de razonamiento ligadas a este tipo de construcciones.

Símbolos del lenguaje

Formalmente, el *lenguaje de la Lógica Proposicional* consta de:

1. Un conjunto numerable de **variables proposicionales**: $VP = \{p, p_0, p_1, \ldots, q, q_0, q_1, \ldots, r, r_0, \ldots\}$
2. Operadores básicos de contrucción, **Conectivas lógicas**:
   - De aridad 1: $\neg$ (negación).
   - De aridad 2: $\vee$ (disyunción), $\wedge$ (conjunción), $\rightarrow$ (condicional) y $\leftrightarrow$ (bicondicional).
3. **Símbolos auxiliares**: "(" y ")".

Fórmulas

El conjunto de las *fórmulas proposicionales*, **PROP**, es el menor conjunto de expresiones que verifica:

- $VP \subseteq \mathbf{PROP}$,
- Es cerrado bajo las conectivas, es decir:
  - Si $F \in \mathbf{PROP}$, entonces $\neg F \in \mathbf{PROP}$.
  - Si $F, G \in \mathbf{PROP}$, entonces $(F \vee G)$, $(F \wedge G)$, $(F \rightarrow G)$, $(F \leftrightarrow G) \in \mathbf{PROP}$.

La sintaxis del lenguaje pretende evitar la ambigüedad en la interpretación de las fórmulas. Esa es la función de los símbolos auxiliares.

Árboles de formación

- Asociamos a cada fórmula un *árbol de formación* (esencialmente único) que describe el modo en que se construye la fórmula a partir de otras más sencillas.
- **Ejemplo**:

  - $\neg(\neg(p \vee q) \to (\neg r \wedge s))$
    - $(\neg(p \vee q) \to (\neg r \wedge s))$
      - $\neg(p \vee q)$
        - $(p \vee q)$
          - p
          - q
      - $(\neg r \wedge s)$
        - $\neg r$
          - r
        - s

- Las fórmulas que aparecen en el árbol de formación de una fórmula F se denominan **subfórmulas** de F.

Reducción de paréntesis

Para facilitar la lectura de las fórmulas adoptaremos los siguientes convenios de notación:

1. Omitiremos los paréntesis externos.
2. Daremos a las conectivas una precedencia de asociación. De mayor a menor, están ordenadas por: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.
   - **Ejemplo**: $F \wedge G \rightarrow \neg F \vee G$ es $((F \wedge G) \rightarrow (\neg F \vee G))$.
3. Cuando una conectiva se usa repetidamente, se asocia por la derecha: $F \vee G \vee H$ es $(F \vee (G \vee H))$.

Principio de Inducción sobre fórmulas

Gracias a la definición de **PROP** si deseamos probar que toda fórmula proposicional satisface cierta propiedad Ψ, podemos probarlo por **inducción sobre fórmulas**.

Para ello probamos:

1. **Caso base:** Todos los elementos de VP tienen la propiedad Ψ.
2. **Paso de inducción:**
   2.1 Si $F \in$ **PROP** tiene la propiedad Ψ, entonces $\neg F$ tiene la propiedad Ψ.
   2.2 Si $F, G \in$ **PROP** tienen la propiedad Ψ, entonces las fórmulas $(F \vee G)$, $(F \wedge G)$, $(F \rightarrow G)$ y $(F \leftrightarrow G)$ también tienen la propiedad Ψ.

Funciones de verdad

- Los elementos del conjunto $\{0,1\}$ se llaman **valores de verdad**. Se dice que 0 es el valor **falso** y el 1 es el valor **verdadero**.
- El significado de una conectiva se determina mediante su **función de verdad**:
  - $H_\neg(i) = \begin{cases} 1, & \text{si } i = 0; \\ 0, & \text{si } i = 1. \end{cases}$
  - $H_\vee(i,j) = \begin{cases} 0, & \text{si } i = j = 0; \\ 1, & \text{en otro caso.} \end{cases}$
  - $H_\wedge(i,j) = \begin{cases} 1, & \text{si } i = j = 1; \\ 0, & \text{en otro caso.} \end{cases}$
  - $H_\rightarrow(i,j) = \begin{cases} 0, & \text{si } i = 1, j = 0; \\ 1, & \text{en otro caso.} \end{cases}$
  - $H_\leftrightarrow(i,j) = \begin{cases} 1, & \text{si } i = j; \\ 0, & \text{en otro caso.} \end{cases}$

Valoraciones

- Las variables proposicionales se interpretan mediante una **valoración de verdad** (o interpretación), es decir, una aplicación

$$v : VP \to \{0, 1\}$$

- Podemos extender cada valoración, v, **de forma única**, al conjunto de todas las fórmulas de manera que para cada fórmula F se verifique:
  - $v(\neg F) = H_{\neg}(v(F))$.
  - $v((F \vee G)) = H_{\vee}(v(F), v(G))$.
  - $v((F \wedge G)) = H_{\wedge}(v(F), v(G))$.
  - $v((F \to G)) = H_{\to}(v(F), v(G))$.
  - $v((F \leftrightarrow G)) = H_{\leftrightarrow}(v(F), v(G))$.
- Se dice que $v(F)$ es el **valor de verdad** de F respecto de v.

Valor de verdad

Veamos el cálculo de $v(\neg(\neg(p \vee q) \vee (\neg r \vee s))$ en el árbol de formación (de abajo a arriba):

- $\neg(\neg(p \vee q) \vee (\neg r \vee s))$ (0)
  - $\neg(p \vee q) \vee (\neg r \vee s)$ (1)
    - $\neg(p \vee q)$ (0)
      - $(p \vee q)$ (1)
        - p **(1)**
        - q **(1)**
    - $(\neg r \vee s)$ (1)
      - $\neg r$ (1)
        - r **(0)**
      - s **(0)**

Tablas de verdad

Dada una valoración, v, el valor de verdad de una fórmula F respecto de v está determinado por los valores de verdad de las subfórmulas de F.

Ejemplo: si $v(p) = v(q) = 0$ y $v(r) = 1$, entonces

$$\begin{aligned} v(\neg((p \to q) \vee r)) &= H_\neg(H_\vee(v(p \to q), v(r))) = \\ &= H_\neg(H_\vee(H_\to(v(p), v(q)), 1)) = 0 \end{aligned}$$

Fijada v podemos presentar el cálculo de F mediante una tabla que recorre los valores de sus subfórmulas:

p	q	r	$p \to q$	$(p \to q) \vee r$	$\neg((p \to q) \vee r)$
0	0	1	1	1	0

Una **tabla de verdad** para F es una tabla similar que contiene una fila por cada posible valoración que asigne valores a las variables proposicionales que aparecen en F.

Validez y satisfactibilidad (I)

- Decimos que una fórmula F es **válida en** v, o que v es un **modelo** de F, si $v(F) = 1$.
  - Notación: $v \models F$.
  - Una valoración v es *modelo* de un conjunto de fórmulas U, $v \models U$, si v es modelo de todas las fórmulas de U.
- Una fórmula F es una **tautología** (o **válida**) si es válida para toda valoración (notación $\models F$).
- Una fórmula F es **satisfactible** (o consistente) si existe una valoración que es modelo de F. En caso contrario diremos que es **insatisfactible** (o inconsistente).
  - Análogamente, un conjunto de fórmulas U es satisfactible (o consistente) si existe una valoración que es modelo de U. En caso contrario diremos que es insatisfactible (o inconsistente).

Validez y satisfactibilidad (II)

Relación entre ambos conceptos:

Lema. Para cada $F \in \mathbf{PROP}$ se verifica:

- Si F es un tautología entonces F es satisfactible.
- F es una tautología si y sólo si $\neg F$ insatisfactible.

Ejemplos:

- Son tautologías: $(p \vee \neg p)$ y $((p \to q) \to p) \to p$.
- $p \wedge \neg p$ es insatisfactible y, por tanto, $\neg(p \wedge \neg p)$ es una tautología.
- $(p \to q) \to p$ es satisfactible pero no es una tautología.

Consecuencia Lógica

- Una fórmula F es **consecuencia lógica** de un conjunto de fórmulas U, si todo modelo de U es modelo de F. Es decir, para toda valoración, v,

$$v \models U \quad \Longrightarrow \quad v \models F$$

  - Notación: $U \models F$.
  - La relación de consecuencia lógica permite formular el problema básico en el marco de la lógica proposicional.

Relación entre consecuencia lógica, consistencia y validez:

Proposición. Sea $\{F_1, \ldots F_n\} \subseteq \mathbf{PROP}$. Son equivalentes:

- $\{F_1, \ldots, F_n\} \models F$
- $F_1 \wedge \cdots \wedge F_n \rightarrow F$ es un tautología.
- $\{F_1, \ldots, F_n, \neg F\}$ es insatisfactible.

Algoritmos de decisión (I)

Dado un conjunto de fórmulas proposicionales, U, un **algoritmo de decisión** para U es un algoritmo que dada $A \in PROP$, devuelve `SI` cuando $A \in U$, y `NO` si $A \notin U$.

Casos especialmente interesantes:

- **SAT** $= \{A \in PROP : \ A$ es satisfactible$\}$
- **TAUT** $= \{A \in PROP : \ A$ es una tautología$\}$
- Fijado $U \subseteq PROP$, la **Teoría de** U es

$$\mathcal{T}(U) = \{A \in PROP : \ U \models A\}$$

Un algoritmo de decisión para $\mathcal{T}(U)$ propociona una respuesta al Problema Básico expuesto al principio del tema.

Algoritmos de decisión (II)

Problema Básico:
Obtener un algoritmo que, dado un conjunto finito de fórmulas proposicionales, U, y una fórmula F, decida si $U \models F$.

El problema anterior se reduce a decidir la **satisfactibilidad** de una cierta fórmula (o si se prefiere, la **validez** de otra). Por tanto,

- La construcción de tablas de verdad proporciona un algoritmo (ineficiente) para decidir la consecuencia lógica.
- El Problema Básico es resoluble algorítmicamente, aunque no se conoce ninguna solución eficiente y se duda de la existencia de algoritmos de decisión eficientes para este problema, ya que ...
- ... determinar la satisfactibilidad de una fórmula proposicional es un problema **NP-completo**.

Algoritmos de decisión (III)

Problema Básico (bis):
Obtener un algoritmo eficiente que, dado un conjunto finito de fórmulas proposicionales, U, y una fórmula F, decida si $U \models F$.

Observaciones:

- Este problema es equivalente al de obtener un algoritmo eficiente para determinar la satisfactibilidad de una fórmula proposicional.
- Se trata de un **problema abierto**, que posiblemente tendrá una respuesta negativa (se cree que no existen algoritmos eficientes para resolver SAT).
- Para propósitos prácticos puede bastar con algoritmos eficientes para alguna clase especial de fórmulas.

Limitaciones de la lógica proposicional

- La lógica proposicional posee un semántica sencilla y existen algoritmos de decisión para sus problemas básicos, como SAT o la consecuencia lógica.
- Sin embargo, la expresividad de la lógica proposicional es bastante limitada.
- Existen problemas cuya descripción mediante lógica proposicional es complicada, ya que requieren un gran número de fórmulas o bien fórmulas de gran tamaño.
- Más aún, existen formas de razonamiento válido que no pueden ser expresadas mediante la lógica proposicional, por ejemplo:
  - Todos los hombres son mortales
  - Sócrates es un hombre.
  - Por tanto, Sócrates es mortal.
- La Lógica de Primer Orden extiende a la Lógica Proposicional proporcionando mayor expresividad.

Ejemplo (I)

Consideremos las siguientes afirmaciones:

1. Marco era pompeyano.
2. Todos los pompeyanos eran romanos.
3. Cada romano, o era leal a César, o le odiaba.
4. Todo el mundo es leal a alguien.
5. La gente sólo intenta asesinar a aquellos a quienes no es leal.
6. Marco intentó asesinar a César.
7. Todo pompeyano es leal a su padre.

¿Podemos deducir a partir de esta información que Marco era leal a César? ¿Podemos deducir que Marco odiaba a César? ¿Era César el padre de Marco?

Ejemplo (II)

- Podemos formalizar las afirmaciones observando que todas ellas expresan propiedades de los elementos de un cierto conjunto de individuos (romanos) y las relaciones que se dan entre ellos.
- Introduzcamos símbolos para expresar estas relaciones y para referirnos a los individuos de los que estamos hablando:
  - $P(x)$ expresa "x es pompeyano".
  - $R(x)$ expresa "x es romano".
  - $L(x, y)$: "x es leal a y".
  - $O(x, y)$: "x odia a y".
  - $IA(x, y)$: "x intentó asesinar a y".
  - Por último, parece natural introducir una función f que para cada x, devuelve el padre de x, $f(x)$.

Ejemplo (III)

Ahora podemos formalizar los enunciados anteriores:

1. $P(\mathbf{Marco})$ expresa "Marco es pompeyano"
2. $\forall x\,(P(x) \rightarrow R(x))$
   - "Todos los pompeyanos son romanos"
3. $\forall x\,(R(x) \rightarrow (L(x, \mathbf{Cesar}) \vee O(x, \mathbf{Cesar}))$
   - "Todo romano es leal a César o le odia"
4. $\forall x\,\exists y\, L(x, y)$
   - "Todo el mundo es leal a alguien".
5. $\forall x\,\forall y\,(IA(x, y) \rightarrow \neg L(x, y))$
   - "La gente sólo intenta asesinar a aquellos a quienes no es leal".
6. $IA(\mathbf{Marco}, \mathbf{Cesar})$
   - "Marco intentó asesinar a César".
7. $\forall x\,(P(x) \rightarrow L(x, f(x)))$
   - "Todo pompeyano es leal a su padre".

Ejemplo (IV)

- Para las preguntas podemos escribir:
  a. $L(\mathbf{Marco}, \mathbf{Cesar})$: Marco es leal a César.
  b. $O(\mathbf{Marco}, \mathbf{Cesar})$: Marco odia a César.
- Sin embargo, no podemos expresar que "Marco es el padre de César" sin considerar algún símbolo más.
- Una posibilidad es añadir a nuestro lenguaje el símbolo "$=$" para expresar al igualdad entre dos objetos. De este modo tendríamos:
  - $f(\mathbf{Marco}) = \mathbf{Cesar}$: César es el padre de Marco.
- Como puede verse, hemos ampliado el conjunto de símbolos disponibles en la lógica proposicional.
- El conjunto de símbolos introducidos constituye lo que denominamos un **Lenguaje de Primer Orden**.

Lenguaje de Primer Orden

- Un **lenguaje de primer orden** L consta de:
  - Símbolos lógicos (comunes a todos los lenguajes):
    1. Un conjunto de **variables**: $V = \{x_0, x_1, \dots\}$.
    2. **Conectivas lógicas**: $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$.
    3. **Cuantificadores**: $\exists$ (existencial), $\forall$ (universal).
    4. **Símbolos auxiliares**: "(", ")" y ","
  - Símbolos no lógicos (propios de cada lenguaje):
    1. Un conjunto L_C de **constantes**.
    2. Un conjunto de **símbolos de función** $L_F = \{f_0, f_1, \dots\}$, cada uno con su aridad.
    3. Un conjunto de **símbolos de predicados** $L_P = \{p_0, p_1, \dots\}$, cada uno con su aridad.

  (Los conjuntos V, L_F, L_C y L_P son disjuntos)
- Los símbolos de predicado de aridad 0 actúan como símbolos proposicionales.
- El símbolo $=$ no es un predicado común a todos los lenguajes de primer orden. Cuando está incluido en el lenguaje decimos que se trata de un **Lenguaje de Primer Orden con igualdad**.

Ejemplos

- En el ejemplo de los romanos se introdujo el lenguaje:

$$LR = \{\underbrace{\textbf{Marco}, \textbf{Cesar}}_{\text{constantes}}, \underbrace{P, L, O, R, IA}_{\text{símb. predicado}}, \underbrace{\quad f \quad}_{\text{símb. función}}\}$$

  P, R y f tienen aridad 1. L, O y IA tienen aridad 2.

- El lenguaje de la Aritmética (números naturales):

$$LA = \{\underbrace{\mathbf{0} \ , \ \mathbf{1}}_{\text{constantes}}, \overbrace{< \ , \ =}^{\text{símb. predicado}}, \underbrace{\cdot \ , \ +}_{\text{símb. de función}}\}$$

  $<, +$ y $\cdot$ tienen aridad 2.

- Un lenguaje para el parentesco:

$$LP = \{\underbrace{padre\_de, madre\_de, hijo, hermano, casados}_{\text{símb. predicado}}\}$$

  Todos de aridad 2.

Términos

- Los **términos** de un lenguaje L se definen como:
  1. Las variables y las constantes son términos.
  2. Si $t_1, \ldots, t_n$ son términos y f es un símbolo de función de L de aridad n, entonces $f(t_1, \ldots, t_n)$ es un término.
- Los términos son expresiones que me permiten hablar de objetos del mundo.
- Ejemplos:
  - Son términos del lenguaje LR:

$$\mathbf{Marco}, \mathbf{Cesar}, f(x), f(\mathbf{Cesar}), f(f(\mathbf{Cesar})), \ldots$$

  - Son términos del lenguaje de la Aritmética:

$$\mathbf{0}, +(x, y), \cdot(x, +(y, \mathbf{1})), \ldots$$

Utilizando la notación infija tradicional se escriben

$$x + y, x \cdot (y + \mathbf{1})$$

Fórmulas

- Las fórmulas son expresiones que permiten hablar de veracidad y falsedad.
- Las **fórmulas atómicas** de L son las expresiones $p(t_1, \ldots, t_n)$, donde p es un símbolo de predicado de aridad n y $t_1, \ldots, t_n$ son términos.
- Las **fórmulas** de L se definen como sigue:
  1. Las fórmulas atómicas de L son fórmulas de L.
  2. Si F y G son fórmulas de L, entonces $\neg F$, $(F \vee G)$ $(F \wedge G)$, $(F \to G)$ y $(F \leftrightarrow G)$ también lo son.
  3. Si x es una variable y F es una fórmula de L, entonces $\exists x\, F$ y $\forall x\, F$ también son fórmulas.

Ejemplos

- En LA, $\neg\exists x(x \cdot \mathbf{0} = y)$
- En LP, $\exists x(padre\_de(x,y) \wedge padre\_de(x,z))$.
  Pero $\exists x\ padre\_de(padre\_de(x,y),z)$, NO es una fórmula.
- En LR,
  $$\forall x\, \exists y\, L(x,y)$$
  $$\forall x\, (R(x) \to (L(x, \mathbf{Cesar}) \vee O(x, \mathbf{Cesar})))$$
- **Notación**: Para facilitar la lectura de las fórmulas y reducir el número de paréntesis adoptamos los mismos convenios que para la lógica proposicional:
  - Omitiremos los paréntesis externos.
  - Daremos a las conectivas una precedencia de asociación. De mayor a menor, están ordenadas por: $\neg, \wedge, \vee, \to, \leftrightarrow$.

Árboles de formación

Análisis sintáctico de la expresión $\exists x(y + y = x \cdot \mathbf{0} \vee \mathbf{1} < y)$

- $\exists x$
  - $\vee$
    - $=$
      - $+$
        - y
        - y
      - $\cdot$
        - x
        - $\mathbf{0}$
    - $<$
      - $\mathbf{1}$
      - y

O también:

- $\exists x(y + y = x \cdot \mathbf{0} \vee \mathbf{1} < y)$
  - $(y + y = x \cdot \mathbf{0} \vee \mathbf{1} < y)$
    - $y + y = x \cdot \mathbf{0}$
    - $\mathbf{1} < y$

Las fórmulas de los nodos se denominan **subfórmulas**.

Tratamiento de la cuantificación

- Significado intuitivo de $\exists x(y \cdot x = \mathbf{1})$:
- Dado y, existe un elemento, que denotamos por x (no sabemos exactamente su valor), que satisface la propiedad $x \cdot y = \mathbf{1}$, pero **no es cualquiera**.
- El símbolo que usemos para ese elemento no es importante: la fórmula $\exists z(y \cdot z = \mathbf{1})$ expresa la misma propiedad para y.
- La fórmula dice algo sobre y (en este caso, si sustituyo y por un elemento del universo, afirma que tal elemento tiene inverso a la derecha), no sobre el elemento x: Si cambio x por y, la fórmula resultante $\exists y(y \cdot y = \mathbf{1})$ **no expresa** lo mismo que la original.

Estancias libres y ligadas

- Una **estancia ligada** de una variable x en una fórmula F es una aparición de x en una subfórmula del tipo $\exists x\, F$ o $\forall x\, F$. En otro caso, diremos que es una **estancia libre**.
  - **Variable libre** en F: Al menos una estancia libre.
  - **Variable ligada** en F: Al menos una estancia ligada.
- Según las estancias de sus variables, podemos distinguir los siguientes tipos de expresiones:
  - Término **cerrado**: no contiene variables.
  - Fórmula **cerrada**: no contiene variables libres.
  - Fórmula **abierta**: no contiene cuantificadores.

Ejemplos

- $\exists x \, \forall y \, (x \cdot y = z \cdot \mathbf{1})$ no es cerrada (z es libre).
- $\exists x \, (\forall y \, (x \cdot y = \mathbf{1}) \vee x \cdot y = x)$ no es cerrada.
  - La variable y aparece libre **y** ligada.
  - Aunque sintácticamente es correcto, no escribiremos fórmulas en las que una misma variable aparezca libre y ligada. Usaremos en su lugar la fórmula

$$\exists x \, (\forall y \, (x \cdot y = \mathbf{1}) \vee (x \cdot z = x))$$

- $\forall x \exists y \, \forall z \, (z < x \leftrightarrow z < y)$ es una fórmula cerrada.
- $padre\_de(y, x) \vee hermano(z, x)$ es abierta.
- La fórmula

$$L(x, y) \wedge \exists z \, IA(y, z) \rightarrow \neg IA(x, z)$$

no es cerrada ni abierta.

Sustituciones (I)

- Una **sustitución**, θ, es una asignación de términos a un conjunto finito de variables.
- La forma de describirla, si $\theta(x_1) = t_1, \dots, \theta(x_n) = t_n$ y las restantes variables quedan invariantes, es
  $\theta = \{x_1/t_1, \dots, x_n/t_n\}$ ó $\theta = \{(x_1, t_1), \dots, (x_n, t_n)\}$
- Aplicación de θ a un término t:
  $$\theta(t) := \begin{cases} \theta(t), & \text{si } t \text{ es una variable;} \\ f(\theta(t_1), \dots, \theta(t_n)), & \text{si } t \equiv f(t_1, \dots, t_n) \end{cases}$$
  (también se denota por $t\{x_1/t_1, \dots, x_n/t_n\}$).
- Ejemplos:
  - Si $\theta = \{x/(x+y), z/\mathbf{0}, u/\mathbf{1}\}$, y $t = (x+y)+z$, entonces
    $$\theta(t) \equiv ((x+y)+y)+\mathbf{0}$$
  - $(x \cdot \mathbf{1})\{x/y, y/\mathbf{1}\} \equiv y \cdot \mathbf{1}$

Sustituciones (II)

- Aplicación de $\theta = \{x/t\}$ a una fórmula F:

$$F\{x/t\} := \begin{cases} p(t_1\{x/t\}, \dots, t_n\{x/t\}), & \text{si } F \equiv p(t_1, \dots, t_n) \\ \neg G\{x/t\}, & \text{si } F \equiv \neg G; \\ G\{x/t\} \vee H\{x/t\}, & \text{si } F \equiv G \vee H: \\ G\{x/t\} \wedge H\{x/t\} & \text{si } F \equiv G \wedge H: \\ G\{x/t\} \to H\{x/t\}, & \text{si } F \equiv G \to H: \\ G\{x/t\} \leftrightarrow H\{x/t\}, & \text{si } F \equiv G \leftrightarrow H: \\ \exists y G\{x/t\}, & \text{si } F \equiv \exists y G \text{ y } x \neq y; \\ \forall y G\{x/t\}, & \text{si } F \equiv \forall y G \text{ y } x \neq y; \\ \exists x G, & \text{si } F \equiv \exists x G; \\ \forall x G, & \text{si } F \equiv \forall x G; \end{cases}$$

- Análogamente se define $F\{x_1/t_1, \dots, x_n/t_n\}$.

Sustituciones (III)

- No toda sustitución es admisible:
  Si $F \equiv \exists x \neg (x = y)$ ("existen al menos dos elementos") y $\theta = \{y/x\}$, entonces $\theta(F) \equiv \exists x \neg (x = x)$ ¡Que es falso!
- Solución: No admitir la **creación** de nuevas estancias ligadas.
- Una variable x de F es **sustituible** por el término t si se cumple una de las siguientes condiciones:
  1. F es atómica;
  2. $F \equiv \neg G$ y x es sustituible por t en G;
  3. $F \equiv G \vee H$, $F \equiv G \wedge H$, $F \equiv G \to H$ o bien $F \equiv G \leftrightarrow H$ y x es sustituible por t en G y en H;
  4. $F \equiv \exists x G$; o bien, $F \equiv \exists y G$, $x \neq y$, y no ocurre en t, y x es sustituible por t en G.
  5. $F \equiv \forall x G$; o bien, $F \equiv \forall y G$, $x \neq y$, y no ocurre en t, y x es sustituible por t en G.
- x es sustituible por t en F si al hacer la sustitución no se crean estancias ligadas nuevas.

Notación

- En lo sucesivo, al escribir $F\{x/t\}$, supondremos que x es sustituible por t en F.
- Escribiremos $F(x_1, \ldots, x_n)$ si $x_1, \ldots, x_n$ son sus variables libres.
- Abreviaremos $F\{x_1/t_1, \ldots x_n/t_n\}$ por $F(t_1, \ldots, t_n)$.

Semántica

- **Objetivo**: Dotar de significado a los términos y fórmulas de un lenguaje de primer orden.
  - Términos cerrados: elementos del universo.
  - Significado de las fórmulas: propiedades sobre los elementos del universo.
- Una L-**estructura** (o **interpretación**) $\mathcal{M}$, consta de:
  - Un conjunto no vacío $M \neq \emptyset$ (el **universo** de la estructura).
  - Una interpretación en M para cada símbolo de L:
    1. Para cada constante c, $c^{\mathcal{M}} \in M$.
    2. Para cada función, f, de aridad $n > 0$, $f^{\mathcal{M}} : M^n \to M$.
    3. Para cada predicado, p, de aridad $n > 0$, $p^{\mathcal{M}} : M^n \to \{0, 1\}$ (equiv., $p^{\mathcal{M}} \subseteq M^n$).
    4. Si L es un LPO *con igualdad* la interpretación de $=$ es

$$\{(a, a) : \ a \in M\}$$

- Si no hay confusión, escribiremos M en vez de $\mathcal{M}$, p^M en lugar de $p^{\mathcal{M}}$, etc.

Ejemplos (I)

- Para LP, sea $\mathcal{M}_1$ la estructura dada por:
  - Universo: $M_1 = \{Pedro,\ Pablo,\ Ana,\ Laura\}$.
  - $padre\_de^{M_1} = \{(Pablo, Ana), (Pedro, Pablo)\}$.
  - $madre\_de^{M_1} = \{(Ana, Laura)\}$.
  - $hermano^{M_1} = \emptyset$.
  - $casados^{M_1} = \emptyset$.
- Para LP, consideremos $\mathcal{M}_2$ dada por:
  - Universo: $M_2 = \{0, 1, 2, 3, 4, 5, 6\}$.
  - $padre\_de^{M_2} \equiv$ ser múltiplo de.
  - $madre\_de^{M_2} \equiv$ ser menor.
  - $hermano^{M_2} \equiv$ primos entre sí.
  - $casados^{M_2} = \emptyset$.

Ejemplos (II)

- Para LA, sea $\mathcal{M}_3$ dada por:
  - Universo: $M_3 = \mathbb{N}$
  - $\mathbf{0}^{M_3} = 0$.
  - $\mathbf{1}^{M_3} = 1$.
  - La función $+^{M_3}$ es la suma de números naturales.
  - La función $\cdot^{M_3}$ es el producto de números naturales.
  - $=^{M_3}$ es la igualdad entre números naturales.
  - $<^{M_3}$ es el orden entre números naturales.
- Para LA, sea $\mathcal{M}_4$ dada por:
  - Universo: $M_4 = \mathbb{Q}$
  - $\mathbf{0}^{M_4} = \frac{1}{2}$.
  - $\mathbf{1}^{M_4} = 2$.
  - La función $+^{M_4}$ es la diferencia de números racionales.
  - La función $\cdot^{M_4}$ está dada por $p \cdot^{M_4} q = p$.
  - $=^{M_4}$ es la igualdad entre números naturales.
  - $<^{M_4}$ es el orden entre números racionales.

Interpretación de términos (I)

- Dada una L–estructura $\mathcal{M}$, a cada término t de L, **sin variables**, le corresponde un elemento de M, que denotamos por $t^{\mathcal{M}}$ (su **interpretación** en $\mathcal{M}$):
  - Si $t \equiv c$ una constante, entonces $t^{\mathcal{M}} = c^{\mathcal{M}} \in M$.
  - Si $t \equiv f(t_1, \ldots, t_n)$, entonces $t^{\mathcal{M}} = f^{\mathcal{M}}(t_1^{\mathcal{M}}, \ldots, t_n^{\mathcal{M}})$.

- Ejemplos:

$$
\begin{array}{rcl}
((\mathbf{0} \cdot \mathbf{1}) + \mathbf{1})^{M_3} & = & ((\mathbf{0} \cdot \mathbf{1})^{M_3} +^{M_3} \mathbf{1}^{M_3}) \\
 & = & (\mathbf{0}^{M_3} \cdot^{M_3} \mathbf{1}^{M_3}) + 1 \\
 & = & (0 \cdot 1) + 1 = 1
\end{array}
$$

$$
\begin{array}{rcl}
((\mathbf{0} \cdot \mathbf{1}) + \mathbf{1})^{M_4} & = & ((\mathbf{0} \cdot \mathbf{1})^{M_4} +^{M_4} \mathbf{1}^{M_4}) \\
 & = & (\mathbf{0}^{M_4} \cdot^{M_4} \mathbf{1}^{M_4}) - 2 \\
 & = & (\frac{1}{2} \cdot^{M_4} 2) - 2 \\
 & = & \frac{1}{2} - 2 = -\frac{3}{2}
\end{array}
$$

Interpretación de términos (II)

- Asociamos a cada L–estructura, $\mathcal{M}$, un lenguaje $L(\mathcal{M})$, que tiene todos los símbolos de L y, además, una constante $\underline{a}$ por cada elemento $a \in M$.
- La interpretación de los símbolos de $L(\mathcal{M})$ en $\mathcal{M}$ es la misma para los símbolos de L, y para cada $a \in M$,

$$\underline{a}^{\mathcal{M}} = a$$

- Ahora podemos calcular $t^{\mathcal{M}}$ para todo término de $L(\mathcal{M})$ sin variables:

$$\begin{array}{rcl} ((\underline{2} \cdot \underline{5}) + \mathbf{1})^{M_3} & = & ((\underline{2} \cdot \underline{5})^{M_3} +^{M_3} \mathbf{1}^{M_3}) \\ & = & (\underline{2}^{M_3} \cdot^{M_3} \underline{5}^{M_3}) + 1 \\ & = & (2 \cdot 5) + 1 = 11 \end{array}$$

$$\begin{array}{rcl} ((\underline{2}^{M_4} \cdot \underline{5}^{M_4}) + \mathbf{1})^{M_4} & = & ((x \cdot y)^{M_4} +^{M_4} \mathbf{1}^{M_4}) \\ & = & (\underline{2}^{M_4} \cdot^{M_4} \underline{5}^{M_4}) - 2 \\ & = & 2 - 2 = 0 \end{array}$$

Interpretación de fórmulas (I)

Dada una L–estructura $\mathcal{M}$, decimos que una fórmula F cerrada de $L(\mathcal{M})$ se **satisface** en $\mathcal{M}$, $\mathcal{M} \models F$, si:

- Si F es $p(t_1, \dots, t_n)$ (atómica), entonces $\mathcal{M} \models F$ sii $(t_1^{\mathcal{M}}, \dots, t_n^{\mathcal{M}}) \in p^{\mathcal{M}}$.
- Si F es $F_1 \vee F_2$, entonces $\mathcal{M} \models F$ sii se verifica que

$$\mathcal{M} \models F_1 \text{ ó } \mathcal{M} \models F_2$$

- Las conectivas $\wedge$, $\rightarrow$ y $\leftrightarrow$ se tratan de manera similar.
- Si F es $\neg F_1$, entonces $\mathcal{M} \models F$ sii no se tiene $\mathcal{M} \models F_1$.
- Si F es $\exists x F_1(x)$, entonces $\mathcal{M} \models F$ sii

$$\text{existe } b \in M \text{ tal que } \mathcal{M} \models F_1(\underline{b})$$

- Si F es $\forall x F_1(x)$, entonces $\mathcal{M} \models F$ sii

$$\text{para todo } b \in M, \text{ se tiene } \mathcal{M} \models F_1(\underline{b})$$

Interpretación de fórmulas (II)

- En particular, la definición anterior nos permite precisar cuándo una fórmula cerrada de L, F, es válida en $\mathcal{M}$ (o bien que $\mathcal{M}$ es un modelo de F) y escribir $\mathcal{M} \models F$.
- Si F no es cerrada, por definición,

$$\mathcal{M} \models F(x_1, \ldots x_n) \quad \Longleftrightarrow \quad \mathcal{M} \models \forall x_1 \cdots \forall x_n F(x_1, \ldots x_n)$$

- Si Σ es un conjunto de fórmulas de un lenguaje L y $\mathcal{M}$ una estructura para L, decimos que $\mathcal{M}$ **es un modelo de** Σ, si

$$\text{para toda fórmula } F \in \Sigma, \quad \mathcal{M} \models F.$$

Ejemplos

En $\mathcal{M}_1$:

- Universo: $M_1 = \{Pedro,\ Pablo,\ Ana,\ Laura\}$.
- $padre\_de^{M_1} = \{(Pablo, Ana), (Pedro, Pablo)\}$.
- $madre\_de^{M_1} = \{(Ana, Laura)\}$.
- $hermano^{M_1} = \emptyset$, $casados^{M_1} = \emptyset$.

Se tiene:

- $\mathcal{M}_1 \models \exists x(padre\_de(\underline{Pablo}, x) \wedge madre\_de(x, \underline{Laura}))$
- $\mathcal{M}_1 \models \neg\exists x\ padre\_de(x, \underline{Laura})$
- $\mathcal{M}_1 \models \forall x\, \forall y\, \forall z\, (padre(x, z) \wedge madre(y, z) \rightarrow \neg casados(x, y))$.
- $\mathcal{M}_1 \models hermano(x, y) \leftrightarrow hermano(y, x)$
- $\mathcal{M}_1 \not\models \exists x\ padre\_de(x, y)$

Ejemplos (II)

En $\mathcal{M}_2$:

- Universo: $M_2 = \{0, 1, 2, 3, 4, 5, 6\}$.
- $padre\_de^{M_2} \equiv$ ser múltiplo de.
- $madre\_de^{M_2} \equiv$ ser menor.
- $hermano^{M_2} \equiv$ primos entre sí, $casados^{M_2} = \emptyset$.

Se tiene:

- $\mathcal{M}_2 \models \exists x(padre\_de(\underline{4}, x) \wedge madre\_de(x, \underline{3}))$
- $\mathcal{M}_2 \models \exists x \; padre\_de(x, \underline{3})$
- $\mathcal{M}_2 \models hermano(x, y) \leftrightarrow hermano(y, x)$
- $\mathcal{M}_2 \models \exists x \forall y \; padre\_de(x, y)$ $[x = 0]$
- ¿Se tiene $\mathcal{M}_2 \models hermano(x, y) \rightarrow \neg padre\_de(x, y)$?

Validez y Consistencia

- Una fórmula $F(x_1, \ldots, x_n)$ de L es **satisfactible** si existe una L–estructura $\mathcal{M}$ y elementos $a_1, \ldots, a_n \in M$ tales que

$$\mathcal{M} \models F(\underline{a}_1, \ldots, \underline{a}_n)$$

  - Ejemplo: $\exists x \; padre\_de(x, y)$
- Un conjunto de fórmulas cerradas Σ de un lenguaje L es **consistente** si existe una L–estructura, $\mathcal{M}$, tal que

$$\text{para toda formula } F \in \Sigma, \quad \mathcal{M} \models F$$

- Una fórmula F es **lógicamente válida** si para toda estructura $\mathcal{M}$ se tiene que $\mathcal{M} \models F$ (Notación: $\models F$).
  - Ejemplo: $\forall x P(x) \vee \exists x \neg P(x)$

Consecuencia lógica y equivalencia

- Diremos que una fórmula F es **consecuencia lógica** de un conjunto de fórmulas cerradas Σ, $(\Sigma \models F)$, si para toda L–estructura $\mathcal{M}$ se tiene que

$$\text{si } \mathcal{M} \models \Sigma, \text{ entonces } \mathcal{M} \models F$$

  - Es decir, si todo modelo de Σ es modelo de F.
- Dos fórmulas F y G son (lógicamente) **equivalentes** $F \equiv G$ si la fórmula $F \leftrightarrow G$ es lógicamente válida.
- Los problemas de la consistencia, consecuencia lógica y la validez para la lógica primer orden, **no son decidibles**.