# Tema 3.
# Lógica de predicados

Lecturas :
• CAPÍTULOS 8,9 de Russell & Norvig
• CAPÍTULOS 15,16 de Nilsson

BIBLIOGRAFÍA:


• E. Paniagua Arís, J. L. Sánchez González, F. Martín Rubio, "Lógica computacional", Thomson

• Melvin Fitting "First-order logic and automated theorem proving", Springer-Verlag (New-York 1990)

# Lógicas de orden superior

- La lógica proposicional carece de capacidad para expresar ciertos tipos de conocimiento. Por ejemplo:
  - Descripción concisa de entornos con muchos objetos.
  - Enunciados generales relativos a todos o algunos de los objetos en un dominio específico, las relaciones entre ellos, la existencia de objetos con ciertas propiedades, etc.
- Tipos de lógica
  - **Lógica de orden cero** o **"Lógica proposicional"**

    Objetos, conectores lógicos.
  - **Lógica de primer orden** o **"Lógica de predicados"**

    ++ funciones, predicados, cuantificadores cuyos argumentos son variables cuyo dominio incluye un conjunto objetos

    ej. $(\forall x)\ Humano(x) \Rightarrow Mortal(x)$
  - **Lógica de segundo orden**:

    ++ funciones, predicados, cuantificadores cuyos argumentos pueden ser predicados

    ejs. $(\exists P)\ [P(A) \wedge P(B)]$

    $(\forall x)(\forall y)\ [(x=y) \Leftrightarrow (\forall P)(P(x) \Leftrightarrow P(y))]$ [***Leibniz***, 1666]
  - **Lógicas de orden superior**:

    ++ funciones, predicados, cuantificadores cuyos argumentos pueden ser predicados de predicados ...
  - **Teoría de tipos**

# Lógica de primer orden

**Ejemplo**: Familia + amigos

- **Ontología**
  - **Conectores, Variables, Cuantificadores**
  - **Constantes**

    **Objetos**: Pedro, Pablo, María

    **Funciones**: $madreDe^1$, $mejorAmigoDe^1$

    **Predicados**: $Hombre^1$, $Mujer^1$ , $Niñ@^2$ , $Hijo^2$ , $Hija^2$, $Casad@^1$, $Feliz^1$,…

- **Definición**: "La madre de uno es su progenitor mujer"

  $(\forall x,y)\ (MadreDe(x,y)\ \Leftrightarrow\ ProgenitorDe(x,y) \wedge Mujer(x))$

- **Aserciones generales**:

  "Todos los descendientes de María están casados, pero no son felices"

  $(\forall x)\ (DescendienteDe(x,María) \Rightarrow\ Casado(x) \wedge\ \neg Feliz(x))$

- **Aserciones existenciales**: "Algún hijo del mejor amigo de Pedro tiene algún descendiente"

  $(\exists x)(\exists y)(HijoDe(x,mejorAmigoDe(Pedro)) \wedge DescendienteDe(y,x))$

# Lenguaje, I

- **Constantes**:
  - **Objetos simbólicos** (normalmente en mayúscula)
    `Ej.      P, Q, Juan, PuertaDeAlcalá`
  - **Funciones** (en general, en minúscula)
    Argumentos de entrada: lista de términos entre paréntesis
    Evalúa a: un término.

    Ej. $\texttt{padreDe}^{1}$ , $\texttt{distanciaEntre}^{2}$
  - **Relaciones o predicados** (en general, en mayúscula)
    Argumentos de entrada: lista de términos entre paréntesis
    Evalúa a: un valor de verdad ("Verdadero" o "Falso").

    Ejs. $\texttt{PadreDe}^{2}$, $\texttt{Blanco}^{1}$
    **Las relaciones unarias se denominan "propiedades"**

**Notas:**

- El superíndice denota la aridad de la función o predicado (es decir, el número de argumentos)
- Los objetos simbólicos pueden ser considerados como funciones de aridad cero

# Lenguaje, II

- **Signos de puntuación**
  - **Coma: ,**
  - **Paréntesis:** ( ) [ ] { }

- **Conectores de lógica proposicional**:
  ¬, ∧, ∨, ⇒, ⇔ (en orden de prioridad)

- **Variables** (en general, en minúscula)

  Las variables pueden tomar valores en un dominio.
  Ejs. x (reales), n (enteros), p (personas), etc.

  `Ejs. x,y,p,q,…, p1,p2,…`

- **Cuantificadores**
  - **Cuantificador universal (∀)**
  - **Cuantificador existencial (∃)**

  Se usan con variables. Dichas variables se dicen que están ligadas al cuantificador.

$$(\forall x)\ \{P(x) \Rightarrow (\exists y)\ [R(x,y) \lor S(f(x))]\}$$

Ámbito de x

Ámbito de y

# Término, átomo, literal

- **Un término** puede ser
  - Un **objeto**

    `Ej.    Pedro, P, P1,…`
  - Una **variable**

    `Ej.    x,y`
  - Una **función** de aridad n **seguida de n términos** separados por comas y entre paréntesis.

    `Ej.    padreDe(x), distanciaEntre(A,B)`
  - Un caso especial es un **término sin variables** ("término base", "ground term").

- Una **fórmula atómica (átomo)** es una **relación** de aridad n **seguida de n términos** separados por comas y entre paréntesis.

  `Ej.    PadreDe(x,Pedro)`

- **Un literal** puede ser:
  - Un **literal positivo**: Una fórmula atómica

    `Ej.    HermanoDe(x,Pedro)`
  - Un **literal negativo**: La negación de una fórmula atómica

    `Ej.    ¬ PadreDe(Juan,padreDe(Pedro))`

# Fórmulas bien formadas

- Una **fbf** se construye usando átomos y conectores del mismo modo que en lógica proposicional.
- Si w es una fbf y x es una variable, las siguientes expresiones son también fbfs
  - $(\forall x)\ w$
  - $(\exists x)\ w$

  En general (pero no siempre), la variable $x$ forma parte de $w$.
  Si $x$ forma parte de $w$, se suele escribir $w(x)$

- **Fbf cerrada**: Aquella en la que todas las variables están cuantificadas.
  - $(\forall x)\ [P(x) \Rightarrow R(x)]$
  - $(\exists x)\ [P(x) \Rightarrow (\exists y)(R(x,y) \Rightarrow S(f(x))]$

Nota: El **orden** en el que aparecen $\forall, \exists$ en la fbf es **importante**
$x, y \in \{Personas\}$
$PadreDe(x,y)$ "x es el padre de y" (relación binaria)

| | |
|---|---|
| $(\forall x)(\forall y)PadreDe(x,y)$ | "Todo el mundo es padre de todo el mundo" |
| $(\exists x)(\exists y)PadreDe(x,y)$ | "Existe alguien que tiene al menos un padre" |
| $(\forall x)(\exists y)PadreDe(x,y)$ | "Todo el mundo es padre de alguien" |
| $(\exists y)(\forall x)PadreDe(x,y)$ | "Hay alguien que tiene a todo el mundo como padre" |
| $(\exists x)(\forall y)PadreDe(x,y)$ | "Hay alguien que es padre de todo el mundo" |
| $(\forall y)(\exists x)PadreDe(x,y)$ | "Todo el mundo tiene al menos un padre" |

# Interpretación y semántica

- **Constantes**
  - Las **constantes son objetos simbólicos** que corresponden a **objetos del mundo real**
  - Los **predicados de aridad 1** corresponden a **propiedades del objeto**
  - Los **predicados de aridad n > 1** expresan **relaciones entre objetos**

- **Variables**
  - El **dominio** de una variable es el conjunto de objetos simbólicos entre los que puede tomar valor dicha variable
  - **Asignación**: Operación mediante la cual una variable que aparece en una fbf es reemplazada por un valor particular dentro de su dominio.

- **Cuantificadores**
  - **Cuantificador universal**: `(∀x) w(x)` tiene valor `Verdadero` sii `w(x)` tiene valor `Verdadero` para todas las posibles asignaciones de la variable `x`
  - **Cuantificador existencial**: `(∃x) w(x) tiene` valor `Verdadero` sii `w(x)` tiene valor `Verdadero` para alguna de las posibles asignaciones de `x`

# Reglas de equivalencia y reglas de inferencia

- **Reglas de equivalencia**
  - **Reglas de equivalencia de lógica proposicional**
  - **Renombramiento de variables cuantificadas.**
    El nuevo nombre debe ser diferente que el de las otras variables en la fbf
    - $(\forall x)\ w(x) \equiv (\forall y)\ w(y)$
    - $(\exists x)\ w(x) \equiv (\exists y)\ w(y)$
  - $\neg(\forall x)\ w(x) \equiv (\exists x)\ \neg w(x)$
  - $\neg(\exists x)\ w(x) \equiv (\forall x)\ \neg w(x)$

- **Reglas de inferencia**
  - **Reglas de inferencia de lógica proposicional**
  - **Instanciación del universal** (IU) [Correcta]
    $(\forall x)\ w(x) \vdash_{UI} w(A)$, donde $A$ es alguno de los valores en el dominio de $x$
  - **Generalización del existencial** (GE) [Correcta]
    $w(A) \vdash_{EG} (\exists x)\ w(x)$, donde $x$ es el símbolo de una variable cuyo dominio incluye a $A$

# Skolemización

Un **cuantificador existencial** (∃) puede ser **eliminado** de una fbf **reemplazando** cada ocurrencia de la **variable cuantificada existencialmente** por o bien

- Un **objeto de Skolem (constante)**, si no hay variables cuantificadas universalmente cuyo ámbito abarque el ámbito de la variable cuyo ∃ se está eliminando.

  `Ej. (∃x) w(x)`

  **`forma de Skolem:`** `w(Sk)`

  Sk es un objeto cuya identidad desconocemos, pero que sabemos que existe

- Una **función de Skolem** cuyos argumentos son las variables cuantificadas de manera universal cuyo ámbito abarque el ámbito de la variable cuyo ∃ se está eliminando.

  `Ej. "Todas las personas tienen una altura"`

  `(∀p)[(∃h) Altura(p,h)]`

  dominio de p: Personas.

  dominio de h: Reales positivos.

  **`Forma de Skolem:`** `(∀p) Altura(p,a(p))`

  `a(p)` es una función de Skolem (desconocida, pero que sabemos que existe) que toma como argumento una persona y devuelve su altura

# Metateoremas para formas de Skolem

- Metateorema SK1: La forma de Skolem de una fbf NO es equivalente a la fbf original

  $w(Sk) \models (\exists x)\ w(x)$ PERO $(\exists x)\ w(x) \not\models w(Sk)$

  Ej.

  $P(A) \lor P(B) \models (\exists x)\ P(x)$

  PERO $P(A) \lor P(B) \not\models P(Sk)$

- Metateorema SK2 (Loveland, 1978):

  La forma de Skolem de un conjunto de fbfs es **equivalente inferencialmente** al conjunto original de fbfs.

  Esto es, la forma de Skolem de un conjunto de fbfs es satisfacible exactamente en aquellos casos donde la base de conocimiento original es satisfacible

  - Un conjunto de fbfs es satisfacible sii su forma de Skolem es satisfacible.
  - Un conjunto de fbfs es insatisfacible sii su forma de Skolem es insatisfacible

# FNC en lógica de primer orden

1. **Eliminar implicaciones** $\Leftrightarrow$, $\Rightarrow$
2. **Reducir el ámbito de la negación** $\neg$
   - Leyes de De Morgan

     $\neg(w1 \lor w2) \equiv \neg w1 \land \neg w2$

     $\neg(w1 \land w2) \equiv \neg w1 \lor \neg w2$
   - Eliminación de negaciones dobles $(\neg\neg w \equiv w)$
   - Combinación de $\neg$ con cuantificadores

     $\neg(\forall x)\ w(x) \equiv (\exists x)\ \neg w(x)$

     $\neg(\exists x)\ w(x) \equiv (\forall x)\ \neg w(x)$
3. **Estandarizar las variables**: Renombrar las variables de tal forma que variables distintas tengan símbolos (nombres) diferentes

   $[(\forall x)\ [P(x) \Rightarrow R(x)]] \lor [(\exists x)\ P(x)] \equiv [(\forall x)\ [P(x) \Rightarrow R(x)]] \lor [(\exists y)\ P(y)]$
4. **Skolemización**: Eliminar los cuantificadores existenciales reemplazando las variables correspondientes por constantes de Skolem o funciones de Skolem.
5. **Convertir a forma prenexa** desplazando todos los cuantificadores universales al principio de la fbf

   fbf en **forma prenexa** = **Prefijo** (lista de cuantificadores)
   \+ **Matriz** (fórmula sin cuantificadores)
6. **Eliminar los cuantificadores universales**
7. **Usar leyes distributivas y reglas de equivalencia** de lógica proposicional para transformar la **matriz a FNC**

# Ejemplo 1: Conversión a FNC

$[(\forall x)\ Q(x)] \Rightarrow$
$(\forall x)(\forall y)\ [(\exists z)\ [P(x,y,z) \Rightarrow (\forall u)\ R(x,y,u,z)]$

**1. Eliminación de implicaciones ⇔, ⇒**

$\neg[(\forall x)\ Q(x)] \vee$
$(\forall x)(\forall y)\ [(\exists z)\ [\neg P(x,y,z) \vee (\forall u)\ R(x,y,u,z)]$

**2. Reducción del ámbito de las negaciones**:

$[(\exists x)\ \neg Q(x)] \vee$
$(\forall x)(\forall y)\ [(\exists z)\ [\neg P(x,y,z) \vee (\forall u)\ R(x,y,u,z)]$

**3. Estandarización de variables**

$[(\exists w)\ \neg Q(w)] \vee$
$(\forall x)(\forall y)\ [(\exists z)\ [\neg P(x,y,z) \vee (\forall u)\ R(x,y,u,z)]$

**4. Skolemización**:

$\neg Q(A) \vee (\forall x,y)[\neg P(x,y,f(x,y)) \vee (\forall u) R(x,y,u,f(x,y))]$

**5. Forma prenexa:**

$(\forall x,y,u)[\neg Q(A) \vee \neg P(x,y,f(x,y)) \vee R(x,y,u,f(x,y))]$

**6. Eliminación de los cuantificadores universales**

$\neg Q(A) \vee \neg P(x,y,f(x,y)) \vee R(x,y,u,f(x,y))$

**7. Conversión a FNC:** la fbf ya está en FNC

# Ejemplo 2: Conversión a FNC

"Todo el que ama a todos los animales es amado por alguien"

$$(\forall x)[(\forall y)\{Animal(y) \Rightarrow AmaA(x,y)\} \Rightarrow (\exists y)\ AmaA(y,x)]$$

**1. Eliminación de implicaciones ⇔, ⇒**

$$(\forall x)[\neg(\forall y)\{\neg Animal(y) \lor AmaA(x,y)\} \lor (\exists y)\ AmaA(y,x)]$$

**2. Reducción del ámbito de las negaciones**:

$$(\forall x)[(\exists y)\{Animal(y) \land \neg AmaA(x,y)\} \lor (\exists y)\ AmaA(y,x)]$$

**3. Estandarización de variables**

$$(\forall x)[(\exists y)\{Animal(y)\land\neg AmaA(x,y)\}\lor(\exists z)AmaA(z,x)]$$

**4. Skolemización**:

$$(\forall x)[\{Animal(f(x)) \land \neg AmaA(x,f(x))\} \lor AmaA(g(x),x)]$$

**5. Forma prenexa:** la fbf ya está en forma prenexa

**6. Eliminación de cuantificadores universales**

$$\{Animal(f(x))\land\neg AmaA(x,f(x))\} \lor AmaA(g(x),x)$$

**7. Conversión a FNC usando leyes distributivas y reglas de equivalencia**

$$\{Animal(f(x)) \lor AmaA(g(x),x)\} \land \{\neg AmaA(x,f(x)) \lor AmaA(g(x),x)\}$$

# Sustitución

- Considérese la expresión $w$ que contiene las variables $v_1, v_2, v_3, \ldots, v_n$.
  La sustitución $\sigma$ es **un reemplazamiento simultáneo** de variables en w por términos.

  $$w = w(v_1, v_2, v_3, \ldots, v_n)$$
  $$\big\downarrow \quad \boldsymbol{\sigma} = \{v_1:=t_1,\ v_2:=t_2,\ \ldots, v_n:=t_n\}$$
  $$w\boldsymbol{\sigma} = w(t_1, t_2, t_3, \ldots, t_n)$$

  El término $t_i$ no puede contener a la variable $v_i$.
  El término $t_i$ es una instanciación de la variable $v_i$.
  $w\sigma$ es una **instancia** de $w$.
- Considérese las siguientes sustituciones

  $$\boldsymbol{\sigma_1} = \{x_1:=t_1,\ x_2:=t_2,\ \ldots, x_n:=t_n\}$$
  $$\boldsymbol{\sigma_2} = \{y_1:=s_1,\ y_2:=s_2,\ \ldots, y_m:=s_m\}$$

  La composición de estas sustituciones es

  $$\boldsymbol{\sigma_1 \cdot \sigma_2} = \{x_1:=t_1\boldsymbol{\sigma_2},\ x_2:=t_2\boldsymbol{\sigma_2},\ \ldots, x_n:= t_n\boldsymbol{\sigma_2},\ y_1:=s_1,\ y_2:=s_2,\ \ldots, y_m:=s_m\}$$

  - $w(\boldsymbol{\sigma_2 \cdot \sigma_1}) = (w\boldsymbol{\sigma_2})\boldsymbol{\sigma_1}$
  - $(\boldsymbol{\sigma_1 \cdot \sigma_2})\cdot\boldsymbol{\sigma_3} = \boldsymbol{\sigma_1}\cdot(\boldsymbol{\sigma_2 \cdot \sigma_3})$
  - $\boldsymbol{\sigma_1 \cdot \sigma_2} \neq \boldsymbol{\sigma_2 \cdot \sigma_1}$ [En general]

  $w=P(x,y)$, $\boldsymbol{\sigma_1}=\{x:=f(y)\}$, $\boldsymbol{\sigma_2}=\{y:=A\}$

  $\boldsymbol{\sigma_1 \cdot \sigma_2}=\{x:=f(A),\ y:=A\}$, $\quad \boldsymbol{\sigma_2 \cdot \sigma_1}=\{y:=A,\ x:=f(y)\}$

  $w\boldsymbol{\sigma_1}= P(f(y),y) \quad w(\boldsymbol{\sigma_1 \cdot \sigma_2}) = (w\boldsymbol{\sigma_1})\boldsymbol{\sigma_2}= P(f(A),A)$

  $w\boldsymbol{\sigma_2}= P(x,A) \quad w(\boldsymbol{\sigma_2 \cdot \sigma_1}) = (w\boldsymbol{\sigma_2})\boldsymbol{\sigma_1}= P(f(y),A)$

# Sustitución: ejemplos

- Ejemplo 1: $w = A(x,y,f(z))$
  $\sigma = \{x:=D,\ y:=g(x),\ z:=C\}$
  $w\sigma = A(D,g(x),f(C))$

- Ejemplo 2: $w = B(x) \lor C(x,f(y))$
  $\sigma = \{x:=y,\ y:=g(A)\}$
  $w\sigma = B(y) \lor C(y,f(g(A)))$

- Ejemplo 3: $w = P(x,f(y),B)$
  - $\sigma_1 = \{x:=z,\ y:=w\}$
    $w\sigma_1 = P(z,f(w),B)$ [Variante alfabética]
  - $\sigma_2 = \{y:=A\}$
    $w\sigma_2 = P(x,f(A),B)$
  - $\sigma_3 = \{x:=g(z),y:=A\}$
    $w\sigma_2 = P(g(z),f(A),B)$
  - $\sigma_4 = \{x:=C,y:=A\}$
    $w\sigma_2 = P(C,f(A),B)$ [Instancia sin variables (instancia base)]

Una instancia base de una fbf es una fbf obtenida por sustitución de la fbf original que no contiene ninguna variable.

# Unificación

- Se dice que la **sustitución** $\sigma$ es el **unificador** de un conjunto de fbfs $\Gamma = \{w_1, w_2, \ldots, w_m\}$ cuando se cumple $w_1\sigma = w_2\sigma = \ldots = w_m\sigma$.
  - Este proceso se llama **unificación**
  - El resultado de la unificación es **único** excepto posibles variantes alfabéticas

  Ej. {P(x,f(y),B), P(z,f(B),B)}

  $\sigma$ = {x:=A, y:=B, z:=A}

  Unificación: {P(A,f(B),B)}
- **La unificación es correcta:** Dado que todas las variables están universalmente cuantificadas, la unificación es una forma particular de instanciación universal, que es una regla de inferencia correcta

- **Unificador más general (UMG)**

  La sustitución $\mu$ es el unificador más general de un conjunto de fbfs $\Gamma = \{w_1, w_2, \ldots, w_m\}$,

  $\mu = \text{umg}(\Gamma)$, si cualquier unificador del mismo conjunto de fbfs $\sigma$ puede ser expresado como

  $\sigma = \mu \cdot \sigma'$

- El conjunto de **discrepancia** de un conjunto no vacío de fbfs $\Gamma = \{w_1, w_2, \ldots, w_m\}$ es la primera subexpresión (la que está más a la izquierda) donde las fbfs en $\Gamma$ no concuerdan

  Ej. $\Gamma$ = {P(x,f(A),B), P(x,g(A),B)}

  D($\Gamma$) = {f(A),g(A)}

# Algoritmo de unificación

- **Algoritmo de unificación** (Chang & Lee 1973)

  **Entrada**: $\Gamma$ . **Salida**: umg($\Gamma$) [ unificador más general de $\Gamma$ ]

  1. $k \leftarrow 0$; $\Gamma_k \leftarrow \Gamma$; $\sigma_k \leftarrow \varepsilon$ (sustitución vacía)
  2. Si $\Gamma_k$ tiene un solo elemento, devolver $\sigma_k$. En caso contrario continuar
  3. $D_k \leftarrow$ conjunto de discrepancia de $\Gamma_k$
  4. Si $D_k$ contiene (i) un término $t_k$
     (ii) una variable $v_k$ que no esté en $t_k$

     entonces,

     $\sigma_{k+1} \leftarrow \sigma_k \cdot \{v_k := t_k\}$

     $\Gamma_{k+1} \leftarrow \Gamma_k\{v_k := t_k\}$(eliminar fbfs repetidas)

     En caso contrario salir devolviendo fallo [$\Gamma$ no es unificable]

  5. $k \leftarrow k+1$ e ir al paso 2.

# Algoritmo de unificación

Ejemplo 1.

$\Gamma$ = {P[f(x,g(A,y)),g(A,y)], P[f(x,z),z]}

1. $D_0 \leftarrow$ {z,g(A,y)};
   $\sigma_1 \leftarrow$ {z:=g(A,y)}, $\Gamma_1 \leftarrow$ {P[f(x,g(A,y)),g(A,y)]};

Ejemplo 2.

$\Gamma$={A(x,z,g(x,y,f(z))), A(y,f(x),w)}

1. $D_0 \leftarrow$ {x,y};
   $\sigma_1 \leftarrow$ {x:=y}, $\Gamma_1 \leftarrow$ {A(y,z,g(y,y,f(z))), A(y,f(y),w)}
2. $D_1 \leftarrow$ {z,f(y)};
   $\sigma_2 \leftarrow \sigma_1 \cdot$ {z:=f(y)};
   $\Gamma_2 \leftarrow$ {A(y, f(y),g(y,y,f(f(y)))),A(y,f(y),w)};
3. $D_2 \leftarrow$ {g(y,y,f(f(y))),w};
   $\sigma_3 \leftarrow \sigma_2 \cdot$ {w:= g(y,y,f(f(y)))};
   $\Gamma_3 \leftarrow$ {A(y,f(y),g(y,y,f(f(y))))};

# Resolución en LPO

- **Una cláusula** es una disyunción de literales
  (puede ser representada como un conjunto de literales con disyunciones implícitas entre todos los literales del conjunto).

- Una fbf en **FNC** es una conjunción de cláusulas
  (puede ser representada como un conjunto de cláusulas con conjunciones implícitas entre las cláusulas del conjunto).

- **Resolución binaria entre cláusulas**
  Asumamos que los literales $\{\phi,\psi\}$ tienen un unificador más general $\mu$ .
  Entonces, del conjunto de cláusulas $\{\{\phi\}\cup\Sigma_1,\{\neg\psi\}\cup\Sigma_2\}$, es posible inferir $w = (\Sigma_1\cup\Sigma_2)\mu$

  Ej. $\Gamma=\{P(x)\vee Q(f(x)),\ R(g(y))\vee\neg Q(f(A))\}$

  $\phi = Q(f(x));$

  $\psi = Q(f(A));$

  $\mu = mgu(\{\psi,\phi\}) = \{x:=A\}$

  Se infiere por resolución $\{P(A)\vee R(g(y))\}$

# Resolución generalizada

- La resolución binaria es **correcta** pero **no es completa para refutación**

  Ej. $\Gamma = \{\neg P(x) \vee \neg P(y),\ P(z) \vee P(r)\}$

  $\mu_1 = \{z := x\}$

  Se infiere por resolución $\{\neg P(y) \vee P(r)\}$

  De $\{\neg P(x) \vee \neg P(y),\ \neg P(y) \vee P(r)\}$

  usando el umg $\mu_2 = \{r := y\}$

  se infiere por resolución $\{\neg P(x) \vee \neg P(y)\}$

  Todas las resoluciones binarias sólo producen variantes alfabéticas de estas cláusulas.

- Asumamos que todos los literales de la cláusula $K_1$ pueden ser unificados con los literales negados de la cláusula $K_2$ a través del unificador más general $\mu$.

  Entonces, del conjunto de cláusulas $\{K_1 \cup \Sigma_1,\ K_2 \cup \Sigma_2\}$ es posible inferir $w = (\Sigma_1 \cup \Sigma_2)\mu$

  Ej. $\Gamma = \{\neg P(x) \vee \neg P(y), P(z) \vee P(r)\}$

  $\mu = \{y := x,\ z := x,\ r := x\}$

  Se infiere por resolución $\{\square\}$

# Subsunción generalizada

- La cláusula $K_1$ **subsume a la cláusula** $K_2$ si existe una **sustitución** $\sigma$ tal que

$$K_1\sigma \subset K_2$$

- Desde el punto de vista del mecanismo de **inferencia** la **cláusula subsumida**, $K_2$ puede ser **eliminada** de la base de conocimiento

- Ejemplo:

**$K_1$:** `DisfrutaCon(Ringo,z)`

**$K_2$:** `¬Placentero(Leer)∨DisfrutaCon(Ringo,Leer)`

**$\sigma$ :** `{z:= Leer}`

[Intuición: Cláusula $K_1$ es más "restrictiva" que la cláusula $K_2$]

# *Modus ponens* generalizado

Asumiendo que las fbfs $w_1$ y $w_2$ tienen un unificador más general $\mu$, se puede inferir $w_3\mu$ de $w_1$ y $w_2 \Rightarrow w_3$

- <u>Ejemplo 1:</u> ¿Es Juan malvado?

  Reglas: $(\forall x)$ [Rey(x)∧Avaricioso(x)⇒Malvado (x)]

  Hechos: Rey(Juan), Avaricioso(Juan)

  se infiere ($\mu$={x:=Juan})

  Malvado(Juan)

- <u>Ejemplo2:</u> ¿Le gustan a Ringo los cacahuetes?

  Reglas: $(\forall x)$ [Comida(x)⇒LeGusta(Ringo,x)]

  $(\forall p,x)$ [¬Muerto(p)∧ComeA(p,x) ⇒Comida(x)]

  $(\forall x)$ [ComeA(Juan,x)⇒ComeA(Jorge,x)]

  Hechos: ComeA(Juan,Cacahuetes), ¬Muerto(Juan), Comida(Manzana), Comida(Pollo)

  De: ComeA(Juan,Cacahuetes), ¬Muerto(Juan),

  [¬Muerto(p)∧ComeA(p,x) ⇒Comida (x)]

  (usamos {p:= Juan, x:=Cacahuetes})

  Se infiere: Comida(Cacahuetes)

  De: [Comida(x)⇒LeGusta(Ringo,x)],Comida(Cacahuetes)

  ($\mu$={x:=Cacahuetes})

  Se infiere: **LeGusta(Ringo,Cacahuetes)**

# Extracción de respuestas

Consideremos la Base de Conocimiento $\Delta$.
Asumamos que necesitamos una prueba de $w = (\exists x) P(x)$
y queremos saber qué valores de $x$ hacen que esta expresión sea consecuencia lógica de $\Delta$.

**Truco de Green** (1999):
Usar refutación por resolución e incluir un literal "respuesta" en $\neg w$ para llevar la cuenta de las sustituciones realizadas.

Añadir a la BC: $(\forall x)\ [\neg P(x) \vee Respuesta(x)]$

**En general:** Añadir un literal $Respuesta(x_1, x_2, \ldots, x_m)$ a cada cláusula generada de la negación del teorema que se quiere demostrar.

# Extracción de respuestas

Ejemplo:

BC: Todos los humanos son mortales. $(\forall x)[\mathrm{Humano}(x) \Rightarrow \mathrm{Mortal}(x)]$

Sócrates es humano. $\mathrm{Humano}(\mathrm{Sócrates})$

¿Hay algún humano mortal? $(\exists x)[\mathrm{Humano}(x) \wedge \mathrm{Mortal}(x)]$

Añadir a la BC: $(\forall x)[\neg \mathrm{Humano}(x) \vee \neg \mathrm{Mortal}(x) \vee \mathrm{Ans}(x)]$

$\alpha = \{\neg \mathrm{Humano}(x) \vee \mathrm{Mortal}(x),\ \mathrm{Humano}(\mathrm{Sócrates}),$
$\neg \mathrm{Humano}(y) \vee \neg \mathrm{Mortal}(y) \vee \mathrm{Ans}(y)\}$

$\mathrm{Mortal}(\mathrm{Sócrates})$
$\neg \mathrm{Mortal}(\mathrm{Sócrates}) \vee \mathrm{Ans}(\mathrm{Sócrates})$
$\mathrm{Ans}(\mathrm{Sócrates})$

# Consecuencia lógica en LPO

- **Consecuencia lógica:** La fbf $w$ es consecuencia lógica de la BC $\Delta$ si todo modelo de $\Delta$ es también modelo de $w$.

$$\Delta \models w$$

- **Mecanismos de inferencia:**
  - **Proposicionalización**
    - Convertir fbfs de LPO a fbfs proposicionales eliminando todos los cuantificadores universales y existenciales.
    - Apicar a la BC proposicionalizada mecanismos de inferencia de lógica proposicional.

  - Uso de **reglas de inferencia generalizadas.**

    En particular, usar prueba mediante refutación con una regla de inferencia por resolución generalizada.

- Metateorema [Turing (1936), Church (1936)]

  "**El problema de determinar si una fbf $w$ es consecuencia lógica de la BC $\Delta$ en LPO es semi-decidible**".

  Existen algoritmos tales que siempre que la expresión es consecuencia lógica, dicen SÍ en tiempo finito.

  No existen algoritmos tales que siempre que la expresión no sea consecuencia lógica, digan NO en tiempo finito.

# Proposicionalización

- Solución a $\Delta_{LPO} \models w_{LPO}$ en LPO por **proposicionalización**:
  - Construir $\alpha_{LPO} = \Delta_{LPO} \wedge \neg w_{LPO}$
  - Transformar $\alpha_{LPO}$ a FNC ($\exists$ eliminados por Skolemización)
  - Transformar toda cláusula en $\alpha_{LPO}$ a cláusulas proposicionales haciendo todas las posibles asignaciones a las variables en $\alpha_{FOL}$ , de tal forma que sólo aparezcan términos base (términos sin variables).

  **Problema**: Si consideramos funciones, hay **infinitos** términos base,

  ej., `hijoDe(hijoDe(…hijoDe(hijoDe(`*`Pablo`*`))…))`
- Metateorema [**Herbrand** (1930)]

  El conjunto de fbfs en LPO en FNC $\alpha_{LPO}$ es insatisfacible sii hay un conjunto finito de cláusulas base (cláusulas sin variables) de $\alpha_{LPO}$ que es insatisfacible en cálculo proposicional.
- Algoritmo

  n ← 0; $\alpha_{LPO}$

  1. n ← n+1
  2. Crear la FNC proposicional $\alpha_n$ instanciando los términos en $\alpha_{LPO}$ usando una profundidad máxima n en las evaluaciones de funciones.

  Hasta que la resolución sobre $\alpha_n$ produzca la cláusula vacía

  Problema: El algoritmo funciona si $\alpha_{LPO}$ es INSAT, pero nunca para si $\alpha_{LPO}$ es SAT.

# Refutación por resolución en LPO

**Refutación por resolución generalizada**

Considerar la base de conocimiento $\Delta$ y la fbf $w$ en LPO.
¿Se cumple $\Delta \models w$ ?

1. Incluir la fbf negada en la BC $\alpha = \{\Delta \wedge \neg w\}$
2. Convertir a FNC
3. Aplicar resolución generalizada
   (i) Si la resolución genera la cláusula vacía ($\alpha$ es INSAT) entonces $\Delta \models w$
   (ii) Si la resolución no genera la cláusula vacía ($\alpha$ es SAT) entonces w no es consecuencia lógica de $\Delta$.

- Si $\Delta \models w$, el algoritmo para y devuelve "Sí".
- Si $\Delta \not\models w$, el algoritmo puede no parar nunca.

**La resolución** es **correcta**, **completa para refutación** pero **semidecidible**.

# Trucos

Normalmente,

$$(\forall x) \quad [w1(x) \Rightarrow w2(x)]$$

$$(\exists x) \quad [w1(x) \wedge w2(x)]$$

Ej. "Todo el mundo en la UAM es inteligente"

$(\forall x) \quad [En(x,UAM) \Rightarrow Inteligente(x)]$

"Hay gente en la UAM que es inteligente"

$(\exists x) \quad [En(x,UAM) \wedge Inteligente(x)]$

ERRORES COMUNES

$(\exists x) \quad [En(x,UAM) \Rightarrow Inteligente(x)]$ [DEMASIADO DÉBIL]

"Hay alguien que o bien es inteligente o no está en la UAM"

$(\forall x) \quad [En(x,UAM) \wedge Inteligente(x)]$ [DEMASIADO RESTRICTIVA]

"Todo el mundo está en la UAM y es inteligente"

# ¿Mató la curiosidad al gato?

- Todo el que ama a los animales es amado por alguien.
- El que mata a un animal no es amado por nadie.
- Javier ama a todos los animales.
- El gato Tuno fue matado por Javier o por la curiosidad.
- ¿Mató la curiosidad al gato?

# ¿Mató la curiosidad al gato?

- Todo el que ama a todos los animales es amado por alguien.

  $(\forall x)[(\forall y)[Animal(y) \Rightarrow AmaA(x,y)] \Rightarrow (\exists z)\ AmaA(z,x)]$

- Cualquiera que mate a un animal no es amado por nadie.

  $(\forall x)[(\exists y)(Animal(y) \wedge MataA(x,y)) \Rightarrow (\forall z)\ \neg AmaA(z,x)]$

- Javier ama a todos los animales.

  $(\forall x)\ [Animal(x) \Rightarrow AmaA(Javier,x)]$

- El gato Tuno fue matado por Javier o por la curiosidad.

  Gato(Tuno)

  $MataA(Javier,Tuno) \vee MataA(Curiosidad,Tuno)$

- Un gato es un animal

  $(\forall x)\ Gato(x) \Rightarrow Animal(x)$

- ¿Mató la curiosidad a Tuno?

  MataA(Curiosidad,Tuno) ???

# ¿Mató la curiosidad al gato?

A. Animal(f(x)) ∨ AmaA(g(x),x)

B. ¬AmaA(u,f(u)) ∨ AmaA(g(u),u)

C. ¬Animal(y) ∨ ¬MataA(x,y) ∨ ¬AmaA(z,x)

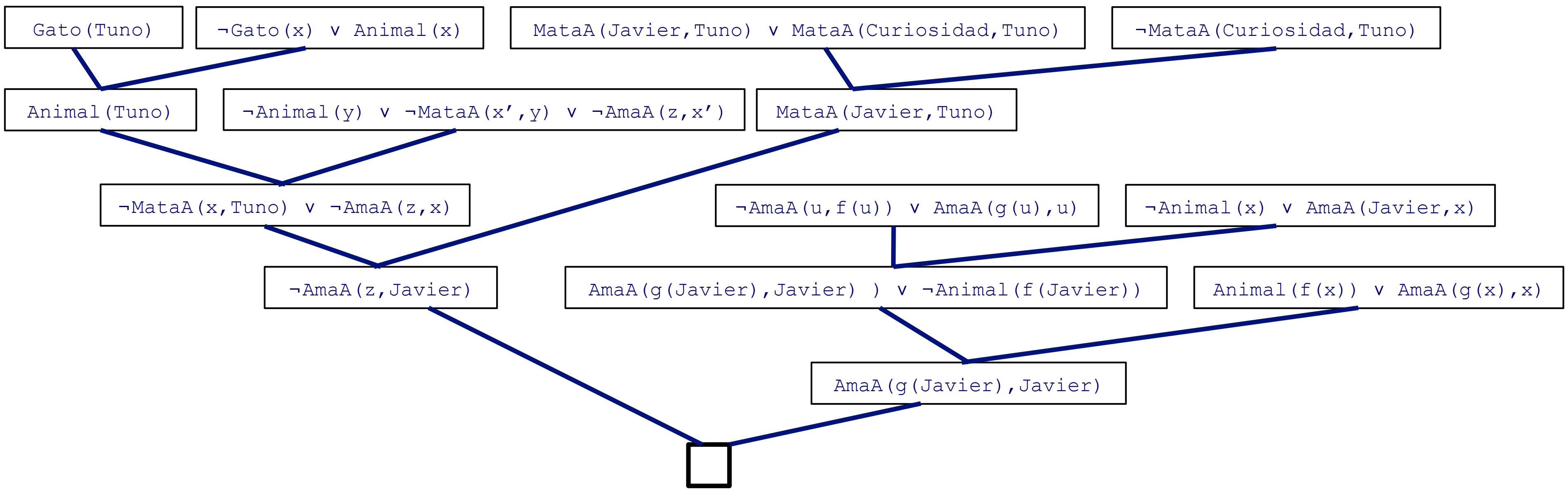
D. ¬Animal(x) ∨ AmaA(Javier,x)

E. MataA(Javier,Tuno) ∨ MataA(Curiosidad,Tuno)

F. Gato(Tuno)

G. ¬Gato(x) ∨ Animal(x)

H. ¬MataA(Curiosidad,Tuno)




**Nota**: Las variables de cláusulas diferentes son diferentes (incluso si tienen el mismo nombre)

# Ontología para conjuntos

- **Ontología** para **teoría de conjuntos** (vocabulario de objetos, predicados, funciones)
  - Objetos:
    - Conjunto vacío: `{}`
    - Elementos del conjunto: `A, B, C,...`
  - Predicados: $\mathrm{Conjunto}^1$, $\mathrm{Miembro}^2(\in)$, $\mathrm{Subconjunto}^2(\subset)$, $\mathrm{Igual}^2(=)$
  - Funciones: $\mathrm{adjoin}^2$ $(\{\,|\,\})$, $\mathrm{unión}^2(\cup)$, $\mathrm{intersección}^2(\cap)$

- **Axiomas** de la teoría de conjuntos:

1. $\forall s\,[\mathrm{Conjunto}(s) \Leftrightarrow (s=\{\}) \vee (\exists x,s')\,[\mathrm{Conjunto}(s') \wedge (s=\{x|s'\})]]$
2. $\neg(\exists x,s)\,[\{x|s\}=\{\}]$
3. $(\forall x,s)\,[x\in s \Leftrightarrow s=\{x|s\}]$
4. $(\forall x,s)\,[x\in s \Leftrightarrow (\exists y,s')\,[s=\{y|s'\} \wedge (x=y \vee x\in s')]]$
5. $(\forall s_1,s_2)\,[s_1\subset s_2 \Leftrightarrow (\forall x)\,[x\in s_1 \Rightarrow x\in s_2]]$
6. $(\forall s_1,s_2)\,[s_1=s_2 \Leftrightarrow (s_1\subset s_2 \wedge s_2\subset s_1)]$
7. $(\forall x,s_1,s_2)\,[x\in(s_1 \cap s_2) \Leftrightarrow (x\in s_1 \wedge x\in s_2)]$
8. $(\forall x,s_1,s_2)\,[x\in(s_1 \cup s_2) \Leftrightarrow (x\in s_1 \vee x\in s_2)]$

# Listas

- Ontología para listas
  - Objetos constantes: `Nil, A,B,C... (átomos)`
  - Predicados: $\texttt{Find}^2$, $\texttt{Atom}^1$, $\texttt{Listp}^1$ , $\texttt{Null}^1$
  - Funciones: $\texttt{cons}^2$, $\texttt{append}^2$, $\texttt{first}^1$, $\texttt{rest}^1$

- ¿Axiomas?

# Números naturales

- Ontología para números naturales
  - Objeto constante: `0`
  - Predicados
    - Test de número natural: $\texttt{NumNat}^1$
    - Predicado igualdad: $\texttt{Igual}^2$ (=)
  - Funciones
    - Función sucesor: $\texttt{sucesor}^1$
    - Función adición: $\texttt{suma}^2$ (+)

- Axiomas de Peano

$$\mathrm{NumNat}(0)$$
$$(\forall n)[\mathrm{NumNat}(n) \Rightarrow \mathrm{NumNat}(\mathrm{sucesor}(n))]$$
$$(\forall n)[0 \neq \mathrm{sucesor}(n)]$$
$$(\forall m,n)[m \neq n \Rightarrow \mathrm{sucesor}(m) \neq \mathrm{sucesor}(n)]$$
$$(\forall n)[\mathrm{NumNat}(n) \Rightarrow \mathrm{suma}(n,0)=n]$$
$$(\forall m,n)[\mathrm{NumNat}(m) \wedge \mathrm{NumNat}(n) \Rightarrow \mathrm{suma}(\mathrm{sucesor}(m),n)=\mathrm{sucesor}(\mathrm{suma}(m,n))]$$

Nota: El principio de **inducción** sólo puede ser formulado en **lógica de segundo orden**, donde las relaciones y funciones de lógica de primer orden se tratan como objetos.
Por ello, se pueden hacer afirmaciones sobre ellas (ej. se puede escribir un predicado de segundo orden que especifica cuándo una relación de primer orden es transitiva).

# Predicado igualdad

- **Igualdad:** Predicado especial que permite expresar que dos términos diferentes se refieren al mismo objeto del mundo real.
  - Igual(padreDe(Juan), Pedro)

    [**también**: padreDe(Juan)=Pedro )

    Significado: "Pedro es el padre de Juan"
  - $(\exists x,y)[Hermano(x,Jose) \wedge Hermano(y,Jose) \wedge \neg Igual(x,y)]$

    (**también**: $(\exists x,y)[Hermano(x,Jose) \wedge Hermano(y,Jose) \wedge (x \neq y)]$)

    Significado: "Jose tiene al menos dos hermanos"

- **Axiomatización de la igualdad**
  - **Reflexiva**: $(\forall x)[x=x]$
  - **Simétrica**: $(\forall x,y)[x=y \Rightarrow y=x]$
  - **Transitiva**: $(\forall x,y,z)[[x=y] \wedge [y=z] \Rightarrow x=z]$
  - **Reglas de sustitución**: Los Igual pueden ser sustituidos por Igual en relaciones y funciones.

    $(\forall x,y)[x=y \Rightarrow f_i^{(1)}(x)=f_i^{(1)}(y)];\ i=1,2,...$

    $(\forall x,y)[x=y \Rightarrow P_i^{(1)}(x)=P_i^{(1)}(y)];\ i=1,2,...$

    $(\forall x,y,z,t)\ [(x=z) \wedge (y=t) \Rightarrow f_i^{(2)}(x,y)=f_i^{(2)}(z,t)];\ i=1,2,...$

    $(\forall x,y,z,t)\ [(x=z) \wedge (y=t) \Rightarrow P_i^{(2)}(x,y)=P_i^{(2)}(z,t)];\ i=1,2,...$

    ......

# Paramodulación

- En dominios complejos **no es práctico** especificar todos los axiomas que corresponden a **reglas de sustitución** para igualdad.
- **Paramodulación**: **Regla de inferencia específica** en BCs que incluye el **predicado igualdad.** Con una pequeña extensión, usada **en combinación con refutación por resolución,** forma un conjunto completo de reglas de inferencia

  Dados $K_1=\{\lambda(\tau)\cup K_1'\}$; $K_2=\{(\alpha=\beta)\cup K_2'\}$, donde $\alpha,\beta,\tau$ son términos; $K_1, K_2, K_1', K_2'$ son cláusulas;

  $\lambda(\tau)$ es un literal que contiene (entre otros), al término $\tau$.

  Si $\mu$ es el unificador más general de $\alpha,\tau$, entonces se infiere el **paramodulador** binario

  $\lambda\mu(\beta\mu) \cup K_1'\mu \cup K_2'\mu$

  donde $\lambda\mu(\beta\mu)$ es el resultado de reemplazar una sola ocurrencia de $\tau\mu$ en $\lambda\mu$ por $\beta\mu$.

  [Regla simétrica con los roles de $\alpha,\beta$ dados la vuelta]

Ejemplo 1: ¿Se cumple $P(A)\wedge(A=B) \models P(B)$?

Paramodulante[P(A),(A=B)] = P(B)

Ejemplo 2:

Paramodulante[P(g(f(x)))∨Q(x),(f(g(B))=A)∨R(g(C))] =

{P(g(A)) ∨ Q(g(B)) ∨ R(g(C)), P(g(f(A))) ∨ Q(f(g(B))) ∨ R(g(C))}

}

# Evaluación de expresiones

- Para algunos predicados y funciones en vez de usar inferencia es más sencillo y eficiente **evaluar expresiones** que contengan **términos base**.
  - Predicado de igualdad:

    Evaluar `¬(2222=4444)`
  - Comparaciones Booleanas:

    Evaluar `(2222<=4444)`
  - Funciones aritméticas:

    Evaluar `(2222+4444)`

- En inferencia con fbfs en FNC
  - Cláusulas cuya evaluación es
    - `Falso` hacen que el proceso de inferencia termine (con una contradicción)
    - `Verdadero` pueden ser eliminadas de la Base de Conocimiento.
  - Literales cuya evaluación es
    - `Falso` pueden ser eliminados de la cláusula.
    - `Verdadero` nos permiten eliminar la cláusula de la Base de Conocimiento.