Departamento de Inteligencia Artificial
Escuela Técnica Superior de
Ingenieros Informáticos

Lógica
Tema 7: Lenguajes de Primer Orden

Profesor: Emilio Serrano
emilioserra@fi.upm.es

Bloque

- **Estructura de la asignatura**

Bloque 1
Lógica Proposicional

Bloque 2
Lógica de Primer Orden

Temas

- **Bloque II**

Tema 7
Lenguajes de primer orden: sintaxis y uso en la formalización de argumentos

Tema 8
Semántica formal: Funciones de verdad, tautologicidad, consecuencia lógica

Representación

Tema 9
Razonamiento semántico: definición de modelos y contra-modelos

Tema 10
Cálculo de deducción natural de primer orden

Razonamiento

Tema 11
Forma Normal de Skolem. Forma clausular

Tema 12
Sustitución y Unificación máximamente general

Tema 13
Cálculo de resolución con Unificación

Computación

Introducción

❑ **¿Lógica de proposiciones y lógica de predicados?**

Lógica proposicional:

- Estudio de la consecuencia. Razonamientos válidos y correctos.
- Estudio de los conjuntos de creencias consistentes.
- Sintaxis + Semántica.
- Inferencia.

Lógica de predicados (Lógica de Primer Orden). A lo anterior añade un aumento de la capacidad expresiva:

- **Se analizan los enunciados atómicos.**
- **Aparecen los cuantificadores.**

Introducción

❑ La Lógica de Primer Orden

- La lógica proposicional sólo puede representar hechos acerca del mundo:
  - Solo se consideran frases declarativas, verdaderas o falsas y sin ningún otro valor de verdad.
  - La asignación de valores de verdad se realiza sin consideraciones de contexto ni de la estructura interna de los enunciados simples.
- Ejemplo: *Todo natural es entero y 2 es natural, luego 2 es entero*.

  $p \wedge q \mid\!-\ r$ es un razonamiento válido, pero la validez del razonamiento depende de la estructura interna de las proposiciones.
- La lógica de primer orden proporciona **más expresividad al captar más detalles del lenguaje natural**:
  - **Considerar una estructura interna en los enunciados atómicos**. Se puede acceder a los elementos de la proposición.
  - **Considerar propiedades**.

Introducción

❑ Sistema formal de la lógica de primer orden

- Es un sistema lógico para inferir conclusiones a partir de premisas.
- Trabaja con problemas de razonamiento desde el punto de vista de la estructura.

Lenguaje formal: alfabeto + reglas para formación de fórmulas lógicas

Teoría semántica: relación entre el lenguaje y el conjunto de significados de una fórmula lógica (V o F).

Sistemas de deducción: métodos deductivos para determinar la validez de los razonamientos. Permiten obtener conclusiones utilizando reglas de inferencia.

Objetivo: Más interés en *Cómo* se razona, y menos en *Qué* se razona.

Alfabeto de la lógica de primer orden

- **La Lógica de Primer Orden – Idea intuitiva.**
  - La lógica de primer orden describe un mundo que consta de:
    - **objetos (términos)**
    - **propiedades (o predicados) de esos objetos**.
  - Entre los objetos, se verifican varias **relaciones** p.ej.
    Progenitor(Marcos, José).
  - Una **función** es una relación en la cual sólo hay un valor para un input dado.
  - Ejemplos
    Objetos: gente, casas, números, planetas,...
    Relaciones: Progenitor, Hermano-de, Mayor-que,...
    Propiedades: Rojo, Pequeño, Primo,...
    Funciones: padre-de, uno-más-que

Alfabeto de la lógica de primer orden

❑ La Lógica de Primer Orden

- La lógica de primer orden contiene a la proposicional, pero es más potente. Utiliza una clase de lenguajes que son conocidos como **lenguajes de primer orden**, introducidos por Frege en 1879. El alfabeto de estos lenguajes dispone de símbolos que permiten:

  - Representar **elementos arbitrarios** del dominio o universo del discurso, por medio de **símbolos de variable**.
  - Representar **elementos específicos** del universo del discurso, por medio de **símbolos de constante**.
  - Expresar **propiedades o relaciones entre los elementos** del universo del discurso, por medio de **símbolos de predicado**.
  - Representar **generadores de elementos** del universo del discurso a partir de uno o varios elementos de dicho universo, por medio de **símbolos de función**.
  - Expresar que **nos referimos a algunos o a todos los elementos** del universo del discurso, por medio de **símbolos de cuantificación o cuantificadores**.

  - **Se añaden a la lógica proposicional los componentes de los predicados y todo lo que deriva de ello: variables, constantes, funciones, cuantificadores.**

Alfabeto de la lógica de primer orden

❑ Alfabeto de LPO:

- El alfabeto de un lenguaje de primer orden consta de los siguientes símbolos:
  - las **conectivas** de la lógica proposicional *¬, →, ∧, ∨, ↔*.
  - los **símbolos de puntuación** ***(*** y ***)***. No son necesarios si se utiliza notación prefija.
  - los **símbolos de cuantificación** *∀* (universal) y *∃* (existencial).
  - un conjunto infinito numerable, $V = \{x, y, z, v, \ldots, x_1, y_1, z_1, v_1, \ldots, x_n, y_n, z_n, v_n, \ldots\}$, de **símbolos de variables**.
  - un conjunto numerable (posiblemente vacío), ***C***, de **símbolos de constante**: ***a, b, c,...***
  - un conjunto numerable (posiblemente vacío), ***F***, de **símbolos de función** y una función r_1 que asigna a cada símbolo de función un elemento de N^* llamado su aridad (que representa el número de argumentos).
    - ***f (_), g(_,_), . . . (aridad = n. de args: f /1, g/2)***
  - un conjunto numerable y no vacío, ***P***, de **símbolos de predicado** y una función r_2 que asigna a cada símbolo de predicado un elemento de N^* llamado su aridad (que representa el número de argumentos).
    - Las proposiciones son predicados sin argumentos.
- La elección de los conjuntos *C, F y P* proporciona un lenguaje específico de primer orden y viene determinada por la aplicación que se pretende.
- Supondremos que los conjuntos *V, C,F y P* son disjuntos dos a dos.
- Partiendo de un alfabeto el conjunto de **expresiones de un LPO** está formado por cualquier **concatenación (finita) de símbolos de su alfabeto**.

Sintaxis. Términos

❑ Sintaxis

- **Términos**. Son expresiones. Se denominan términos a:
  - **Constantes**. Una constante es un término. Representan un objeto concreto, siempre el mismo.
    Ej: *Juan, Mi_casa, a, b, 0, 1*
  - **Variables**. Una variable es un término. Representan objetos sin identificar. Referencia a un objeto concreto, según el contexto.
    Ej: *x, y, padre, hijo*
  - **Funciones**. Representan (implícitamente) un objeto concreto que está relacionado con n objetos que participan en la función. Referencia a un objeto concreto, de forma indirecta. Cada símbolo de función tiene asociado un entero (>1) denominado grado o aridad, que indica cuantos argumentos tomará el símbolo de función.
    - si $t_1, \ldots, t_n$ son términos y f es un símbolo de función con aridad $n \geq 1$, entonces $f(t_1, \ldots, t_n)$ es un término
    Ej: *hijo_de(Juan, Ana), coseno(45)*

Sintaxis. Fórmulas bien Formadas

❑ Fórmulas

- Los **predicados** representan una **propiedad de un término** o **relaciones entre varios términos**, y **se aplican sobre los términos para formar las fórmulas atómicas**.
- Las **fórmulas atómicas** son expresiones de la forma $P(t_1, t_2, \ldots, t_n)$, siendo P un símbolo de predicado de grado n y $t_1, t_2, \ldots, t_n$ términos.
- Las fórmulas atómicas **expresan relaciones** entre los objetos que denotan sus términos o **propiedades de términos**:
  - JEFE(Pedro, Luis) Pedro es el jefe de Luis
  - RESPETA(Luis, madre(Luis)) Luis respeta a su madre

- Las **fórmulas bien formadas (FBF's)** se definen inductivamente por:

  - 1. Una formula Atómica es una FBF.
  - 2. Si α es una FBF, $\neg\alpha$ es una FBF.
  - 3. Si α y β son FBF's $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \rightarrow \beta$, $\alpha \leftrightarrow \beta$ son FBF's .
  - 4. Si α es una FBF, $\forall x\alpha$ y $\exists x\alpha$ son FBF
  - 5. El conjunto de FBF's es el cierre transitivo del conjunto de fórmulas atómicas con las leyes 1), 2), 3) y 4)

Sintaxis. Literales y Precedencia

❑ Literales y Precedencia

- **Literal**. Un literal **es un átomo** o **la negación de un átomo**.

  Ej: p, $\neg p$, $Q(a, f(1))$, $\neg Q(a, f(1))$, $Cat(g(x, y))$, $\neg Cat(g(x, y))$

- **Precedencia**. Se usan las reglas de la lógica proposicional, pero hay que ocuparse de los cuantificadores.
  - Igual que la negación, **los cuantificadores tiene mayor precedencia que las otras conectivas**
    - $\exists xF \wedge G$ es lo mismo que $(\exists xF) \wedge G$
    - significa que la segunda x en $\exists xP(x) \wedge Q(x)$ no está cuantificada como la primera

Sintaxis. Cuantificadores y Alcance

❑ Cuantificadores y alcance.

- Sea la FBF $Qx\alpha$, con Q uno de $\forall$ o $\exists$. Se denominan:
  - **<u>cuantificador</u> (sobre *x*)**: ***Qx***
  - **<u>alcance del cuantificador</u>**: $\boldsymbol{\alpha}$

- **Cuantificador universal ("para todo"): ∀**

  $\forall x\, A$ es verdad para cualquier valor de x que aparezca en A

- **Cuantificador existencial ("existe"): ∃**

  $\exists x\, A$ existe al menos un objeto del dominio para el cual A es cierto

- Las **apariciones de variables** que están bajo el alcance de un cuantificador se dice que están **ligadas**, sino, se dice que están **libres**.

- Ejemplos de alcance del cuantificador.

$\forall x\, P(x) \rightarrow \forall y\, R(x, y)$
$\forall x\, P(x) \rightarrow \forall x\, R(x, y)$
$\forall x\, \forall y\, (P(x) \rightarrow R(x, y))$

Sintaxis

- <u>Ejemplo 1 de creación LPO</u>.

Juan y Pedro son amigos, aunque sólo Juan está casado.

  - La oración incluye dos individuos concretos, por lo que son necesarias dos constantes: *a, b* para representarlos en el lenguaje formal.
  - La oración afirma una relación de amistad entre ambos: por tanto es necesario un predicado con dos argumentos : *A(x,y)*.
  - La oración afirma una propiedad de Juan: estar casado: *C(x)*

Por tanto, el lenguaje formal necesario para formalizar la oración tiene como alfabeto:

$$\{ a, b, A(x,y), C(x) \}$$

Y la fórmula de ese lenguaje que recoge lo dicho en la oración es:

$$A(a,b) \wedge C(a) \wedge \neg C(b)$$

Sintaxis

- Ejemplo 2 de creación LPO.

Eva y María son amigas aunque sus maridos sean enemigos desde que el marido de Eva despidiese al de María.

- La oración habla de cuatro individuos. Dos de ellos son identificados por sus nombres propios y los otros dos se identifican *en función de* los anteriores. Por tanto, son necesarias *dos constantes: a, b* y una función *f() para representar "ser el marido de"*.

- La oración menciona una relación de amistad entre individuos: es necesario un predicado con dos argumentos : $A(x,y)$

- La oración menciona una relación en la que un individuo x despide a otro y: $D(x,y)$

Por tanto, el lenguaje formal necesario para formalizar la oración tiene como alfabeto:

$$\{ a, b, f(x), A(x,y), D(x,y) \}$$

Y la fórmula de ese lenguaje que recoge lo dicho en la oración es:

$$A(a,b) \wedge \neg A(f(a), f(b)) \wedge D(f(a), f(b))$$

Sintaxis

❑ Esquemas de formalización

Ejemplo 1: Universo de discurso: personajes de la novela de Chrétien de Troyes

a := Arturo
i := Ginebra
e := Lanzarote

$A(x,y)$:= x es amigo de y
$Q(x,y)$:= x ama a y
$O(x,y)$:= x odiar a y

Lanzarote ama a la reina Ginebra, pero ella no ama a todos los que la aman.

$$Q(e,i) \wedge \neg\forall x\,(Q(x,i) \rightarrow Q(i,x))$$

Lanzarote no ama a ninguno de sus amigos.

$$\forall x\,(A(x,e) \rightarrow \neg Q(e,x))$$

Los amigos de Lanzarote odian a aquellos a quienes Arturo ama.

$$\forall x\,(A(x,e) \rightarrow \forall y\,(Q(a,y) \rightarrow O(x,y)))$$

Ejemplo 2:

Todos los hombres son mortales

$$\forall x(H(x) \rightarrow M(x))$$

Los hombres son mamíferos bípedos

$$\forall x(H(x) \rightarrow (M(x) \wedge B(x)))$$

Todos los satélites de Júpiter son rocosos

$$\forall x(S(x,a) \rightarrow R(x))$$

…en relación a los universos de discurso (divulgación)

- ¿Podemos decir que las siguientes sentencias son verdaderas o falsas?
  - Si te muerde un zombi te conviertes en zombi
  - Harry Potter es un mago
  - El rey de Francia es calvo

[Nonexistent Objects & Imaginary Worlds]

Sintaxis

❑ Ejercicios

<u>Ejercicio 1</u>: Seleccionar las constantes, funciones y predicados necesarios para definir un LPO en el que formalizar las siguientes oraciones:

1. Mi casa es roja

R(x)	*x es roja*
a	*Mi casa*

R(a)

2. Luisa y María son brasileñas pero Vicente es mejicano

B(x)	*x es brasileño*
M(x)	*x es mexicano*
a	*Luisa*
b	*María*
c	*Vicente*

B(a)* ∧ *B(b)* ∧ *M(c)

3. Jorge adora a Juan

A(x,y)	*x adora a y*
a	*Jorge*
b	*Juan*

A(a,b)

4. Juan ama a Rosa pero ella no le corresponde

A(x,y)	*x ama a y*
a	*Juan*
b	*Rosa*

A(a,b)* ∧ ¬ *A(b,a)

Sintaxis

❑ Ejercicios

Ejercicio 1: Seleccionar las constantes, funciones y predicados necesarios para definir un LPO en el que formalizar las siguientes oraciones:

5. Pedro sujetó a Juan y María le atizó

S(x,y)	x sujeta a y
A(x,y)	x atiza a y
a	Pedro
b	Juan
c	María

S(a,b) ∧ A(c,b)

6. Nieves se peina a sí misma y también peina a Juan

P(x,y)	*x peina a y*
a	*Nieves*
b	*Juan*

P(a,a) ∧ P(a,b)

7. Si Colón descubrió América, merece un lugar en la Historia

D(x,y) x descubre y

L(x) x merecer lugar en la historia

a	*Colón*
b	*América*

D(a,b) → L(a)

8. El asesino de mi padre es Juan o Pedro, pero no Alberto

A(x, y)	*x asesina a y*		
Juan	*a*	*Mi padre*	*d*
Pedro	*b*		
Alberto	*c*		

(A(a,d) ∨ A(b,d))∧ ¬A(c,d)

Sintaxis

❑ Ejercicios

Ejercicio 2: Seleccionar las constantes, funciones y predicados necesarios para definir un LPO en el que formalizar las siguientes oraciones:

1. Hay al menos un número primo $\exists x P(x)$
2. Algunas cantantes de ópera no están gordas $\exists x\, (C(x) \wedge \neg G(x))$
3. Cualquier crimen será castigado $\forall x\, (C(x) \rightarrow P(x))$
4. No todos los crímenes merecen la pena capital $\neg \forall x\, (C(x) \rightarrow P(x))$ $\exists x\, (C(x) \wedge \neg P(x))$
5. Hay profesores que no saben explicar $\exists x\, (P(x) \wedge \neg E(x))$
6. Sólo los suecos entienden a Bergman $\forall x\, (E(x,a) \rightarrow S(x))$
7. Hay genios, pero no todos los informáticos lo son $\exists x\, G(x) \wedge \neg \forall x\, (I(x) \rightarrow G(x))$

Sintaxis. Variables Libres y Ligadas

- **Variables**. Dada una fbf ***(Qx)A***, decimos que ***x*** es la variable del cuantificador y que ***A*** es el rango o alcance del cuantificador (o de la variable cuantificada). Es decir, el rango de un cuantificador es la fbf a la que se aplica.
- **Variables libres y ligadas**. En una fórmula de un LPO:
  - Una **variable** se encuentra **ligada** en una fórmula **cuando esta bajo el alcance de un cuantificador ($\forall$*x* o $\exists$*x*)**.
  - Una **variable** se encuentra **libre** en una fórmula **cuando no se encuentra ligada**.
  - Un mismo símbolo de variable puede estar libre y ligado en una misma fórmula.

Ejemplos:

- En $\exists xp(x)$ la ocurrencia de la variable *x* está ligada.
- En $\exists xp(x, y)$ la ocurrencia de la variable *x* está ligada mientras que la ocurrencia de la variable *y* está libre.
- En $p(x,y)$ la ocurrencia de la variable *x* está libre y la ocurrencia de la variable *y* está libre.
- En $\forall xp(x,y)$^$q(x)$ la primera ocurrencia de la variable *x* está ligada mientras que la segunda ocurrencia de la variable *x* está libre. La (única) ocurrencia de la variable *y* está libre.

Sintaxis. Fórmulas Abiertas y Cerradas

- **Fórmulas abiertas y cerradas:**
  - Una **fórmula es abierta cuando contiene al menos una variable libre**.
  - Una **fórmula es cerrada si todas sus variables están ligadas**.

    Ejemplos:
    - $\forall x\ \forall y\ (P(x) \rightarrow R(x, y))$ FORMULA CERRADA
    - $\forall x\ P(x) \rightarrow \forall x\ R(x, y)$ FORMULA NO CERRADA (la variable y es libre)

  - El cierre de una fórmula es una fórmula cerrada.

- La clausura o cierre universal de una fórmula $A(x_1, x_2, \ldots, x_n)$ es la sentencia

$$\forall x_1\ \forall x_2 \ldots \forall x_n\ A(x_1, x_2, \ldots, x_n)$$

- La clausura o cierre existencial de una fórmula $A(x_1, x_2, \ldots, x_n)$ es la sentencia

$$\exists x_1\ \exists x_2 \ldots \exists x_n\ A(x_1, x_2, \ldots, x_n)$$

- Ejemplo:

$$Cierre_{\forall}(p(x) \vee \exists y q(x, z, y)) = \forall x \forall z (p(x) \vee \exists y q(x, z, y))$$

Sintaxis

❑ Ejercicios

Ejercicio 3: Señalar las variables libres y ligadas en las siguientes fórmulas:

1. $\exists x(P(x,f(y)) \rightarrow \exists yQ(x,y))$
2. $\exists xP(x) \rightarrow \forall yQ(x,f(y))$
3. $\exists x\exists y(P(x,y) \vee Q(x,y)) \wedge R(a,y)$
4. $\exists x\exists y((P(x,y) \vee Q(x,y)) \wedge R(x,y))$
5. $\forall x(x=y \rightarrow \exists zP(x,z))$
6. $\exists x\forall yP(x,f(x,y)) \rightarrow \exists yQ(x,y)$
7. $x = y + z \rightarrow x \leq y + z$
8. $\forall x(x + 0 = x)$

Sintaxis. Sustituciones

❑ Sustituciones

- Trataremos las fórmulas como secuencias de símbolos.
- Se utilizan las sustituciones para:
  - o definir la semántica del lenguaje ($\forall$, $\exists$)
  - o en la lógica computacional (instancias, unificación, resolución).
- **Sustitución de variables**. Es una operación sintáctica sobre fórmulas y términos que se aplica **solamente sobre variables libres**. **Asocia variables con términos**.
  - o $A(x)$ inicia la aparición de al menos una ocurrencia libre de x en A.
  - o Dada una fórmula A, la expresión $\boldsymbol{A\{x/t\}}$ denota la fórmula obtenida al **sustituir todas las ocurrencias libres de la variable *x* en *A* por el término *t*** (por el que es sustituible).
  - o En general, $A\{x_1/t_1; : : : ; x_n/t_n\}$, donde **las sustituciones son simultáneas**.

Ejemplos:

Si A denota a la fórmula P(x), entonces $A\{x/b\}$ denota a la fórmula $P(b)$.

Si A denota a $P(x,f(y)) \rightarrow \exists y Q(x,y)$, entonces $A\{x/a\}$: $P(a,f(y)) \rightarrow \exists y Q(a,y)$.

Si $A(y)$: $\exists x((P(x,y) \vee Q(x,y)) \wedge R(x,y)$, entonces $A\{y/f(z)\}$: $\exists x((P(x, f(z)) \vee Q(x, f(z))) \wedge R(x,f(z))$

Sintaxis. Sustituciones

❑ Sustituciones

- En general, una función f se puede escribir como un conjunto de "pares".
  - Ejemplo: $f(x) = 2x$

    {(1, 2), (2, 4), (3, 6), (4, 8), ...}

    {1/2, 2/4, 3/6, 4/8, ...}

- **Terminología**:
  - **ligadura**: un par x_i/t_i
  - **Dominio (α)** = $\{ x \mid \exists t(x/t \in \alpha) \}$
    - (variables que aparecen en el primer elemento de los pares de una sustitución)
  - **Rango (α)** = $\{ y \mid \exists t(\exists x(x/t \in \alpha) \wedge y$ aparece en $t) \}$
    - (variables que aparecen en el segundo elemento de los pares de una sustitución)
  - λ = {} (sustitución vacía, no hace nada)

Sintaxis

❑ Sustituciones

- Ejemplos:
  - α_1 = { x/f (a), y/x, z/h(b, y), w/a }
    Dominio (α_1) = {x, y, z, w}
    Rango (α_1) = {x, y}
  - α_2 = { x/a, y/a, z/h(b, c), w/f (d) }
    Dominio (α_2) = {x, y, z, w}
    Rango (α_2) = {}
  - α_3 = { x/y, z/w }
    Dominio (α_3) = {x, z}
    Rango (α_3) = {y, w}
  - λ = {}
    Dominio (λ) = {}
    Rango (λ) = {}

Sintaxis. Sustituciones

Composición de sustituciones

- Dadas $\alpha = \{x_1/t_1, .., x_n/t_n\}$ y $\beta = \{y_1/s_1, .., y_m/s_m\}$, la **composición** $\boldsymbol{\alpha\beta}$ de estas sustituciones se define como el conjunto

$$\{ x_1/(t_1\beta), .., x_n/(t_n\beta), y_1/s_1, .., y_m/s_m\}$$

del que se eliminan los elementos tales que

$$x_i \equiv t_i\beta;$$
$$y_j \in \{x_1, .., x_n\}$$

- Ejemplos

$$\alpha = \{ x/3, y/f(x, 1) \} ; \qquad \beta = \{ x/4 \}$$
$$\alpha\beta = \{ x/3, y/f(4, 1) \}$$
$$\beta\alpha = \{ x/4, y/f(x, 1) \}$$

- **Propiedades:** Para *F*,

$$(\alpha\beta)\gamma = \alpha(\beta\gamma)$$
$$\alpha\lambda = \lambda\alpha = \alpha$$
$$\alpha\beta \neq \beta\alpha$$
$$(F\alpha)\beta = F(\alpha\beta)$$

Sintaxis. Sustituciones

❑ Sustituciones

- En general, si sustituyo una variable x por un término t en una fórmula, la afirmación que hacía la fórmula sobre x, ahora la hace sobre t, pero no siempre ocurre esto.

  Ejemplo: $\exists y\ x = 2*y$ dice "x es par".

  Sustituyendo x por y se cambia el significado:
  $\exists y\ y = 2*y$ dice "existe un número que es el doble de sí mismo."

  Sustituyendo x por $z + 1$, se dice de $z + 1$ lo mismo que se decía de x:
  $\exists y\ z + 1 = 2*y$ dice "$z + 1$ es par" (o sea, "z es impar")

  Sustituyendo x por $x + 1$, se dice de $x + 1$ lo mismo que se decía de x:
  $\exists y\ x + 1 = 2*y$ dice "$x + 1$ es par" (o sea, "x es impar")

- Una **variable x es sustituible por un término t en una fórmula A si ninguna ocurrencia de alguna variable z que aparece libre en el término t queda ligada al hacer la sustitución**.

  Ejemplos: $A \equiv (\exists z\ p(x,z) \wedge q(y))$
  Si $t \equiv f(z)$, entonces x NO es sustituible por t en A.
  Si $t \equiv f(y)$, entonces x ES sustituible por t en A.
  Si $t \equiv f(x)$, entonces x ES sustituible por t en A.

Sintaxis

❑ Sustituciones

- De otra manera:

Una sustitución α NO se puede aplicar a una fórmula F si pasa lo siguiente:

- o α contiene una ligadura x/t y t contiene la variable z
- o hay una ocurrencia de x en F que se puede reemplazar por t según las reglas de la aplicación libre, pero dicha ocurrencia está dentro del ámbito de un cuantificador sobre z

Ejemplo:

$$\exists z q(x, z)\{x/f(z)\} = \exists z q(f(z), z)$$

no se puede hacer

Similar a la visibilidad de las variables en un lenguaje de programación:

```
for (int i = 0; i < 100; i++) {... int i=4;...}
```

Uso en la formalización de conocimiento

- **Resumiendo**:
  - Sólo sustituiremos ocurrencias **libres** de las variables
  - Las ocurrencias de variables que aporte cada término sustituyente deben resultar libres en la fórmula final.

- **Cierre extraño**:
  - Ejemplo:

    $Cierre_{\exists}(\forall x p(x) \wedge q(x, y)) = \exists x \exists y(\forall x p(x) \wedge q(x, y))$

    ☞ hay dos cuantificadores sobre x cuyo ámbito se solapa

    Es algo que se tiene que evitar (p. ej. renombrando).

- **Situación imposible**:
  - Ejemplo:

    $\exists z q(x, z)\{x/f(z)\} = \exists z q(f(z), z)$ $\qquad \exists z' q(x, z')\{x/f(z)\}$

    Hay confusión entre variables que conceptualmente son distintas pero tienen el mismo nombre.
    La mejor solución es renombrar las ocurrencias ligadas de variables.

Sintaxis

❑ Ejemplos

Realizar las siguientes sustituciones:

1. $(\ \exists x(P(x,f(y)) \rightarrow \exists y Q(x,y))\ )\{y/g(z)\}$

   $\exists x(P(x,f(g(z))) \rightarrow \exists y Q(x,y))$

2. $(\ \forall x \forall y(P(x,y) \rightarrow Q(x,y))\ )\{y/a\}$

   $\forall x \forall y(P(x,y) \rightarrow Q(x, y))$

3. $(\ \forall x(\forall y P(x,y) \rightarrow Q(x,y))\ )\{y/a\}$

   $\forall x(\forall y P(x,y) \rightarrow Q(x,a))$

4. $(\ \exists x(\forall y(P(x,y) \vee Q(x,y)) \wedge R(x,y)\ )\{y/b\}$

   $\exists x(\forall y(P(x,y) \vee Q(x,y)) \wedge R(x,b)$

Sintaxis

❑ Ejercicios

Ejercicio 4: Realizar las siguientes sustituciones:

1. $(\ \exists x(\forall y(P(x,y) \vee Q(x,y))) \wedge R(x,y)\ )\{x/b\}$
2. $(\ \exists x \forall y(P(x,y) \vee Q(x,y)) \wedge R(x,y)\ )\{x/a,\ y/b\}$
3. $(\ \forall x(P(x,y) \rightarrow Q(x,y))\ )\{y/f(x,a)\}$
4. $(\forall y P(x,y) \rightarrow \forall x Q(x,y)\ )\{y/f(x,a)\}$
5. $(\ x = y + z \rightarrow x \leq y + z\ )\{x/1,\ y/2\}$
6. $(\ x = y + z \rightarrow x \leq y + z\ )\{x/s(x)\}$
7. $(\ x = y + z \rightarrow x \leq y + z\ )\{x/s(y)\}$
8. $(\ \forall x(\ x + 0 = x)\ )\{x/1\}$

Uso en la formalización de conocimiento

- Ejemplos de formalización de argumentos:
  - Ejemplo 1:

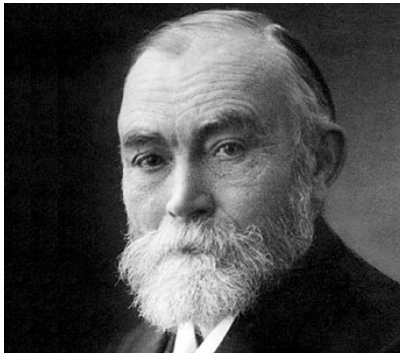
La Tierra orbita en torno al Sol. La Luna orbita en torno a la Tierra. Todo cuerpo que orbita en torno al Sol es un planeta. Son satélites los cuerpos que orbita en torno a planetas. Luego la Tierra es un planeta y la Luna un satélite.

a: la Tierra
b: el Sol
c: la luna
O(x,y): x orbita en torno a y
P(x): x es un planeta
S(x): x es un satélite

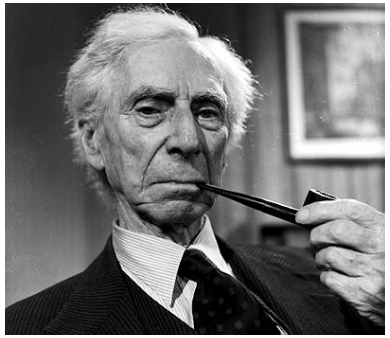
$O(a,b)$
$O(c,a)$
$\forall x(O(x,b) \rightarrow P(x))$
$\forall x \forall y(O(x,y) \land P(y) \rightarrow S(x))$
$\vdash P(a) \land S(c)$

La paradoja de Russell (divulgación)

Gottlob Frege,
1848-1925



Bertrand Russell,
1872- 1970



$$\forall x \qquad \text{afeita}(x, barbero) \iff \neg\text{afeita}(x, x)$$
$$\text{afeita}(barbero, barbero) \iff \neg\text{afeita}(barbero, barbero)$$

Estructuras básicas

- Todos los hombres son mortales
  - $\forall x(H(x) \rightarrow M(x))$ o $\neg\exists x(H(x) \wedge \neg M(x))$
- No todo hombre es mortal
  - $\neg \forall x(H(x) \rightarrow M(x))$ o $\exists x(H(x) \wedge \neg M(x))$
- Al menos un hombre es mortal, hay hombres mortales
  - $\exists x(H(x) \wedge M(x))$
- Ningún hombre es mortal
  - $\neg \exists x(H(x) \wedge M(x))$ o $\forall x(H(x) \rightarrow \neg M(x))$
- Sólo los hombres son mortales
  - $\forall x(M(x) \rightarrow H(x))$ o $\neg\exists x(M(x) \wedge \neg H(x))$
- (Típicamente, $\forall$ se usa con implicación y $\exists$ con conjunción).

Uso en la formalización de conocimiento

- Ejemplo 2. Definir LPOs en los que formalizar los siguientes argumentos:

Aquel que no existe no puede engañarse. Yo me engaño. Luego yo existo.

$$\{ \forall x(\neg E(x) \to \neg P(x)), P(a) \} \vdash E(a)$$

Solamente las personas bien educadas están suscritas al Times. Ningún puercoespín sabe leer. Las personas bien educadas saben leer. Luego ningún puercoespín está suscrito al Times.

$$\{ \forall x(T(x) \to B(x)), \forall x(P(x) \to \neg L(x)), \forall x(B(x) \to L(x)) \} \vdash \forall x(P(x) \to \neg T(x))$$

$$\{ \forall x(T(x) \to B(x)), \neg \exists x(P(x) \wedge L(x)), \forall x(B(x) \to L(x)) \} \vdash \neg \exists x(P(x) \wedge T(x))$$

Todos los filósofos se han preguntado qué es la Filosofía. Todos los que se han preguntado qué es la Filosofía han dado en la locura. Nietzsche es un filósofo. El Padre Ceballos no acabó loco. Luego Nietzsche y el Padre Ceballos no son la misma persona.

$$\{ \forall x(F(x) \to P(x)), \forall x(P(x) \to L(x)), F(a), \neg L(b) \} \vdash \neg(a = b)$$

Formalización de argumentos

Definir LPOs en los que formalizar los siguientes argumentos:

1. *Todos los libros de texto son tediosos*

   Algunos libros de texto están llenos de ejercicios

   ∴ Algunos libros llenos de ejercicios son tediosos

   $\forall x(Tx \rightarrow Dx)$
   $\exists x(Tx \wedge Ex)$
   $\exists x(Ex \wedge Dx)$

2. *Todos los políticos tienen algo que ocultar*
   Algunos políticos salen en televisión

   ∴ Hay quienes tienen algo que ocultar y salen en televisión

   $\forall x(Px \rightarrow Ox)$
   $\exists x(Px \wedge Tx)$
   $\exists x(Ox \wedge Tx)$

3. *Todos los chimpancés son primates*

   Sara es un chimpancé

   ∴ Sara es un primate

   $\forall x(Cx \rightarrow Px)$
   Ca
   Pa

4. *Algunos primates no son chimpancés*

   Algunos chimpancés saben hablar

   ∴ Algunos primates no saben hablar

   $\exists x(Px \wedge \neg Cx)$
   $\exists x(Cx \wedge Hx)$
   $\exists x(Px \wedge \neg Hx)$

5. *Ningún individuo que no sea chimpancé es un primate*
   Ninguno que no sea primate es inteligente
   Sara no es un chimpancé

   ∴ Sara no es inteligente

   $\forall x(\neg Cx \rightarrow \neg Px)$
   ...o $\neg \exists x(\neg Cx \wedge Px)$
   $\forall x(\neg Px \rightarrow \neg Ix)$
   ...o $\neg \exists x(\neg Px \wedge Ix)$
   $\neg Ca$
   $\neg Ia$

Formalización de argumentos II

Definir LPOs en los que formalizar los siguientes argumentos:

1. *Hay individuos inteligentes o que saben hablar. Juan no sabe hablar. Luego Juan no es inteligente.*
2. *Todo elemento químico es oxidante o reductor. El carbono es un elemento químico no oxidante. Luego el carbono es reductor.*
3. *No todos los seres humanos saben hablar o son inteligentes. Sara es un ser humano pero no sabe hablar. En consecuencia, Sara es inteligente.*
4. *Todos los chimpancés saben hablar. Algunos primates no saben hablar. Algunos primates son humanos. Por tanto, algunos seres humanos son chimpancés y saben hablar.*
5. *Todos los chimpancés son primates. Algunos seres humanos son inteligentes. Algunos primates son seres humanos. Juan es un chimpancé y Sara es un ser humano que sabe hablar. Así pues, Juan es un primate y Sara es inteligente.*
6. *Todos los rinocerontes tienen un cuerno. Todos y sólo los rinocerontes son dignos de ser cazados. Luego todos los animales dignos de ser cazados tienen un cuerno.*
7. *Todas las selvas tropicales tienen color verde. Nada que tenga color verde esta seco. Por tanto, ninguna selva tropical esta seca.*

Formalización de argumentos II (soluciones)

- *Hay individuos inteligentes o que saben hablar. Juan no sabe hablar. Luego Juan no es inteligente.*

$$\{ \exists x(I(x) \vee S(x)), \neg S(a) \} \vdash \neg I(a)$$

- *Todo elemento químico es oxidante o reductor. El carbono es un elemento químico no oxidante. Luego el carbono es reductor.*

$$\{ \forall x(E(x) \rightarrow (O(x) \vee R(x)), E(c) \wedge \neg O(c) \} \vdash R(c)$$

- *No todos los seres humanos saben hablar o son inteligentes. Sara es un ser humano pero no sabe hablar. En consecuencia, Sara es inteligente.*

$$\{\neg \forall x(H(x) \rightarrow S(x) \vee I(x)), H(a) \wedge \neg S(a) \} \models I(a)$$

- *Todos los chimpancés saben hablar. Algunos primates no saben hablar. Algunos primates son humanos. Por tanto, algunos seres humanos son chimpancés y saben hablar.*

$$\{ \forall x(C(x) \rightarrow S(x)), \exists x(P(x) \wedge \neg S(x)), \exists x(P(x) \wedge H(x)) \} \mid\!- \exists x(H(x) \wedge C(x) \wedge S(x))$$

- *Todos los chimpancés son primates. Algunos seres humanos son inteligentes. Algunos primates son seres humanos. Juan es un chimpancé y Sara es un ser humano que sabe hablar. Así pues, Juan es un primate y Sara es inteligente.*

$$\{ \forall x(C(x) \rightarrow P(x)), \exists x(H(x) \wedge I(x)), \exists x(P(x) \wedge H(x)), C(a) \wedge H(b) \wedge S(b) \} \mid\!- P(a) \wedge I(b)$$

- *Todos los rinocerontes tienen un cuerno. Todos y sólo los rinocerontes son dignos de ser cazados. Luego todos los animales dignos de ser cazados tienen un cuerno.*

$$\{ \forall x(R(x) \rightarrow T(x)), \forall x(R(x) \leftrightarrow D(x)) \} \mid\!- \forall x(D(x) \rightarrow T(x))$$

- *Todas las selvas tropicales tienen color verde. Nada que tenga color verde esta seco. Por tanto, ninguna selva tropical esta seca.*

$$\{ \forall x(T(x) \rightarrow V(x)), \forall x(V(x) \rightarrow \neg S(x)) \} \mid\!- \forall x(T(x) \rightarrow \neg S(x))$$

o

$$\{ \forall x(T(x) \rightarrow V(x)), \neg \exists x(V(x) \wedge S(x)) \} \mid\!- \neg \exists x(T(x) \wedge S(x))$$

Lógica
Tema 8: Semántica Formal

Profesor: Emilio Serrano
emilioserra@fi.upm.es

Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingenieros Informáticos

Temas

❑ Bloque II

Representación

- **Tema 7** — Lenguajes de primer orden: sintaxis y uso en la formalización de argumentos
- **Tema 8** — Semántica formal: Funciones de verdad, tautologicidad, consecuencia lógica

Razonamiento

- **Tema 9** — Razonamiento semántico: definición de modelos y contra-modelos
- **Tema 10** — Cálculo de deducción natural de primer orden

Computación

- **Tema 11** — Forma Normal de Skolem. Forma clausular
- **Tema 12** — Sustitución y Unificación máximamente general
- **Tema 13** — Cálculo de resolución con Unificación

Índice

Introducción a la semántica formal

- **¿Qué es la semántica?**

- Estudia el **significado de los símbolos**, por lo que se introduce el concepto de **interpretación** (conjunto de reglas precisas que permiten asignar objetos de un dominio a ciertas expresiones de un lenguaje formal).

- **Asigna un significado a las construcciones sintácticas**. Junto con la sintáctica ayuda a definir un sistema formal.

Introducción a la semántica formal

- En general, dar significado a las fórmulas (interpretar) de un lenguaje formal consiste en definir una función: *FBF(L)* $\Rightarrow$ *{ V, F}*

- Le **asigna el "valor de verdad" a las f.b.f.** (decir si son verdaderas o falsas).

- Composicionalidad: el significado de una expresión compleja depende de los significados de sus expresiones componentes.

- Cálculo de la función FBF(L):
  - Asignación de valor de verdad a todas las fórmulas atómicas de L:
    - Función de valoración (v): *FA(L)* $\Rightarrow$ *{ V, F }*.
      - Ej. *v(p) = V ; v(q) = F*

Introducción a la semántica formal

- Asignación de valor de verdad a todas las fórmulas moleculares de L:
  - Cada conectiva esta completamente definida por su función de verdad:
    - $fv_{\neg}(F) = V$; $fv_{\neg}(V) = F$
    - $fv_{\wedge}(V,V) = V$; $fv_{\wedge}(V,F) = fv_{\wedge}(F,V) = fv_{\wedge}(F,F) = F$
    - $fv_{\vee}(F,F) = F$; $fv_{\vee}(V,F) = fv_{\vee}(F,V) = fv_{\vee}(V,V) = V$
    - $fv_{\rightarrow}(V,F) = F$; $fv_{\rightarrow}(V,V) = fv_{\rightarrow}(F,V) = fv_{\rightarrow}(F,F) = V$
    - $fv_{\leftrightarrow}(V,F) = fv_{\leftrightarrow}(F,V) = F$; $fv_{\leftrightarrow}(V,V) = fv_{\leftrightarrow}(F,F) = V$
  - El valor de verdad de una fórmula molecular es función de su conectiva principal.
  - Ampliación de la función de valoración (*v*) a fórmulas moleculares: $FM(L) \Rightarrow \{V, F\}$
    - $v(\neg A) = fv_{\neg}(v(A))$
    - $v(A \wedge B) = fv_{\wedge}(v(A),v(B))$
    - $v(A \vee B) = fv_{\vee}(v(A),v(B))$
    - $v(A \rightarrow B) = fv_{\rightarrow}(v(A),v(B))$
    - $v(A \leftrightarrow B) = fv_{\leftrightarrow}(v(A),v(B))$

Interpretaciones

❑ **Objetivo de la semántica formal.**

- Dado un LPO, definir de modo preciso el **significado de sus fórmulas**.

❑ **Conceptos semánticos usados en semántica formal.**

- **Dominio de interpretación *D*** :
  - o Dominio: conjunto no vacío de objetos
  - o Relaciones n-arias: subconjuntos de Dominio n
  - o Funciones n-arias: n-tuplas de objetos del dominio ↦ objetos del dominio
- **Función de interpretación *i()*** :
  - o Fórmulas ↦{V, F}
  - o Términos ↦ objetos del dominio
  - o Predicados y funciones ↦ relaciones y funciones sobre objetos del dominio
- **Interpretación I**: *<D, i()>*
  - o Un dominio no vacío de individuos, D
  - o Una función i() de individuos de D, funciones y relaciones sobre D a todas las constantes, funciones y predicados del LPO.
- **Modelo**: tipo particular de interpretación en la que:
  - o Las premisas de una teoría T son verdaderas: T ↦ {V}

Interpretaciones

- **Dominios de interpretación:**
  - Idea de partida: las expresiones de un lenguaje significan cuando *refieren* a algo distinto de ellas mismas, cuando *hablan de* algo: un dominio, un universo de discurso.
  - Por tanto, el punto de partida para dar significado a las fórmulas de un LPO es la elección de un **dominio de interpretación**:
    - **Dominio = Conjunto no vacío de individuos**
    - Ejemplos:
      - D = {Sol, Tierra, Luna}
      - D = N
      - D = {♦, ❍, ❑ }
      - Cualquier conjunto bien definido de objetos es aceptable como dominio de interpretación.

Interpretaciones

- **Función de Interpretación para un lenguaje L:**
  - Una vez escogido un dominio D, definimos sobre él una función de interpretación *i()* que asigna significado al vocabulario de un LPO:

    - Toda **constante** $a_i \in L$: $i(a_i) = d_i \in D$
      - Todo elemento de *D* tiene un nombre (distinto) en *L* : para todo $d \in D$ existe una constante $a \in L$ tal que $i(a) = d$.
      - Si *L* no posee de partida suficientes constantes, **se amplía** para cumplir este requisito: *L(D)* es idéntico a *L* excepto por la incorporación de tantas constantes como sea necesario para nombrar a todos los individuos de *D*.
    - Toda **función** $f_n \in L$: $i(f_n) = f_D$
      - f_D: $\langle d_1,\ldots,d_n \rangle \Rightarrow d$ (aplicación de D^n en *D*)
      - Un símbolo de función significa una aplicación de elementos de *D* en *D*.
    - Todo **predicado** n-ario $P \in L$: $i(P) = P_D$
      - P_D: $\langle d_1,\ldots,d_n \rangle \Rightarrow \{V, F\}$ (aplicación de D^n en {V, F})

Interpretaciones

- **Interpretaciones:**
  - Definimos una **interpretación** ***I*** como un par ***<D, i()>***:
    - Un conjunto no vacío de individuos, ***D***
    - Una función ***i()*** que asocia individuos de *D*, funciones y relaciones sobre *D* con constantes, funciones y predicados de *L*, respectivamente.
  - Asignar significado a las fórmulas de *L* implica:
    1. Asignar significado a las fórmulas atómicas
    2. Asignar significado al resto de fórmulas mediante inducción

Interpretaciones

- **1.- Interpretación de fórmulas atómicas básicas:**
  - Asignación de valor de verdad a las fórmulas **atómicas** básicas de *L*:

    $P_n(t_1,\ldots,t_n)$ es una **fórmula atómica básica sii** $t_1,\ldots,t_n$ **no contienen variables**.

    $$\mathbf{i(P_n(t_1,\ldots,t_n)) = V/F \quad sii \quad P_{nD}(i(t_1),\ldots,i(t_n)) = V/F}$$

  - Cada interpretación concreta asignará un y sólo un valor de verdad a cada fórmula atómica básica de *L*. Dos interpretaciones con igual dominio e idénticas asignaciones al vocabulario, pueden diferir en el valor de verdad que asignan a sus fórmulas atómicas básicas.
  - En el caso del predicado *=*, su semántica es fija, sin variación entre interpretaciones:

    $i(t_1=t_2) = V/F$ sii $i(t_1)$ es idéntico a / no es idéntico a $i(t_2)$

Interpretaciones

- **2.- Interpretación de fórmulas moleculares:**
  - Asignación de valor de verdad a las fórmulas moleculares:
    1. ***i(¬A) = V*** *sii i(A) = F;* ***i(¬A) = F*** *sii i(A) = V*
    2. ***i(A ∧B) = V*** *sii i(A) = V* y *i(B) = V;* ***i(A ∧B) = F*** *sii i(A) = F* o *i(B) = F*
    3. ***i(A ∨B) = V*** *sii i(A) = V* o *i(B) = V;* ***i(A ∨B) = F*** *sii i(A) = F* y *i(B) = F*
    4. ***i(A →B) = V*** *sii i(A) = F* o *i(B) = V;* ***i(A →B) = F*** *sii i(A) = V* y *i(B) = F*
    5. ***i(A ↔B) = V*** *sii i(A) = i(B);* ***i(A ↔B) = F*** *sii i(A) ≠ i(B);*
    6. ***i(∃xA) = V*** *sii i(A{x/a}) = V* para al menos una sustitución *x/a,* ***a*** *∈L(D)*
    7. ***i(∃xA) = F*** *sii i(A{x/a}) = F* para toda sustitución *x/a*, ***a*** constante de *L(D)*
    8. ***i(∀xA) = V*** *sii i(A{x/a}) = V* para toda sustitución *x/a*, ***a*** constante de *L(D)*
    9. ***i(∀xA) = F*** *sii i(A{x/a}) = F* para al menos una sustitución *x/a*, ***a*** *∈L(D)*
  - Nótese que estas definiciones son comunes a toda función de interpretación *i()*. Dos funciones de interpretación sólo pueden diferir en la constante elegida para sustituir a la variable en 6 y 9.

Interpretaciones

- Ejemplo: **$\forall x(M(a, x) \wedge P(x)) \to \neg\exists y Q(y)$**
  - Construimos una interpretación sobre el dominio de los números naturales: *D = {0, 1, 2, 3, 4, …}*
  - El lenguaje de la fórmula sólo tiene un símbolo de constante (a), por lo que ampliamos el conjunto de constantes y definimos L(D):

    Ctes. = $\{a, a_1, a_2, a_3, a_4, \ldots\}$
  - La función de interpretación *i* podría ser:
    - $i(a) = 0$, $i(a_1) = 1$, $i(a_2) = 2$, $i(a_3) = 3$, …
    - $P_D(x)$: *x* es par; $Q_D(x)$: *x* es impar; $M_D(x,y)$: $x < y$

$i(\forall x(M(a, x) \wedge P(x))) = F$

porque $i((M(a, x) \wedge P(x))\{x/a\}) = i(M(a, a) \wedge P(a)) =$

$M_D(i(a), i(a)) \wedge P_D(i(a)) = M_D(0, 0) \wedge P_D(0) = F$

$i(\neg\exists y Q(y)) = F$

porque $i(\exists y Q(y)) = V$

ya que $i(Q(y)\{y/a_1\}) = i(Q(a_1)) = Q_D(i(a_1)) = Q_D(1) = V$

por tanto **$i(\forall x(M(a, x) \wedge P(x)) \to \neg\exists y Q(y)) = V$**

Interpretaciones

- Ejemplo: **$\forall x(M(a, x) \wedge P(x)) \rightarrow \neg\exists y Q(y)$**
  - Construimos una interpretación sobre el siguiente dominio: *D = {●, ■, ◆}.*
  - El lenguaje de la fórmula sólo tiene un símbolo de constante (a), por lo que ampliamos el conjunto de constantes y definimos L(D): Ctes. = {a, b, c}
  - La función de interpretación i podría ser:
    - i(a) = ●, i(b) = ■, i(c) = ◆

P_D	
●	V
■	V
◆	V

Q_D	
●	V
■	F
◆	V

M_D	●	■	◆
●	V	V	V
■	F	F	V
◆	V	F	F

$i(\forall x(M(a, x) \wedge P(x))) = V$

porque $i((M(a, x) \wedge P(x))\{x/a\}) = M_D(i(a), i(a)) \wedge P_D(i(a)) = M_D(●, ●) \wedge P_D(●) = V$

$i((M(a, x) \wedge P(x))\{x/b\}) = M_D(i(a), i(b)) \wedge P_D(i(b)) = M_D(●, ■) \wedge P_D(■) = V$

$i((M(a, x) \wedge P(x))\{x/c\}) = M_D(i(a), i(c)) \wedge P_D(i(c)) = M_D(●, ◆) \wedge P_D(◆) = V$

$i(\neg\exists y Q(y)) = F$

porque $i(\exists y Q(y)) = V$

ya que $i(Q(y)\{y/a\}) = i(Q(a)) = Q_D(i(a)) = Q_D(●) = V$

por tanto **$i(\forall x(M(a, x) \wedge P(x)) \rightarrow \neg\exists y Q(y)) = F$**

Relacionándolo con la programación

P_D	
●	V
■	V
◆	V

Q_D	
●	V
■	F
◆	V

M_D	●	■	◆
●	V	V	V
■	F	F	V
◆	V	F	F

$$i(\forall x(M(a,x) \wedge P(x))) = V$$

$$i(\neg\exists y Q(y)) = F$$

```
public static void main(String[] args) {
       boolean[] p = {true,true,true};
       boolean[] q = {true,false,true};
       boolean[][] m = new boolean[][]{
           {true, true, true},
           {false, false, true},
           {true, false, false}
       };//todos los arrays tienen la misma longitude (|D|)

       boolean result1=true;
       for (int i = 0; i < m.length; i++) {
           if(! (m[0][i]==true && p[i]==true) ) {
               result1=false;
               break; //¡mala práctica!
           }
       }

       boolean result2=false;
       for (int i = 0; i < m.length; i++) {
           if(q[i]==true){
               result2=true;
               break; //¡mala práctica!
           }
       }
       System.out.println(result1 + " -> " + result2);

   }

}
```

Interpretaciones

- Ejemplo 1: Consideramos las siguientes afimaciones: *La Tierra orbita en torno al Sol. La Luna orbita en torno a la Tierra. Todo cuerpo que orbita en torno al Sol es un planeta. Son satélites los cuerpos que orbita en torno a planetas.*
  - Dadas las fórmulas: {O(a,b), O(c,a), P(a), S(c)}
  - Podemos interpretarlas sobre el dominio D = {Sol, Tierra, Luna} y con una función de interpretación que plasme el **significado intuitivo** de *orbitar*, *planeta* y *satélite*:
    - i(a) = Tierra; i(b) = Sol; i(c) = Luna
    - P_D(x): x es un planeta
      P_D(Sol) = F; P_D(Tierra) = V; P_D(Luna) = F
    - S_D(x): x es un satélite
      S_D(Sol) = F; S_D(Tierra) = F; S_D(Luna) = V
    - O_D(x,y): x orbita en torno a y
      O_D(Sol, Sol) = F; O_D(Sol, Tierra)= F; O_D(Sol, Luna) = F
      O_D(Tierra, Sol) = V; O_D(Tierra, Tierra) = F; O_D(Tierra, Luna) = F
      O_D(Luna, Sol) = F; O_D(Luna, Tierra) = V; O_D(Luna, Luna) = F

Interpretaciones

$i(a)$ = Tierra; $i(b)$ = Sol; $i(c)$ = Luna

- Ejemplo 1: A partir de la interpretación anterior, podemos evaluar las fórmulas:

$$\{ O(a,b), O(c,a), P(a) \wedge S(c), \forall x(O(x,b) \rightarrow P(x)), \forall x \forall y(O(x,y) \wedge P(y) \rightarrow S(x)) \}$$

- $i(O(a, b)) = O_D(i(a), i(b)) = O_D(\text{Tierra}, \text{Sol}) = V$
- $i(O(c, a)) = O_D(i(c), i(a)) = O_D(\text{Luna}, \text{Tierra}) = V$
- $i(P(a) \wedge S(c)) = V$ sii $i(P(a)) = V$ y $i(S(c)) = V$
  $i(P(a)) = P_D(i(a)) = P_D(\text{Tierra}) = V$
  $i(S(c)) = S_D(i(c)) = S_D(\text{Luna}) = V$
- $i(\forall x(O(x, b) \rightarrow P(x))) = V$ porque
  $i((O(x, b) \rightarrow P(x))\{\mathbf{x/a}\}) = i(O(a, b) \rightarrow P(a)) = V$ sii $i(O(a, b))=F$ o bien $i(P(a))=V$
  $i(P(a)) = P_D(\text{Tierra}) = V$
  $i((O(x, b) \rightarrow P(x))\{\mathbf{x/b}\}) = i(O(b, b) \rightarrow P(b)) = V$ sii $i(O(b, b))=F$ o bien $(P(b))=V$
  $i(O(b,b)) = O_D(\text{Sol}, \text{Sol}) = F$
  $i((O(x, b) \rightarrow P(x))\{\mathbf{x/c}\}) = i(O(c, b) \rightarrow P(c)) = V$ sii $i(O(c, b)) = F$ o bien $i(P(c)) = V$
  $i(O(c, b)) = O_D(\text{Luna}, \text{Sol}) = F$

Interpretaciones

- Ejemplo 1: A partir de la interpretación anterior, podemos evaluar las fórmulas:
  - $i(\forall x \forall y(O(x,y) \wedge P(y) \rightarrow S(x))) = V$ sii
    - $i((\forall y(O(x,y) \wedge P(y) \rightarrow S(x)))\{x/a\}) = i(\forall y(O(a,y) \wedge P(y) \rightarrow S(a))) = V$ sii

$i((O(a,y) \wedge P(y) \rightarrow S(a))\{y/a\}) = i(O(a,a) \wedge P(a) \rightarrow S(a)) = V$ sii $i(O(a,a) \wedge P(a)) = F$ o $i(S(a)) = V$

$i((O(a,y) \wedge P(y) \rightarrow S(a))\{y/b\}) = i(O(a,b) \wedge P(b) \rightarrow S(a)) = V$ sii $i(O(a,b) \wedge P(b)) = F$ o $i(S(a)) = V$

$i((O(a,y) \wedge P(y) \rightarrow S(a))\{y/c\}) = i(O(a,c) \wedge P(c) \rightarrow S(a)) = V$ sii $i(O(a,c) \wedge P(c)) = F$ o $i(S(a)) = V$

- $i((\forall y(O(x,y) \wedge P(y) \rightarrow S(x)))\{x/b\}) = i(\forall y(O(b,y) \wedge P(y) \rightarrow S(b))) = V$ sii

$i((O(b,y) \wedge P(y) \rightarrow S(b))\{y/a\}) = i(O(b,a) \wedge P(a) \rightarrow S(b)) = V$ sii $i(O(b,a) \wedge P(a)) = F$ o $i(S(b)) = V$

$i((O(b,y) \wedge P(y) \rightarrow S(b))\{y/b\}) = i(O(b,b) \wedge P(b) \rightarrow S(b)) = V$ sii $i(O(b,b) \wedge P(b)) = F$ o $i(S(b)) = V$

$i((O(b,y) \wedge P(y) \rightarrow S(b))\{y/c\}) = i(O(b,c) \wedge P(c) \rightarrow S(b)) = V$ sii $i(O(b,c) \wedge P(c)) = F$ o $i(S(b)) = V$

- $i((\forall y(O(x,y) \wedge P(y) \rightarrow S(x)))\{x/c\}) = i(\forall y(O(c,y) \wedge P(y) \rightarrow S(c))) = V$ sii

$i((O(c,y) \wedge P(y) \rightarrow S(c))\{y/a\}) = i(O(c,a) \wedge P(a) \rightarrow S(c)) = V$ sii $i(O(c,a) \wedge P(a)) = F$ o $i(S(c)) = V$

$i((O(c,y) \wedge P(y) \rightarrow S(c))\{y/b\}) = i(O(c,b) \wedge P(b) \rightarrow S(c)) = V$ sii $i(O(c,b) \wedge P(b)) = F$ o $i(S(c)) = V$

$i((O(c,y) \wedge P(y) \rightarrow S(c))\{y/c\}) = i(O(c,c) \wedge P(c) \rightarrow S(c)) = V$ sii $i(O(c,c) \wedge P(c)) = F$ o $i(S(c)) = V$

$i(a) =$ Tierra; $i(b) =$ Sol; $i(c) =$ Luna

Interpretaciones

- <u>Ejercicios</u>:

1. Dadas las fórmulas: {O(a,b), O(c,a), P(a), S(c)}, interpretarlas en el dominio {■, ⭘, △} asignando a los predicados O(_,_), P(_) y S(_) los significados “más grande que”, “cuadrado” y “redondo”, respectivamente.
2. Dadas las fórmulas: {N(a), s(a)=b, N(b)}, interpretarlas en el dominio de las letras del alfabeto latino, siendo N(_) la propiedad de ser una vocal, =(_,_) la relación de identidad y s(_) la función sucesor en el orden lexicográfico usual.
3. *Dadas las fórmulas: {P(a,b), P(a,c), H(b,c), p(b)=d, p(c)=e, R(d,e), A(a,d), A(a,e)}, interpretarlas en el dominio {Juan Carlos, Elena, Felipe, Froilán, Leonor} asignando a P(_,_), H(_,_), R(_,_), A(_,_) y P(_) los significados de “padre”, “hermano”, “primo”, “abuelo” y “primogénito”, respectivamente.*
4. Dadas las fórmulas: {C(a), C(a,b), P(b), C(c), C(c,d), P(d), M(a,c), M(b,d)}, interpretarlas en el dominio {Nueva York, Estados Unidos, Madrid, España, París}, asignando a los predicados C(_), C(_,_), P(_) y M(_,_) los significados “ciudad”, “ser la capital de”, “país” y “ser mayor que”, respectivamente.
5. Interpretar las fórmulas del ejercicio anterior en un dominio diferente, asignando a predicados y constantes una interpretación acorde con el dominio elegido.

Interpretaciones

- **Satisfacción de fórmulas cerradas**

  - No tienen variables libres.
  - Una interpretación $\boldsymbol{I}= <D, i()>$ **satisface** una fórmula cerrada (afirmación) $A \in L$ *sii* $i(A) = V$
    - Una fórmula $A \in L$ es **satisfacible** sii existe al menos una interpretación $\boldsymbol{I}$ tal que $i(A) = V$
    - Una fórmula $A \in L$ es **insatisfacible** sii no existe ninguna interpretación $\boldsymbol{I}$ tal que $i(A) = V$

  - Extensión a conjuntos de fórmulas $\{A_1,\ldots,A_n\}$, $A_i \in L$:
    - Una interpretación $\boldsymbol{I}$ **satisface** $\{A_1,\ldots,A_n\}$ *sii* $i(A_i) = V$ para *toda* $A_i \in \{A_1,\ldots,A_n\}$
    - $\{A_1,\ldots,A_n\}$ es **satisfacible** sii existe al menos una interpretación $\boldsymbol{I}$ tal que $i(A_i) = V$ para *toda* $A_i \in \{A_1,\ldots,A_n\}$
    - $\{A_1,\ldots,A_n\}$ es **insatisfacible** sii no existe ninguna interpretación $\boldsymbol{I}$ tal que $i(A_i) = V$ para *alguna* $A_i \in \{A_1,\ldots,A_n\}$

  - Una fórmula A es **válida** sii $i(A) = V$ para toda interpretación $<D, i()>$
    - No depende del dominio

Interpretaciones

- **Fórmulas abiertas:**
  - Son aquellas que tienen al menos una variable libre (no cuantificada)
  - Ejemplos de fórmulas abiertas:
    1. *Alguien es el rey de Canadá*: R(x,a)
    2. *Algo es igual a dos*: x = 2
    3. *Algo orbita en torno a Júpiter*: O(x,b)
    4. *Si es humano será también mortal*: H(x)→M(x)

  - ¿Que significado tienen estas fórmulas abiertas?
    1. ***R(x,a)*** es **falsa** pues ninguna sustitución de *x* por ninguna persona produce una afirmación verdadera: ninguna persona es rey de Canadá.
    2. ***x = 2*** es una fórmula que **puede ser verdadera** si sustituimos *x* por el numeral 2, mientras que será falsa si elegimos cualquier otro numeral.
    3. ***O(x,b)*** es una fórmula que será **verdadera si** sustituimos *x* por los nombres de algunos cuerpos celestes y falsa si la sustituimos por otros.
    4. ***H(x)→M(x)*** es una fórmula que es **verdadera** para cualquier individuo que pongamos en lugar de *x*: no hay nadie que sea humano y no sea mortal.

Interpretaciones

- **Fórmulas abiertas:**
  - El *significado (valor de verdad) de las fórmulas abiertas* depende de:
    - La interpretación de sus símbolos (predicados y constantes)
    - El valor que tome su variable libre.

  - En algunos casos, para ciertas interpretaciones concretas, el significado de las fórmulas abiertas depende sólo de la interpretación dada a los símbolos
    - Fórmulas abiertas verdaderas
      1. *$x>y \rightarrow x+1>y+1$*
      2. *$Padre(x,y) \rightarrow edad(x)>edad(y)$*

      Mientras se mantenga la interpretación pretendida (aritmética, padres e hijos), estas fórmulas son verdaderas para cualesquiera sustituciones de x e y.
    - Fórmulas abiertas falsas
      1. *$x>1 \rightarrow x>x^2$*
      2. *$Padre(x,y) \rightarrow x=y$*

      Mientras se mantenga la interpretación pretendida (aritmética, padres e hijos), estas fórmulas son falsas para cualesquiera sustituciones de x e y.

Interpretaciones

- **Fórmulas abiertas: satisfacibilidad, verdad y validez**

  Sea A una fórmula abierta:

  - **A es satisfacible** cuando puede ser verdadera:
    - Hay una interpretación **I = <D, i()>**y una sustitución θ**= {x_1/c_1,…, x_n/c_n}** de todas sus variables libres por constantes de L(D) que la hacen verdadera.

      (**existen I = <D, i()> y θ tal que: i((A)θ) = V**)

  - **A es verdadera** en una interpretación cuando:
    - Toda sustitución de sus variables libres por constantes de L(D) la hacen verdadera.

      (**se cumple que i((A)θ) = V para toda θ**)

  - **A es válida** cuando es verdadera en toda interpretación.

    (**se cumple que i((A)θ) = V para toda I= <D,i()> y toda θ**)

Interpretaciones

- **Fórmulas abiertas y cerradas**

  - Dada una fórmula abierta $A(x_1,\ldots,x_n)$, definimos:
    - **Cierre existencial** de $A(x_1,\ldots,x_n)$: $\exists x_1,\ldots,x_n A$
    - **Cierre universal** de $A(x_1,\ldots,x_n)$: $\forall x_1,\ldots,x_n A$

  - Relaciones semánticas entre fórmulas abiertas y cerradas:
    - $A(x_1,\ldots,x_n)$ es satisfacible / insatisfacible sii $\exists x_1,\ldots,x_n A$ lo es.
    - $A(x_1,\ldots,x_n)$ es verdadera en **I**/ válida sii $\forall x_1,\ldots,x_n A$ lo es.

  - Interpretación de fórmulas abiertas y cerradas:
    - En ambas se emplea la operación de sustitución de variables por constantes.

Interpretaciones

- **Modelos y Contra-Modelos**
  - **Modelo**: una interpretación I es modelo de una fórmula A *sii* A es verdadera en I
    - Esta definición se aplica igualmente a fórmulas abiertas y cerradas.
  - **Contra-modelo**: una interpretación I es contra-modelo de una fórmula A sii para alguna sustitución θ de todas sus variables libres $i((A)\theta) = F$
    - En el caso de que A sea fórmula cerrada, I es contra-modelo sii $i(A) = F$

  - Ejercicio. Definir un modelo y un contra-modelo para cada una de las siguientes fórmulas:
    - $\forall x(P(x,a) \rightarrow Q(f(x)))$
    - $\exists x(P(x,a) \rightarrow Q(f(x)))$
    - $\exists x P(x) \wedge P(a)$
    - $\forall x P(x) \wedge \neg P(a)$
    - $\forall x \exists y(x = y)$ (Recuerda: orden de cuantificador importa, = tiene semántica única, equivale a Igual(x,y))
    - $\exists y \forall x(x = y)$
    - $\forall y \forall z \exists x(x = y+z)$ (Recuerda: + es una función s(x,y) con semántica ya definida)
    - $\exists x \forall y \forall z(x = y+z)$
    - $\exists x(\forall y P(x,y) \rightarrow Q(f(x))) \vee \forall x P(x,x)$
    - $\forall x \neg(\forall y P(x,y) \rightarrow Q(f(x))) \wedge \exists x \neg P(x,x)$

Ejercicio 10

- Encuentra modelo y contramodelo de FBF_1≡ **∀x¬(∀yP(x,y) →Q(f(x))) ∧∃x¬P(x,x)** ≡ **∀xA ∧ ∃xB**
  - En una conjunción es fácil encontrar un contramodelo.
  - Contramodelo: considero dominio de un elemento 0, y defino $P_D(0,0)$=V, lo que hace i(∃x¬P(x,x))=F y con ello i(FBF_1)= i(∀xA ∧ ∃xB)=F.
  - Modelo: intento considerar $P_D(0,0)$=F para que se cumpla i(∃x¬P(x,x))=V y busco i(∀xA)=V
    - como sólo puedo sustituir x por un elemento i(∀xA)=V sii i(A{x/a})=V=i(¬(∀yP(a,y) →Q(f(a)))=i(¬A1)=V.
    - i(¬A1)=V sii i(A1)=F sii i(∀yP(a,y))=V y i(Q(f(a))=F.
      - Para que i(Q(f(a))=F, puedo definir f_D como la función identidad (f_D (0)=0) y Q_D (0)=F.
      - i(∀yP(a,y))=V sii P_D (0,0)=V, pero habíamos dicho que P_D(0,0)=F... así que no se ha podido encontrar modelo.
      - ...con este dominio y esta interpretación.
  - Ampliamos dominio a D={0,1} y seguimos buscando modelo de FBF1
    - (podríamos haber empezado así y ahorrar el trabajo anterior)
    - Para i(∃x¬P(x,x))=V, P_D(0,0)=F **o bien** P_D(1,1)=F (ver tablas)
    - Para i(∀xA)=V, i(A{x/a})=V **y además** i(A{x/b})=V
      - i(¬(∀yP(a,y) →Q(f(a)))=V sii i(∀yP(a,y) →Q(f(a))=F
      - i(¬(∀yP(b,y) →Q(f(b)))=V sii i(∀yP(b,y) →Q(f(b))=F
        - Es fácil hacer Q(f(a)) y Q(f(b)) falsos (f_D (0)=0 y f_D (1)=1, Q_D (0)=F y Q_D (1)=F)
        - Es más complicado ver si es posible hacer i(∀yP(a,y)) y i(∀yP(b,y)) verdadero: probar todas las sustituciones.
          - Pero no es posible hacer modelo... (ver tablas)

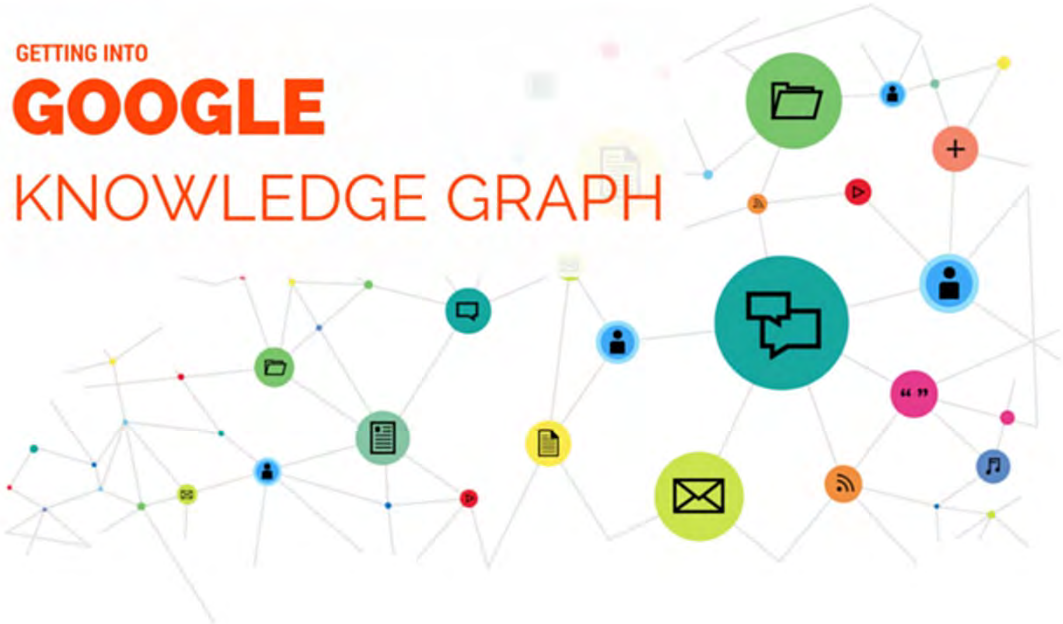
La web semántica (divulgación)

- Supongamos la siguiente búsqueda en google:
  - personas que trabajan en un equipo de futbol de primera división
  - ¿Qué busca google?
- ¿Qué información aporta una web con (sólo) el texto:
  - "Ramos juega en el Real Madrid"?
  - …o un tweet : "Ana es miembro de la seguridad privada del Club Atlético de Madrid"
- Se podría formalizar con lógica la siguiente información:
  - Ramos es una constante del predicado unitario Persona
  - Real Madrid es una constante del predicado unitario Equipo de futbol de primera división
  - Jugar(x,y) y Equipo_futbol_de_primera(y) implica Trabajar(x,y)
    - (jugar es una subpropiedad trabajar)
- Entonces la búsqueda podría incluir Ramos en un listado
  - …y aumentar el listado al considerar varias webs (incluyendo a Ana por ejemplo)
  - …y deducir nueva información…¿es Ramos un deportista profesional?
  - …y permitir búsquedas más complejas: "Deportista profesionales casados/as con presentadores/as de televisión"
  - …y detectar inconsistencias en el conocimiento, argumentos incorrectos…¿puede Ramos estar casado con dos mujeres?

FOL	OWL	DL
unary predicate	class	concept
binary predicate	property	role
constant	individual	individual

La web semántica (divulgación)

```
Código RDFS Schema:
:Jugador rdfs:subClassOf :Persona .
:ramos rdf:type :Jugador .
:ramos rdfs:label "Sergio Ramos" .
:rmadrid rdf:type :EquipoDeFutbolPrimera .
:rmadrid rdfs:label "Real Madrid C. F." .
:juega_en rdfs:subPropertyOf :trabaja_en .
:juega_en rdfs:range :EquipoDeFutbolPrimera .
:ramos :juega_en :rmadrid .
```



Departamento de Inteligencia Artificial
Escuela Técnica Superior de
Ingenieros Informáticos

Lógica
Tema 9: Razonamiento Semántico

Profesor: Emilio Serrano
emilioserra@fi.upm.es

Temas

❑ Bloque II

Tema 7
Lenguajes de primer orden: sintaxis y uso en la formalización de argumentos

Tema 8
Semántica formal: Funciones de verdad, tautologicidad, consecuencia lógica

Representación

Tema 9
Razonamiento semántico: definición de modelos y contra-modelos

Tema 10
Cálculo de deducción natural de primer orden

Razonamiento

Tema 11
Forma Normal de Skolem. Forma clausular

Tema 12
Sustitución y Unificación máximamente general

Tema 13
Cálculo de resolución con Unificación

Computación

Validez y consecuencia lógica

Validez lógica

- Una fórmula A $\in$ L es **válida** (lógicamente válida) sii es verdadera en toda interpretación: **$\models$ A**
- **Sí es posible definir una interpretación que hace falsa la fórmula ⇨ La fórmula no es válida**

Consecuencia lógica

- Dado un conjunto de fórmulas $\Gamma = \{A_1,\ldots,A_n\}$ ($A_i \in L$) y una fórmula B $\in$ L, B es consecuencia lógica de Γ **($\Gamma \models$ B)** sii:
  - Todo modelo de Γ es también modelo de B (toda interpretación que haga verdad a Γ también hace verdad a B)
  - **No existe ninguna interpretación que haga verdad a Γ y que no haga verdad a B**

Recuerda de tema anterior…

Asignación de valor de verdad a las fórmulas moleculares:

1. ***i(¬A) = V*** *sii i(A) = F;* ***i(¬A) = F*** *sii i(A) = V*
2. ***i(A ∧B) = V*** *sii i(A) = V* y *i(B) = V;* ***i(A ∧B) = F*** *siii(A) = F* o *i(B) = F*
3. ***i(A ∨B) = V*** *sii i(A) = V* o *i(B) = V;* ***i(A ∨B) = F*** *siii(A) = F* y *i(B) = F*
4. ***i(A →B) = V*** *sii i(A) = F* o *i(B) = V;* ***i(A →B) = F*** *siii(A) = V* y *i(B) = F*
5. ***i(A ↔B) = V*** *sii i(A) = i(B);* ***i(A ↔B) = F*** *sii i(A) ≠ i(B);*
6. ***i(∃xA) = V*** *sii i(A{x/a}) = V* para al menos una sustitución *x/a,* ***a*** *∈L(D)*
7. ***i(∃xA) = F*** *sii i(A{x/a}) = F* para toda sustitución *x/a*, ***a*** constante de *L(D)*
8. ***i(∀xA) = V*** *sii i(A{x/a}) = V* para toda sustitución *x/a*, ***a*** constante de *L(D)*
9. ***i(∀xA) = F*** *sii i(A{x/a}) = F* para al menos una sustitución *x/a*, ***a*** *∈L(D)*

(En los ejercicios aparece "y además" en todas las sustituciones para i(∀xA) = V y i(∃ xA) = F; y "o bien" en las sustituciones para i(∃xA) = V y para i(∀xA) = F)

Ejemplo de $\models$

¿ $\models \exists x \forall y P(x,y) \rightarrow \forall y \exists x P(x,y)$?

D = {1, 2} i(a)=1, i(b)=2

$i(\exists x \forall y P(x,y) \rightarrow \forall y \exists x P(x,y))$ **= F** sii

$i(\exists x \forall y P(x,y))$ = V sii

- {x/a} $i(\forall y P(a,y))$ = V sii — $\underset{\{y/a\}}{i(P(a,a))}$ = V y $\underset{\{y/b\}}{i(P(a,b))}$ = V
- o bien
- {x/b} $i(\forall y P(b,y))$ = V sii — $\underset{\{y/a\}}{i(P(b,a))}$ = V y $\underset{\{y/b\}}{i(P(b,b))}$ = V

$i(\forall y \exists x P(x,y))$ = F sii

- {y/a} $i(\exists x P(x,a))$ = F sii — $\underset{\{x/a\}}{i(P(a,a))}$ = F y $\underset{\{x/b\}}{i(P(b,a))}$ = F
- o bien
- {y/b} $i(\exists x P(x,b))$ = F sii — $\underset{\{x/a\}}{i(P(a,b))}$ = F y $\underset{\{x/b\}}{i(P(b,b))}$ = F

La imposibilidad de hacer falsa esta fórmula no es exclusiva de esta interpretación. Verificar el antecedente es incompatible con hacer falso el consecuente ⇨ **La fórmula es válida**

Ejemplo de $\not\models$

¿ $\models \forall y \exists x P(x,y) \rightarrow \exists x \forall y P(x,y)$?

D = {1, 2}   i(a)=1, i(b)=2

i($\forall y \exists x P(x,y) \rightarrow \exists x \forall y P(x,y)$) **= F** sii

i($\forall y \exists x$P(x,y)) = V sii

{y/a} i($\exists x$P(x,a)) = V sii   {x/a}   {x/b}

Y además   **i(P(a,a)) = V** o bien i(P(b,a)) = V

{y/b} i($\exists x$P(x,b)) = V sii   {x/a}   {x/b}

i(P(a,b)) = V o bien **i(P(b,b)) = V**

i($\exists x \forall y$P(x,y)) = F sii

{x/a}   i($\forall y$P(a,y)) = F sii   {y/a}   {y/b}

Y además   i(P(a,a)) = F o bien **i(P(a,b)) = F**

{x/b}   i($\forall y$P(b,y)) = F sii   {y/a}   {y/b}

i(P(b,a)) = F o bien i(P(b,b)) = F

Sí es posible definir una interpretación que hace falsa la fórmula ⇨ **La fórmula no es válida, el contramodelo es: $P_D(1,1)$ = V, $P_D(1,2)$ = F $P_D(2,1)$ = F, $P_D(2,2)$ = V**

Corrección de un argumento. Consecuencia Lógica. Ejemplo 1

- Sea el argumento: *Hay individuos inteligentes o que saben hablar. Juan no sabe hablar. Luego Juan no es inteligente.*
- Y su formalización: $\{ \exists x(I(x) \vee H(x)), \neg H(a) \} \vDash \neg I(a)$
- Para *determinar la corrección de este argumento por medios semánticos*, intentamos *encontrar un* ***contramodelo***, es decir una interpretación que:
  1. Verifique las premisas: $i(\exists x(I(x) \vee H(x))) = V$ y $i(\neg H(a)) = V$
  2. Haga falsa la pretendida conclusión: $i(\neg I(a)) = F$
  - Elijamos para ello un dominio de interpretación, p .ej. D = { Juan, Pedro }
  - Y ampliemos el lenguaje en el que hemos formalizado el argumento con una segunda constante: b
  - La interpretación:
    1. $i(a) = Juan$, $i(b) = Pedro$
    2. $I_D(Juan) = V$, $I_D(Pedro) = V$, $H_D(Juan) = F$, $H_D(Pedro) = V$
- La interpretación verifica las premisas y hace falsa la conclusión, luego el argumento no es correcto.

Corrección de un argumento. Consecuencia Lógica. Ejemplo 1

Veamos en detalle la verificación de las premisas y falsificación de la conclusión:

1. **(Premisa)** **$i(\exists x(I(x) \vee H(x))) = V$ sii**
   - $i((I(x) \vee H(x))\{x/c\}) = V$ para alguna constante 'c' del lenguaje L.
     - Sea 'a' esa constante: $i(I(a) \vee H(a)) = V$ sii
       - $i(I(a)) = V$

         o bien
       - $i(H(a)) = V$
     - $i(I(a)) = V$ porque $i(a) =$ Juan y $I_D(\text{Juan}) = V$

2. **(Premisa)** **$i(\neg H(a))) = V$** sii $i(H(a)) = F$
   - Es falso porque $i(a) =$ Juan y $H_D(\text{Juan}) = F$

3. **(Conclusión)** **$i(\neg I(a)) = F$** sii $i(I(a)) = V$
   - y lo es porque $i(a) =$ Juan y $I_D(\text{Juan}) = V$

Corrección de un argumento. Consecuencia Lógica. Ejemplo 2

- Sea el argumento: *Todo elemento químico es oxidante o reductor. El carbono es un elemento químico no oxidante. Luego el carbono es reductor.*
- Y su formalización: $\{ \forall x(E(x) \rightarrow O(x) \vee R(x)), E(a) \wedge \neg O(a) \} \vDash R(a)$
- Para determinar la corrección de este argumento por medios semánticos, tratemos de encontrar un contramodelo que:
  1. Verifique las premisas: $i(\forall x(E(x) \rightarrow O(x) \vee R(x))) = V$ y $i(E(a) \wedge \neg O(a)) = V$
  2. Haga falsa la pretendida conclusión: $i(R(a)) = F$
- Elijamos para ello un dominio de interpretación, p .ej. D = { carbono, oxígeno }
- Ampliemos el lenguaje de formalización del argumento con una nueva constante: b
- La interpretación:
  1. i(a) = carbono, i(b) = oxígeno
  2. E_D(carbono) = E_D(oxígeno) = V, O_D(carbono) = F, O_D(oxígeno) = V, R_D(carbono) = R_D(oxígeno) = F
- No puede verificar las premisas y hacer falsa la conclusión:
  1. $i(\forall x(E(x) \rightarrow O(x) \vee R(x))) = V$ sii

     $i(E(a) \rightarrow O(a) \vee R(a)) = V$ sii **$i(E(a))=F$ o $i(O(a) \vee R(a))=V$**

     $i(E(b) \rightarrow O(b) \vee R(b)) = V$ sii $i(E(b))=F$ o $i(O(b) \vee R(b))=V$
  2. $i(E(a) \wedge \neg O(a)) = V$ sii

     $i(E(a))=V$ y $i(O(a))=F$
  3. $i(R(a)) = F$

◆ En realidad, lo que impide crear el contramodelo ***no depende de I***, luego **el argumento es correcto.** 9

Validez y Consecuencia Lógica. Ejercicios

Demostrar por medios semánticos:

1. $\{\forall x(P(x) \rightarrow Q(x)), P(a)\} \vDash Q(a)$
2. $\{\exists x(P(x) \rightarrow Q(x)), P(a)\} \nvDash Q(a)$
3. $\{\forall x(P(x) \rightarrow R(x)), \exists x(P(x) \wedge \neg Q(x))\} \vDash \exists x(R(x) \wedge \neg Q(x))$
4. $\vDash \forall x(P(x) \rightarrow Q(x)) \leftrightarrow \forall x(\neg Q(x) \rightarrow \neg P(x))$
5. $\nvDash \exists x(P(x) \wedge Q(x)) \leftrightarrow \exists x P(x) \wedge \exists x Q(x)$

Validez de argumentos. Ejercicios

Determina por medios semánticos la validez de los siguientes argumentos:

1. *0 > 1. 1 > 2. La relación ‘>’ es transitiva. Por tanto, 0 > 2.*
2. *Robin Hood es generoso. Todos los ladrones son generosos. Por tanto, Robin Hood es un ladrón.*
3. *Siempre que Pedro discute con María, ésta se enfada con su padre. Una persona que se enfade con otra no la invita a su boda. Por tanto, si Pedro discute con María, ésta no invitará a su padre a su boda.*
4. *Todos los ladrones son generosos. Sólo los ricos son generosos. Robin Hood es un ladrón. Por tanto, Robin Hood es rico.*
5. *Juan Carlos ama a Sofía. Quienes trabajan con Juan Carlos no conocen a fondo a Sofía. Si se ama a alguien, se le conoce a fondo. Sabino trabaja con Juan Carlos. Por tanto, Sabino no ama a Sofía.*
6. *Hay un ladrón generoso. Por tanto, hay un ladrón y hay alguien generoso.*
7. *Hay un ladrón y hay alguien generoso. Por tanto, hay un ladrón generoso.*

Validez de argumentos. Solución de ejercicios

1. *0 > 1. 1 > 2. La relación '>' es transitiva. Por tanto, 0 > 2.*
   - $\{ M(a,b), M(b,c), \forall x \forall y \forall z(M(x,y) \land M(y,z) \to M(x,z)) \} \models M(a,c)$ (¡A3 tiene 27 combinaciones de θ!)
2. *Robin Hood es generoso. Todos los ladrones son generosos. Por tanto, Robin Hood es un ladrón.*
   - $\{ G(a), \forall x(L(x) \to G(x)) \} \not\models L(a)$
3. *Siempre que Pedro discute con María, ésta se enfada con su padre. Una persona que se enfade con otra no la invita a su boda. Por tanto, si Pedro discute con María, ésta no invitará a su padre a su boda.*
   - $\{ D(a,b) \to E(b,f(b)), \forall x \forall y(E(x,y) \to \neg i(x,y)) \} \models D(a,b) \to \neg i(b,f(b))$
4. *Todos los ladrones son generosos. Sólo los ricos son generosos. Robin Hood es un ladrón. Por tanto, Robin Hood es rico.*
   - $\{ \forall x(L(x) \to G(x)), \forall x(\neg R(x) \to \neg G(x)), L(a) \} \models R(a)$
5. *Juan Carlos ama a Sofía. Quienes trabajan con Juan Carlos no conocen a fondo a Sofía. Si se ama a alguien, se le conoce a fondo. Sabino trabaja con Juan Carlos. Por tanto, Sabino no ama a Sofía.*
   - $\{ A(a,b), \forall x(T(x,a) \to \neg C(x,b)), \forall x \forall y(A(x,y) \to C(x,y)), T(c,a) \} \models \neg A(c,b)$
6. *Hay un ladrón generoso. Por tanto, hay un ladrón y hay alguien generoso.*
   - $\exists x(L(x) \land G(x)) \models \exists x L(x) \land \exists x G(x)$
7. *Hay un ladrón y hay alguien generoso. Por tanto, hay un ladrón generoso.*
   - $\exists x L(x) \land \exists x G(x) \not\models \exists x(L(x) \land G(x))$

Ejercicio de examen Julio 2015

Ejercicio 2. Sea L el lenguaje { R , a , b , c }, donde R(x,y) es un símbolo de predicado y a, b, c son constantes. Definir un contramodelo en un dominio de 3 elementos para demostrar lo siguiente:

(2,5 puntos)

$$\forall x \exists y\, R(x,f(y)) \not\models \exists y \forall x\, R(x,f(y))$$

Justificar adecuadamente la respuesta con un análisis semántico.

En la lógica el orden de los cuantificadores es importante. Por ejemplo, que todo el mundo ame a alguien no implica que alguien sea amado por todo el mundo.

Para construir un contramodelo i hay que definir un dominio D, y dar los valores de la función f y el predicado R.

Una solución sencillo sería: dominio de i, D= {1,2,3}.

Interpretación i(a)=1, i(b)= 2, i(c)= 3.

Sea f la función de identidad: f(x) = x; entonces tenemos f(1) =1, f(2)=2, f(3)=3.

Se define R tal que R(x,y) es verdadero en el modelo si y solo si x=y. Entonces los valores de R son <1,1>, <2,2> y <3,3>. Por eso, R(a,a), R(b,b) y R(c,c) son verdaderos en i y otros átomes con R son falsos. Poer ejemplo. piensa en un dominio de egoistas donde todo el mundo se ama a sí mismo y a nadie más.

Ahora es evidente que en el modelo i la formula ∀x ∃y R(x,f(y)) es verdadera. Eso porque i(R(x,f(y)){x/a, y/a} = V, i(R(x,f(y)){x/b, y/b} = V y i(R(x,f(y)){x/c, y/c} = V: para cada sustitución por x se puede elegir una sustitución por y (=f(y)) tal que la fórmula sale verdadero.

Al contrario, la fórmula ∃y ∀x R(x,f(y)) es falsa en i. Con la substitución y/a, la fórmula R(b,a) es falsa; con la substitución y/b, la fórmula R(c,b) es falsa, y con la substitución y/c, la fórmula R(b,c) es falsa. Dado que no hay una substitución por y tal que R(x,f(y)) es verdadero para cada substitución por x, tenemos i(∃y ∀x R(x,f(y))) = F

Tenemos por tanto un contramodelo que demuestra la proposición.

Si no es consecuencia lógica, siempre hay que dar el contramodelo: funciones, interpretación de constantes, e interpretación de predicados.

El uso de tablas bidimensionales para la interpretación de predicados de dos argumentos puede ser útil también. Ya sea para comprobar una hipótesis o para desarrollarla.

R_D	1	2	3
1	V	F	F
2	F	V	F
3	F	F	V

La regla para un examen es siempre: tiene que haber (1) un desarrollo claro y justificado de como alcanzar la conclusión, y (2) una conclusión clara y explícita.

En semántica (LP y LPO) se admite el procedimiento inverso: (1) definir un contramodelo al azar (o guiado, o con tablas de verdad en sucio) para demostrar ⊭, e (2) interpretar las formulas EXPLÍCITAMENTE Y PASO A PASO para comprobar que efectivamente es un contramodelo. (Aunque un error por este procedimiento puede penalizar más).

Lógica

Tema 10: Cálculo Deductivo

Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingenieros Informáticos

Profesor: Emilio Serrano
emilioserra@fi.upm.es

Temas

❑ Bloque II

Tema 7
Lenguajes de primer orden: sintaxis y uso en la formalización de argumentos

Tema 8
Semántica formal: Funciones de verdad, tautologicidad, consecuencia lógica

Representación

Tema 9
Razonamiento semántico: definición de modelos y contra-modelos

Tema 10
Cálculo de deducción natural de primer orden

Razonamiento

Tema 11
Forma Normal de Skolem. Forma clausular

Tema 12
Sustitución y Unificación máximamente general

Tema 13
Cálculo de resolución con Unificación

Computación

Índice de la Parte 2: Lógica de Primer Orden

Introducción

❑ Motivación para construir un cálculo

- Dificultad para determinar **$\Gamma \models A$** por medios semánticos:
  - o En el caso de la Lógica Proposicional, hay que explorar un número exponencialmente creciente de valoraciones
  - o En el caso de la Lógica de Primer Orden, es imposible explorar todas las posibles asignaciones de significado (interpretaciones).

- Alternativa: determinar que **A** se deduce de **Γ** por medios sintácticos:

  $\Gamma \vdash A$
  - o En lugar de razonar sobre el significado de las fórmulas (valoraciones o interpretaciones).
  - o Razonar sobre la forma de las fórmulas.

Un cálculo deductivo para la LPO

- **Axiomas lógicos** de la teoría.
- **Reglas de inferencia**: 14 (las mismas 10 reglas básicas ya estudiadas más dos por cada cuantificador).
- Definición de **prueba**: una prueba de una fórmula en una teoría es una secuencia de fórmulas en la que cada elemento es:
  - **Premisa** o **supuesto temporal** de la teoría, o bien
  - **Resultado** de la aplicación de una regla de inferencia sobre fórmulas anteriores en la secuencia, y tal que
  - la última fórmula de la secuencia es la **fórmula probada**
- Definición de **teorema**: una fórmula B es teorema de una teoría *T[A1,...,An]* si *B* tiene una prueba en dicha teoría *(T[A1,...,An]* $\vdash$ *B)*.

Relación con la consecuencia lógica (recordatorio)

- Si el cálculo es correcto y completo entonces $\vdash$ y $\models$ son equivalentes.
  - Esto es lo que ocurre con la deducción natural.

- **Corrección**: **Teorema de validez**
  - Todos los teoremas de $T[\Gamma]$ son consecuencias lógicas de Γ:

$$\text{si } T[\Gamma] \vdash B \text{ entonces } \Gamma \models B$$

- **Completitud: Teorema de completitud**
  - Dada una teoría $T[\Gamma]$, todas las consecuencias lógicas de Γ son teoremas de $T[\Gamma]$:

$$\text{si } \Gamma \models B \text{ entonces } T[\Gamma] \vdash B$$

Consideraciones previas

¿ $T[\exists x P(x) \rightarrow \forall y Q(y),\ \forall y Q(y) \rightarrow \forall z R(z)] \vdash \exists x P(x) \rightarrow \forall z R(z)$?

Con las reglas básicas que ya conocemos para lenguajes proposicionales podemos demostrar:

$T[p \rightarrow q,\ q \rightarrow r] \vdash p \rightarrow r$

1.	$p \rightarrow q$	premisa
2.	$q \rightarrow r$	premisa
3.	p	supuesto
4.	q	E$\rightarrow$ 1,3
5.	r	E$\rightarrow$ 2,4
6.	$p \rightarrow r$	I$\rightarrow$ 3,5

Sin añadir ninguna regla más también podríamos demostrar la fórmula anterior:

1.	$\exists x P(x) \rightarrow \forall y Q(y)$	premisa
2.	$\forall y Q(y) \rightarrow \forall z R(z)$	premisa
3.	$\exists x P(x)$	supuesto
4.	$\forall y Q(y)$	E$\rightarrow$ 1,3
5.	$\forall z R(z)$	E$\rightarrow$ 2,4
6.	$\exists x P(x) \rightarrow \forall z R(z)$	I$\rightarrow$ 3,5

Consideraciones previas

¿ $T[\forall x(P(x) \rightarrow Q(x)), \forall x(Q(x) \rightarrow R(x))] \vdash \forall x(P(x) \rightarrow R(x))$?

- El argumento es correcto, pero las reglas básicas que conocemos **son insuficientes** para poder demostrarlo

- Necesitamos **añadir nuevas reglas** para (cuando sea necesario):
  1. "abrir" las fórmulas cuantificadas (eliminar cuantificadores).
  2. aplicar las reglas de inferencia proposicionales.
  3. "cerrar" las fórmulas resultantes (introducir cuantificadores).

Reglas de deducción natural en LPO

❑ Reglas del cuantificador existencial

- *∃xA* es el resultado de sustituir apariciones del término *t* en la fórmula *A(t)* por una variable *x*. **Introducción de ∃**:

A(t)

∃xA(x)

T[P(a) v Q(a)] ⊢ ∃x(P(x) v Q(x))

1.	P(a) v Q(a)	premisa
2.	∃x(P(x) v Q(x))	**$I_{\exists}$ 1**

- *A{x/a*}* resulta de sustituir todas las apariciones de *x* por *a* en *∃xA*, *a* una constante que no ha aparecido hasta entonces. **Eliminación de ∃**:

∃xA

A{x/a*}

T[∃x(P(x) ∧ Q(x))] ⊢ ∃xP(x)

1.	∃x(P(x) ∧ Q(x))	premisa
2.	P(a*) ∧ Q(a*)	**$E_{\exists}$ 1**
3.	P(a*)	$E_{\wedge}$ 2
4.	∃xP(x)	$I_{\exists}$ 3

Reglas de deducción natural en LPO

$T[\exists x(P(x) \land Q(x))] \vdash \exists xP(x) \land \exists xQ(x)$

1.	$\exists x(P(x) \land Q(x))$	premisa
2.	$P(a^*) \land Q(a^*)$	$E_{\exists}$ 1
3.	$P(a^*)$	$E_{\land}$ 2
4.	$\exists xP(x)$	$I_{\exists}$ 3
5.	$Q(a^*)$	$E_{\land}$ 2
6.	$\exists xQ(x)$	$I_{\exists}$ 5
7.	$\exists xP(x) \land \exists xQ(x)$	$I_{\land}$ 4,6

se introduce a como una constante nueva, sólo vigente mientras llegamos a deducir la fórmulas que nos interesan, que son $\exists xP(x)$ y $\exists xQ(x)$*

Reglas de deducción natural en LPO

Al usar otra vez a damos a entender que el elemento que cumple Q es el mismo que cumple P, y no tiene por qué ser así*

DEMOSTRACIÓN INCORRECTA

$T[\exists x P(x) \land \exists x Q(x)] \vdash \exists x(P(x) \land Q(x))$

1.	$\exists x P(x) \land \exists x Q(x)$	premisa
2.	$\exists x P(x)$	$E\land$ 1
3.	$P(a^*)$	$E_\exists$ 2
4.	$\exists x Q(x)$	$E\land$ 1
5.	$\boxed{Q(a^*)}$	$E_\exists$ 4
6.	$P(a^*) \land Q(a^*)$	$I\land$ 3,5
7.	$\exists x(P(x) \land Q(x))$	$I_\exists$ 6

Contraejemplo de la deducción:
"hay números que son pares y hay números que son impares, pero no es cierto que haya números que son pares e impares simultáneamente"

Reglas de deducción natural en LPO

❑ Reglas del cuantificador universal

- *A{x/t}* es el resultado de sustituir todas las apariciones de *x* en *∀xA* por un término cualquiera *t*. **Eliminación de ∀** :

$$\frac{\forall xA}{A\{x/t\}}$$

T[∀xP(x)] ⊢ P(f(b))

1. ∀xP(x)	Premisa
2. P(f(b))	**$E_\forall$ 1**

T[∀x(P(x) → Q(x)), P(a)] ⊢ Q(a)

1. ∀x(P(x) → Q(x))	premisa
2. P(a)	premisa
3. P(a) → Q(a)	**$E_\forall$ 1**
4. Q(a)	MP 2,3

T ⊢ ∀xP(x) → ∃xP(x)

1.	∀xP(x)	supuesto
2.	P(a)	**$E_\forall$ 1**
3.	∃xP(x)	$I_\exists$ 2
4. ∀xP(x) → ∃xP(x)		I→ 1,3

Reglas de deducción natural en LPO

❑ Reglas del cuantificador universal

- *x* aparece libre en *A(x)* y no aparece libre en ninguna premisa o supuesto previo no cancelado. **Introducción de ∀** :

$$\frac{A(x)}{\forall x A}$$

CONDICIONES:

- *A(x) no incluye nombres temporales (a*)*
- *x no aparece libre en premisa, libre fuera de la subprueba en la que se aplica la regla, ni libre en un supuesto previo no cancelado*

T[∀xP(x)] ⊢ ∀x(P(x) ∨ Q(x))

1.	∀xP(x)	premisa
2.	P(x)	$E_\forall$ 1
3.	P(x) ∨ Q(x)	I∨ 2
4.	∀x(P(x) ∨ Q(x))	I∀ 3

T[∀x(P(x) → Q(x)), ∀x(Q(x) → R(x))] ⊢ ∀x(P(x) → R(x))

1.	∀x(P(x) → Q(x))	premisa
2.	∀x(Q(x) → R(x))	premisa
3.	P(x) → Q(x)	$E_\forall$ 1
4.	Q(x) → R(x)	$E_\forall$ 2
5.	P(x)	supuesto
6.	Q(x)	MP 3,5
7.	R(x)	MP 4,6
8.	P(x) → R(x)	I→ 5,7
9.	∀x(P(x) → R(x))	**$I_\forall$ 8**

Reglas de deducción natural en LPO

No se puede generalizar una fórmula que tiene constantes temporales

DEMOSTRACIÓN INCORRECTA

$T[\exists x P(x)] \vdash \forall x P(x)$

1.	$\exists x P(x)$	Premisa
2.	$P(a^*)$	$E_\forall$ 1
3.	$\forall x P(x)$	$I_\forall$ 2

Contraejemplo de la deducción: "hay números que son primos, pero eso no nos lleva a afirmar que todos los números lo son"

DEMOSTRACIÓN INCORRECTA

$T[\forall x \exists y M(y,x)] \vdash \exists y \forall x M(y,x)$

1.	$\forall x \exists y M(y,x)$	premisa
2.	$\exists y M(y,x)$	$E_\forall$ 1
3.	$M(a^*,x)$	$E_\exists$ 2
4.	$\forall x M(a^*,x)$	$I_\forall$ 3
5.	$\exists y \forall x M(y,x)$	$I_\exists$ 4

Contraejemplo de la deducción: "todo el mundo tiene madre, pero no hay nadie que sea la madre de todo el mundo"

Otro uso erróneo de $\mathbf{I}_\forall$

No se puede generalizar sobre la x, en los puntos 4 y 8, porque aparece libre fuera del razonamiento del supuesto en el que se aplica la regla

DEMOSTRACIÓN INCORRECTA

$T[\forall x(P(x) \vee Q(x))] \vdash \forall xP(x) \vee \forall xQ(x)$

1.	$\forall x(P(x) \vee Q(x))$	premisa
2.	$P(x) \vee Q(x)$	$E_\forall$ 1
3.	$\quad P(x)$	supuesto
4.	$\quad \forall xP(x)$	**$\mathbf{I}_\forall$ 3**
5.	$\quad \forall xP(x) \vee \forall xQ(x)$	I_v 4
6.	$P(x) \rightarrow \forall xP(x) \vee \forall xQ(x)$	$I\rightarrow$ 3,5
7.	$\quad Q(x)$	supuesto
8.	$\quad \forall xQ(x)$	**$\mathbf{I}_\forall$ 7**
9.	$\quad \forall xP(x) \vee \forall xQ(x)$	I_v 8
10.	$Q(x) \rightarrow \forall xP(x) \vee \forall xQ(x)$	$I\rightarrow$ 7,9
11.	$\forall xP(x) \vee \forall xQ(x)$	E_v 2,6,10

Contraejemplo de la deducción: "todos los números naturales son pares o impares, pero no es cierto que o todos los números son pares o todos son impares"

Estrategia general problemas DN (recordatorio LP)

- Identifica la conectiva principal y el tipo de formula de la conclusión y opera según lo siguiente:
  - Si es $A \rightarrow B$, supón A y usa $\mathbf{I}_{\rightarrow}$ cuando encuentres B.
  - Si es $A \wedge B$, prueba A, prueba B, y usa $\mathbf{I}_{\wedge}$.
  - Si es $A \vee B$, prueba A o B (el más fácil), y usa $\mathbf{I}_{\vee}$.
    - Muy posiblemente hay que ir a la última estrategia o hacer supuestos para aplicar $\mathbf{E}_{\vee}$
  - Si es $A \leftrightarrow B$, prueba $A \rightarrow B$, $B \rightarrow A$, y aplica $\mathbf{I}_{\leftrightarrow}$
  - Si es $\neg A$, supón A, encuentra una contradicción, y usa $\mathbf{I}_{\neg}$.
  - Si es A, intenta encontrar una solución trivial con los elementos previos de la deducción (la $\mathbf{I}_{\vee}$ puede ser útil). Si no lo consigues, supón $\neg A$, encuentra una contradicción, y usa $\mathbf{I}_{\neg}$.
  - Si todo falla, supón la conclusión negada en bloque, encuentra una contradicción, y usa $\mathbf{I}_{\neg}$.

Estrategias Deductivas

Fórmula	Como Premisa	Como Conclusión
P ∧ Q	Deduce P y deduce Q	Demuestra P y Q por separado
P ∨ Q	Demuestra P → __, Q → __, y deduce __	Demuestra o P o Q
P → Q	Demuestra P, y deduce Q	Supón P y demuestra Q
¬P	Si ¬P es ¬¬Q, deduce Q	Supón P y demuestra __ ∧ ¬__
P ↔ Q	Deduce P → Q y Q → P	Demuestra P → Q y Q → P
∀x P(x)	Deduce P(_)	Demuestra P(x) para la variable x
∃x P(x)	Deduce P(a*) para algún nombre temporal nuevo	Demuestra P(_)

Resumen DN

$I_\wedge$

$$\frac{\begin{array}{c}A\\ B\end{array}}{A \wedge B}$$

$E_\wedge$

$$\frac{A \wedge B}{B} \qquad \frac{A \wedge B}{A}$$

$I_\vee$

$$\frac{A}{B \vee A} \qquad \frac{A}{A \vee B}$$

$E_\vee$

$$\frac{\begin{array}{c}A \vee B\\ A \to C\\ B \to C\end{array}}{C}$$

$I_\to$

$$\frac{\begin{array}{c}A \text{ (supuesto)}\\ B\end{array}}{A \to B}$$

$E_\to$

$$\frac{\begin{array}{c}A \to B\\ A\end{array}}{B}$$

$I_\leftrightarrow$

$$\frac{\begin{array}{c}A \to B\\ B \to A\end{array}}{A \leftrightarrow B}$$

$E_\leftrightarrow$

$$\frac{A \leftrightarrow B}{A \to B} \qquad \frac{A \leftrightarrow B}{B \to A}$$

$I_\neg$

$$\frac{\begin{array}{c}A \text{ (supuesto)}\\ B \wedge \neg B\end{array}}{\neg A} \qquad \frac{A \to B \wedge \neg B}{\neg A}$$

$E_\neg$

$$\frac{\neg\neg A}{A}$$

$I_\exists$

$$\frac{A(t)}{\exists x A(x)}$$

$E_\exists$

$$\frac{\exists x A}{A\{x/a^*\}}$$

$I_\forall$

$$\frac{A(x)}{\forall x A}$$

$E_\forall$

$$\frac{\forall x A}{A\{x/t\}}$$

Derivadas

- $T[A \wedge \neg A] \vdash B$ Ex Contradictione Quodlibet
- $T[A \to B, B \to C] \vdash A \to C$ Transitividad
- $T[A \to B, \neg B] \vdash \neg A$ Modus Tollens
- $T[A \vee B, \neg A] \vdash B$Corte
- $T[A \vee B, \neg B] \vdash A$ Corte
- $T[A \vee B, \neg A \vee C] \vdash B \vee C$ Corte
- + regla de iteración
- + teorema del intercambio (y equivalencias)

CONDICIONES:

- *siendo a* un nombre temporal nuevo en la demostración*

CONDICIONES:

- *A(x) no incluye nombres temporales (a*)*
- *x no aparece libre en premisa, libre fuera de la subprueba en la que se aplica la regla, ni en ningún supuesto previo no cancelado*

18

Equivalencias lógicas LPO

Renombrado (si y no aparece libre en A)

- $\forall x A(x) \Leftrightarrow \forall y A(x/y)$
- $\exists x A(x) \Leftrightarrow \exists y A(x/y)$

Cuantificadores vs negación

- $\neg\forall x A(x) \Leftrightarrow \exists x \neg A(x)$
- $\neg\exists x A(x) \Leftrightarrow \forall x \neg A(x)$

Conectivas vs cuantificadores (x no aparece libre en C)

- $\forall x A \wedge C \Leftrightarrow \forall x(A \wedge C)$
- $\exists x A \wedge C \Leftrightarrow \exists x(A \wedge C)$
- $\forall x A \vee C \Leftrightarrow \forall x(A \vee C)$
- $\exists x A \vee C \Leftrightarrow \exists x(A \vee C)$
- $(\forall x A \rightarrow C) \Leftrightarrow \exists x(A \rightarrow C)$
- $(\exists x A \rightarrow C) \Leftrightarrow \forall x(A \rightarrow C)$
- $(A \rightarrow \forall x C) \Leftrightarrow \forall x(A \rightarrow C)$
- $(A \rightarrow \exists x C) \Leftrightarrow \exists x(A \rightarrow C)$
- $\forall x A \wedge \forall x C \Leftrightarrow \forall x(A \wedge C)$
- $(\exists x A \vee \exists x C) \Leftrightarrow \exists x(A \vee C)$

Ejemplo de uso en DN: "Intercambio 3, $\neg\forall x A(x) \Leftrightarrow \exists x \neg A(x)$".

No es válido: "equivalencia", "de Morgan", "$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$)", no incluir el número donde se aplica, no hacer mención al teorema del intercambio, etcétera

+las vistas en LP

- $\neg\neg A \Leftrightarrow A$
- $A \wedge A \Leftrightarrow A$
- $A \vee A \Leftrightarrow A$
- $A \wedge B \Leftrightarrow B \wedge A$
- $A \vee B \Leftrightarrow B \vee A$
- $A \leftrightarrow B \Leftrightarrow B \leftrightarrow A$
- $A \wedge (B \wedge C) \Leftrightarrow (A \wedge B) \wedge C$
- $A \vee (B \vee C) \Leftrightarrow (A \vee B) \vee C$
- $A \rightarrow B \Leftrightarrow \neg A \vee B$
- $A \rightarrow B \Leftrightarrow \neg B \rightarrow \neg A$
- $\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$
- $\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$
- $A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$
- $A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$
- $A \leftrightarrow B \Leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A)$

Ejemplo

Ejemplo 1: Consideramos las siguientes afimaciones: *La Tierra orbita en torno al Sol. La Luna orbita en torno a la Tierra. Todo cuerpo que orbita en torno al Sol es un planeta. Son satélites los cuerpos que orbitan en torno a planetas. Luego la Tierra es un planeta y la Luna un satélite.*

- $T[\ O(a,b),\ O(c,a),\ \forall x(O(x,b) \rightarrow P(x)),\ \forall x \forall y(O(x,y) \wedge P(y) \rightarrow S(x))\ ] \vdash P(a) \wedge S(c)$

1. $O(a,b)$	Premisa
2. $O(c,a)$	Premisa
3. $\forall x(O(x,b) \rightarrow P(x))$	Premisa
4. $\forall x \forall y(O(x,y) \wedge P(y) \rightarrow S(x))$	Premisa
5. $O(a,b) \rightarrow P(a)$	$E_{\forall\ 3}$
6. $P(a)$	$E_{\rightarrow 5,1}$
7. $\forall y(O(c,y) \wedge P(y) \rightarrow S(c))$	$E_{\forall 4}$
8. $O(c,a) \wedge P(a) \rightarrow S(c)$	$E_{\forall 7}$
9. $O(c,a) \wedge P(a)$	$I_{\wedge\ 2,6}$
10. $S(c)$	$E_{\rightarrow 8,9}$
11. $P(a) \wedge S(c)$	$I_{\wedge\ 7,11}$

Ejemplo

Ejemplo 2: Consideramos las siguientes afimaciones: *Todo elemento químico es oxidante o reductor. El carbono es un elemento químico no oxidante. Luego el carbono es reductor.*

- $T[\forall x(E(x) \rightarrow O(x) \vee R(x)), E(a) \wedge \neg O(a)] \vdash R(a)$

1. $\forall x(E(x) \rightarrow O(x) \vee R(x))$	Premisa
2. $E(a) \wedge \neg O(a)$	Premisa
3. $E(a)$	$E_{\wedge 2}$
4. $E(a) \rightarrow O(a) \vee R(a)$	$E_{\forall 1}$
5. $O(a) \vee R(a)$	$E_{\rightarrow 4,3}$
6. $\neg O(a)$	$E_{\wedge 2}$
7. $R(a)$	$\text{Corte}_{5,6}$

(también se puede hacer negando la conclusión y buscando una contradicción)

Ejercicios 1

- $\{\forall x(P(x) \to Q(x)), \exists xP(x)\} \vdash \exists xQ(x)$

1.	$\forall x(P(x) \to Q(x))$	Premisa
2.	$\exists xP(x)$	Premisa
3.	$P(a^*)$	$E_{\exists 2}$
4.	$P(a^*) \to Q(a^*)$	$E_{\forall 1}$
5.	$Q(a^*)$	$E_{\to 4,3}$
6.	$\exists xQ(x)$	$I_{\exists 5}$

Cuidado, ¡el orden importa! usa $E_\exists$ antes que $E_\forall$
Así se crea una nueva constante que podrás usar para $E_{\forall,}$ mientras que en el orden inverso no se podría.
(Pero en $\forall x\, \exists xA$, no es posible aplicar primero $E_\exists$)

Ejercicios 2

- $\forall x \forall y(P(y) \to Q(x)) \vdash \exists y P(y) \to \forall x Q(x)$

1.	$\forall x \forall y(P(y) \to Q(x))$	Premisa
2.	$\exists y P(y)$	Supuesto
3.	$P(a^*)$	$E_{\exists 2}$
4.	$\forall y(P(y) \to Q(x_0))$	$E_{\forall 1}$
5.	$P(a^*) \to Q(x_0)$	$E_{\forall 4}$
6.	$Q(x_0)$	$E_{\to 5,3}$
7.	$\forall x Q(x)$	$I_{\forall 6}$
8.	$\exists y P(y) \to \forall x Q(x)$	$I_{\to 2,7}$

Cuidado, si sustituyo x por una constante en 4, no podría aplicar nunca $I_\forall$
(Si buscas $\forall x A(x)$... nunca sustituyas la x de A por una constante)

Ejercicios 3

- $P(a) \rightarrow \forall x Q(x) \vdash \forall x (P(a) \rightarrow Q(x))$

1.	$P(a) \rightarrow \forall x Q(x)$	Premisa
2.	$P(a)$	Supuesto
3.	$\forall x Q(x)$	$E_{\rightarrow 1,2}$
4.	$Q(x_0)$	$E_{\forall 3}$
5.	$P(a) \rightarrow Q(x_0)$	$I_{\rightarrow 2,4}$
6.	$\forall x (P(a) \rightarrow Q(x))$	$I_{\forall 5}$

En 4 podría dejar $Q(x)$ sin x_0, pero aporta claridad

Ejercicios 4

- $\forall x \forall y \forall z[R(x, y) \wedge R(y, z) \rightarrow R(x, z)], \forall x \neg R(x, x) \vdash \forall x \forall y[R(x, y) \rightarrow \neg R(y, x)]$

1.	$\forall x \forall y \forall z[R(x, y) \wedge R(y, z) \rightarrow R(x, z)]$	Premisa
2.	$\forall x \neg R(x, x)$	Premisa
3.	$R(x_0,y_0)$	Supuesto
4.	$R(y_0,x_0)$	Supuesto (negación de $\neg R(y, x)$)
5.	$\neg R(x_0, x_0)$	$E_{\forall 2}$
6.	$\forall y \forall z[R(x_0, y) \wedge R(y, z) \rightarrow R(x_0, z)]$	$E_{\forall 1}$
7.	$\forall z[R(x_0, y_0) \wedge R(y_0, z) \rightarrow R(x_0, z)]$	$E_{\forall 6}$
8.	$R(x_0, y_0) \wedge R(y_0, x_0) \rightarrow R(x_0, x_0)$	$E_{\forall 7}$ (¿por qué {z/xo}?, fíjate en 5)
9.	$R(x_0,x_0)$	$E_{\rightarrow 3,4,8}$
10.	$R(x_0,x_0) \wedge \neg R(x_0, x_0)$	$I \wedge_{5,9}$
11.	$\neg R(x_0,y_0)$	$I_{\neg 4}$
12.	$R(x_0,y_0) \rightarrow \neg R(y_0,x_0)$	$I_{\rightarrow 3,11}$
13.	$\forall y[R(x_0, y) \rightarrow \neg R(y, x_0)]$	$I_{\forall 12}$
14.	$\forall x \forall y[R(x, y) \rightarrow \neg R(y, x)]$	$I_{\forall 13}$

Hay demostraciones alternativas sin suponer la negación de R(y,x), modus tollen de 5 y 8 para luego aplicar corte con 3

Examen DN Julio de 2015

Ejercicio 3. Demostrar la fórmula $\exists x\,(q(x) \rightarrow p(f(a),x))$ a partir del conjunto de premisas

$\{\ \forall x\,(\neg\, q(x) \vee r(x))\ ,\ \ \forall x\, \exists z\,(\neg\, p(x,z) \rightarrow \neg\, r(z))\ \}$

utilizando el cálculo de **Deducción Natural**. *(2,5 puntos)*

1.	$\forall x\exists z(\neg p(x,z)\rightarrow\neg r(z))$	premisa
2.	$\exists z(\neg p(f(a),z)\rightarrow\neg r(z))$	∀-elim (1)
3.	$\neg p(f(a),c)\rightarrow\neg r(c)$	∃-elim(2)
4.	$\quad q(c)$	supuesto
5.	$\quad \forall x(\neg q(x)\vee r(x))$	premisa
6.	$\quad \neg q(c)\vee r(c)$	∃-elim(5)
7.	$\quad \neg\neg q(c)$	intercambio(4)
8.	$\quad r(c)$	corte(6,7)
9.	$\quad \neg\neg r(c)$	intercambio(8)
10.	$\quad \neg\neg p(f(a),c)$	MT(3,9)
11.	$\quad p(f(a),c)$	¬-elim(10)
12.	$q(c)\rightarrow p(f(a),c)$	→-intro(4-11)
13.	$\exists x(q(x)\rightarrow p(f(a),x))$	∃-intro(12)

¡Deja claro cuando empieza y termina un supuesto!, tabulaciones, llaves, etcétera.

Los supuestos son siempre estratégicos (no se hacen sin saber el objetivo) y deben cerrarse cuando se encuentra ese objetivo.

Aquí uno sabe por la conclusión que se hará supuesto de q(una constante "uc"), buscará p(f(a),uc) dentro del supuesto, y así podrá cerrar el supuesto con q(uc) → p(f(a), uc), y después introducir el existencial.

La segunda premisa tiene un existencial, interesa eliminarlo antes de un supuesto para poder suponer la constante temporal añadida. (Puede que uno se de cuenta al avanzar en la demostración)

Además, uno puede ver en esa premisa, que interesará sustituir la x por f(a), ya que se busca p(f(a),"una constante". Ver estrategia.

¡Las reglas se aplican a rajatabla!, 7 es necesario para el corte en 8, 9 es necesario para el MT en 10. Hacer una simplificación no explícita penalizaría en el ejercicio.

También es importante anotar las sustituciones realizadas. En 2 se añadiría {x/f(a)}, en 5 se añadiría {z/c}

IMPORTANTE: Siempre que uses una equivalencia, como en puntos 7 y 9, pon:

Intercambio, nº donde se aplica, formula con metavariables. Omitir algo de esto penalizaría. Ejemplo: **Intercambio 3, ¬¬A ⇔ A**

Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingenieros Informáticos

Lógica
Tema 11: Forma Clausular

Profesor: Emilio Serrano
emilioserra@fi.upm.es

Temas

❑ Bloque II

Representación

Tema 7
Lenguajes de primer orden: sintaxis y uso en la formalización de argumentos

Tema 8
Semántica formal: Funciones de verdad, tautologicidad, consecuencia lógica

Razonamiento

Tema 9
Razonamiento semántico: definición de modelos y contra-modelos

Tema 10
Cálculo de deducción natural de primer orden

Computación

Tema 11
Forma Normal de Skolem. Forma clausular

Tema 12
Sustitución y Unificación máximamente general

Tema 13
Cálculo de resolución con Unificación

Introducción a la demostración automática

- El **razonamiento automático** se dedica a estudiar cómo usar un ordenador para ayudar en la parte de resolución de problemas que requiere razonamiento.
- Se trata de implementar programas que verifiquen un razonamiento mediante una serie de pasos de inferencia.
- Se suele denominar **deducción automática** porque se suele utilizar razonamiento como proceso deductivo.
- Cuando el trabajo se centra en la obtención de algoritmos que permitan encontrar pruebas de teoremas matemáticos, recibe el nombre de ***demostración automática de teoremas*** **(DAT)**.

- Los D.A.T.:
    - Hacen uso de una representación especial de las formulas: la **forma clausal o clausular**.
    - En la deducción se emplean reglas de inferencia alejadas del razonamiento humano, como **resolución** y para-modulación.
    - Por ultimo, destacar que en la mayoría de los casos, los demostradores automáticos de teoremas realizan **pruebas por contradicción (o refutación)**.

Estandarización de fórmulas

- ¿Cómo trabajar con DAT?
- Para trabajar con DAT es necesario trabajar con **fórmulas estandarizadas**.

- Objetivo: **simplificar las fórmulas**

  Queremos obtener, mediante una serie de ***transformaciones***, una fórmula que sea más fácil de manipular automáticamente, pero que siga teniendo ciertas propiedades de la fórmula original (estandarización):

Fórmula en un lenguaje de primer orden	$\exists y\ (\ \forall x\ (p(x,f(y)) \rightarrow q(z)) \vee \neg\exists w\ p(g(w),y))$
⬇	⬇
Fórmula en forma normal de Skolem	$\forall x\ \forall w\ (\neg p(x,f(b)) \vee q(a) \vee \neg p(g(w),b))$
⬇	⬇
Fórmula en forma clausular	$\{\neg p(x,f(b)) \vee q(a) \vee \neg p(g(w),b)\}$

Estandarización de fórmulas

- ¿Qué es lo que preserva esta transformación?
  - preserva la **satisfacibilidad**
  - pero no preserva todos los modelos (es decir, la semántica): el resultado **<u>no</u>** es equivalente a la fórmula original

- Preservación

  Consideremos una transformación de *F* a *F′*
  - preservar la **semántica** significa que, para toda interpretación *I*, *I* es un modelo de *F* sii es un modelo de *F′*

    $\forall x p(x)$ es semánticamente equivalente a $\neg\exists x \neg p(x)$
  - preservar la **satisfacibilidad** significa que existe un modelo *I* de *F* sii existe un modelo *I′* (probablemente no el mismo) de *F′*

    $SAT(\exists x p(x))$ sii $SAT(p(a))$

Forma Normal de Skolem

- **Formas Normales.** El objetivo de la estandarización de fórmulas es reducir la variedad sintáctica de un LPO, uniformando sus fórmulas.
- La idea es que la **fórmula inicial es satisfacible sii su transformada es satisfacible**.
    - <u>Reducción de la multiplicidad de conectivas</u> (formas normales en la lógica proposicional).
        - Forma normal conjuntiva. FNC
        - Forma normal disyuntiva. FND
    - <u>Uniformización de la posición de los cuantificadores en las fórmulas</u> (Formas Normales en la LPO).
        - Forma Prenex. FNP
    - <u>Cierre de fórmulas abiertas</u> (cuantificación existencial de variables libres).
    - <u>Eliminación de cuantificadores existenciales</u>.
        - Forma de Skolem. FNS
    - <u>Forma Clausular</u> (variante sintáctica de la Forma Skolem).

Forma Normal de Skolem

- **Forma Normal Disyuntiva.** Una fbf A está en forma normal disyuntiva si y solo si $A \equiv A_1 \vee A_2 \vee A_3 \vee \ldots \vee A_n$ con $n \geq 1$, y cada A_i es una conjunción de literales.

  Ejemplo: $(p \wedge \neg q \wedge r) \vee \neg p \vee (r \wedge \neg s)$

- **Forma Normal Conjuntiva.** Una fbf A está en forma normal conjuntiva si y solo si $A \equiv A_1 \wedge A_2 \wedge A_3 \wedge \ldots \wedge A_n$ con $n \geq 1$, y cada A_i es una disyunción de literales.

  Ejemplo: $(q \vee p) \wedge (\neg r \vee s) \wedge \neg q$

- **Forma normal prenexa:** Una fbf A está en forma normal prenexa si y solo si $A \equiv (Q_1 X_1)(Q_2 X_2)\ldots(Q_n X_n)M$, donde $X_1, X_2 \ldots X_n$ son variables diferentes y $Q_i \equiv \exists$ o $Q_i \equiv \forall$, para cada $1 \leq i \leq n$. M es una fórmula que no contiene cuantificadores. Al componente $(Q_1 X_1)(Q_2 X_2)\ldots(Q_n X_n)$ se le llama prefijo, y a M matriz de la fórmula A.

Forma Normal de Skolem

❑ **Forma normal prenexa**.
Ejemplos: Están las siguientes fórmulas en FNP?

- $\neg\forall x\,[P(x) \rightarrow \exists x\,P(x)]$ no
- $\exists x\,\forall y\,[P(x) \wedge \neg P(y)]$ si
- $\exists x\,P(x) \vee \forall y\,Q(y)$ no
- $\exists x\,\forall y\,[P(x) \vee Q(y)]$ si
- $\forall y\,\exists x\,[P(x) \vee Q(y)]$ si
- $\neg(\exists x\,[P(x) \rightarrow Q(x)] \rightarrow \exists x\,[P(x) \rightarrow R(x)])$ no
- $\forall z\,\exists x\,\exists y\,[((\neg P(x) \vee Q(x)) \wedge (\neg Q(y) \vee R(y))) \wedge P(z)]$ si

Born	23 May 1887 Sandsvær, Norway
Died	23 March 1963 (aged 75) Oslo, Norway
Residence	Norway
Nationality	Norwegian
Fields	Mathematician
Institutions	Oslo University Chr. Michelsen Institute
Alma mater	Oslo University
Doctoral advisor	Axel Thue
Doctoral students	Øystein Ore
Known for	Skolem-Noether theorem

Thoralf Skolem
1887-1963

Australia's Commonwealth Scientific and Industrial Research Organization picked up $229-million from technology companies for "violating" its Wi-Fi patent.

Forma Normal de Skolem

❑ **Forma Normal de Skolem**.

- Def.: La fórmula F está en **forma de Skolem** (FS) si es de la forma $\forall x_1 \ldots \forall x_n\, G$, donde $n > 0$ y G no tiene cuantificadores.

Es decir:
- Todos los cuantificadores a la cabeza de la fórmula (forma Prenex).
- No hay variables libres.
- Sólo hay cuantificadores universales.
- Matriz de la fórmula en forma normal conjuntiva (FNC, conjunción de disyunciones de literales).

Ejemplos:
- $\forall x \exists y P(x, y)$ no está en forma de Skolem
- $\forall x P(x, f(x))$ sí está en forma de Skolem
- $\exists x Q(x)$ no está en forma de Skolem
- $Q(a)$ sí está en forma de Skolem

Forma Normal de Skolem

- **Procedimiento para hallar FNS(A)**:

1. Dada una fórmula *A*, poner la fórmula en forma Prenex.

   Prenex (A)

2. Realizar el cierre existencial.

   $Cierre_{\exists}(Prenex(A)) = QM$, donde *Q* y *M* son el prefijo y la matriz cuantificacionales de la fórmula, respectivamente.

3. Poner la fórmula en forma normal conjuntiva.

   Obtención de *FNC(M)*

4. Eliminar los cuantificadores existenciales.

   Obtención de *Skolem(QFNC(M)) = FNS(A)*

Forma Normal de Skolem

1. **Forma normal prenexa.** Aplicando a una fórmula los siguientes pasos se obtiene otra fórmula equivalente y que está en forma normal prenexa:

   1) Rectificar la fórmula usando las equivalencias. Cambio de nombre de variables ligadas:

   $$\vdash \forall x A(x) \leftrightarrow \forall y A(x/y)$$
   $$\vdash \exists x A(x) \leftrightarrow \exists y A(x/y)$$

   donde *y* es una variable que no ocurre libre en *A*.

   (no deben haber varios cuantificadores distintos con la misma variable)

   2) Interdefinición de cuantificadores (interiorizar negaciones usando equivalencias).

   $$\vdash \neg\forall x A(x) \leftrightarrow \exists x \neg A(x)$$
   $$\vdash \neg\exists x A(x) \leftrightarrow \forall x \neg A(x)$$

   (no puede haber cuantificadores con una negación)

Forma Normal de Skolem

3) Distribución de conectivas respecto a cuantificadores.

si x no está libre en la otra subfórmula

$$\vdash \forall xA \wedge C \leftrightarrow \forall x(A \wedge C)$$
$$\vdash \exists xA \wedge C \leftrightarrow \exists x(A \wedge C)$$

$$\vdash \forall xA \vee C \leftrightarrow \forall x(A \vee C)$$
$$\vdash \exists xA \vee C \leftrightarrow \exists x(A \vee C)$$

$\vdash (\forall xA \rightarrow C) \leftrightarrow \exists x(A \rightarrow C)$ *(recuerda:* $\forall xA \rightarrow C \equiv \neg\forall xA \vee C$*)*

$$\vdash (\exists xA \rightarrow C) \leftrightarrow \forall x(A \rightarrow C)$$
$$\vdash (A \rightarrow \forall xC) \leftrightarrow \forall x(A \rightarrow C)$$
$$\vdash (A \rightarrow \exists xC) \leftrightarrow \exists x(A \rightarrow C)$$

$\vdash \forall xA \wedge \forall xC \leftrightarrow \forall x(A \wedge C)$ *(si se aplica (1), no se usarán)*

$\vdash (\exists xA \vee \exists xC) \leftrightarrow \exists x(A \vee C)$ *(nota que $\exists$ con $\wedge$, o $\forall$ con $\vee$ no son validas)*

Lema: Para toda fórmula A, $\vdash A \leftrightarrow Prenex(A)$

Lema: La forma Prenex de una fórmula siempre existe, ***aunque puede no ser única***

Forma Prenex

- Recuerda que siempre existe una forma prenexa, pero puede haber varias distintas.
  - Ejemplo, partiendo de:

$A \equiv \exists x\, P(x) \vee \forall y\, \neg Q(y) \rightarrow \exists z\, P(a,z)$, y considerando $F \equiv P(x) \vee \neg Q(y) \rightarrow P(a,z)$.

  - Podemos obtener: $\forall x \exists y \exists z(F)$, $\exists z \forall x \exists y(F)$, $\exists z \exists y \forall x(F)$, $\exists y \exists z \forall x(F)$... el orden en el que se extraen los cuantificadores afectará a su vez a la eliminación de existenciales.

- También puede ser necesario exteriorizar cuantificadores antes de interiorizar negaciones: $\neg(\exists x\, \forall y\, \neg P(x,y) \rightarrow \forall y\, \exists x\, Q(x,y))$, pasa a: $\exists x\, \forall y\, \exists y'\, \forall x'\, \neg(\neg P(x,y) \rightarrow Q(x',y'))$.
  - En general las equivalencias se pueden aplicar en cualquier orden.

Forma Normal de Skolem

2. **Cierre Existencial.** Las variables libres de la fórmula se ligan existencialmente **poniendo el cuantificador correspondiente en cabeza de la fórmula**:

   Lema: Una fórmula $A(x)$ es satisfacible sii $\exists x A(x)$ es satisfacible

   - Ejemplo:

     $$\forall y \exists z (p(x) \wedge q(y) \rightarrow r(f(z), x))$$

   se transformaría en

     $$\exists x \forall y \exists z (p(x) \wedge q(y) \rightarrow r(f(z), x))$$

Forma Normal de Skolem

3. **Forma Normal Conjuntiva - FNC.** Se utilizarán los siguientes teoremas de equivalencia:

   - Interdefinición (Eliminar bicondicionales y condicionales usando la equivalencia):

     $$\vdash (A \rightarrow B) \leftrightarrow (\neg A \vee B)$$
     $$\vdash (A \leftrightarrow B) \leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A)$$

   - Leyes de De Morgan (interiorizar negaciones usando equivalencias):

     $$\vdash \neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$$
     $$\vdash \neg(A \vee B) \leftrightarrow \neg A \wedge \neg B$$

   - Distribución de $\vee$ y $\wedge$:

     $$\vdash A \wedge (B \vee C) \leftrightarrow (A \wedge B) \vee (A \wedge C) \quad \textit{(pueden ir en los dos sentidos)}$$
     $$\vdash A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C) \quad \textit{(A no es un literal)}$$

Lema: Para toda fórmula A, $\vdash A \leftrightarrow FNC(A)$
Lema: La forma normal conjuntiva de una fórmula siempre existe

Forma Normal Conjuntiva

- Distribución de $\vee$ y $\wedge$:

$$\vdash A \wedge (B \vee C) \leftrightarrow (A \wedge B) \vee (A \wedge C)$$ *(pueden ir en los dos sentidos)*

$$\vdash A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C)$$ *(A no es un literal)*

- ❑ Recuerda que: A, B, C no son literales.
    - ❑ Ejemplo, pasa a clausal: $(\neg p \wedge q) \vee (p \wedge \neg q)$, se parece mucho a la primera equivalencia...
    - ❑ Pero no se puede aplicar (no hay un “A” que extraer).
    - ❑ Se puede considerar cualquiera de las formulas conectadas por la disyunción como “A” y aplicar distributiva de la segunda equivalencia.
        - ❑ $A \vee (p \wedge \neg q) \leftrightarrow (A \vee (p \wedge \neg q)) \wedge (A \vee (p \wedge \neg q))$
- ❑ También que la distributiva (como cualquier equivalencia) va en los dos sentidos:
    - ❑ $(\neg p \wedge q) \vee (t \wedge \neg p)$, se parece mucho a la anterior, y se puede hacer así...
    - ❑ Pero permite “sacar $\neg p$” y dejar $\neg p \wedge (t \vee q)$ usando la primera equivalencia .
- ❑ Pasar a forma clausular se simplifica mucho si renombras formulas con A,B,C.

Forma Normal de Skolem

4. **Eliminación de cuantificadores existenciales.** Se elimina el cuantificador existencial sustituyendo la variable que ligaba por una función de Skolem o constante de Skolem.

- La función de Skolem será:
    - o Una función nueva en la fórmula, aplicada a todas la variables cuantificadas universalmente que aparecen antes que el cuantificador existencial a eliminar.
    - o Si no hay tales variables se utilizará una constante nueva en la fórmula para hacer la sustitución.
- Ejemplos:
    - o $\forall x \exists y(p(x) \to \neg q(y))$ se transformaría en $\forall x(p(x) \to \neg q(f(x))$
    - o $\exists x \forall z(q(x,z) \vee r(a,x))$ se transformaría en $\forall z(q(b,z) \vee r(a,b))$
    - o $\exists x \forall y \exists z(p(x) \wedge q(y) \to r(f(z), x))$ se transformaría en
      $\forall y(p(a) \wedge q(y) \to r(f(g(y)), a))$

Lema: Una fórmula A es satisfacible sii Skolem(A) es satisfacible.

Forma Normal de Skolem

- **Ejemplos de cálculo de FNS:**

<u>Ejemplo1</u>:

o F : $\exists x \forall y \forall z \exists u \forall v \exists w[P(x, y, z) \wedge Q(u, v) \wedge \neg R(w)]$

1.- La fórmula está en FNP.
2.- No hay variables libres.
3.- La fórmula está en FNC.
4.- Eliminar cuantificadores existenciales.

$\exists x$ no se encuentra precedido de cuantificadores universales, se sustituye por una constante *a*

$$\forall y \forall z \exists u \forall v \exists w[P(a, y, z) \wedge Q(u, v) \wedge \neg R(w)]$$

$\exists u$ y $\exists w$ están precedidos de cuantificadores universales. Serán sustituidos por fórmulas *f(y,z)* y *g(y,z,v)*

$$\forall y \forall z \forall v\ [P(a, y, z) \wedge Q(f(y,z), v) \wedge \neg R(g(y,z,v))]$$

Forma Normal de Skolem

Ejemplo2:

FNP, Cierre Existencia, FNC.
Elminar Existenciales:
y se sustituye por h(x)
w se sustituye por g(x,z)

o Sko($\forall$x $\exists$y $\forall$z $\exists$w [$\neg$P(a,w) $\vee$ Q(f (x), y)])

= Sko($\forall$x $\forall$z $\exists$w [$\neg$P(a,w) $\vee$ Q(f (x), h(x))])

= Sko($\forall$x $\forall$z [$\neg$P(a, g(x, z)) $\vee$ Q(f (x), h(x))])

= $\forall$x $\forall$z [$\neg$P(a, g(x, z)) $\vee$ Q(f (x), h(x))]

Ejemplo 3:

FNP, Cierre Existencia, FNC:

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

Transformamos a FNC

$\vdash$ A $\vee$ (B $\wedge$ C) $\leftrightarrow$ (A $\vee$ B) $\wedge$ (A $\vee$ C)

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

Se eliminan los cuantificadores existencias

Eliminar Existenciales:
y se sustituye por f(x)
x se sustituye por g(x)

= $\forall$x [($\neg$P(x,f(x)) $\vee$ R(x, f(x), g(x))) ^ (Q(x, g(x))) $\vee$ R(x, f(x), g(x)))]

Forma Normal de Skolem

- Ejemplo 4: $\forall x \forall y[\exists z P(x, y, z) \wedge (\exists u Q(x, u) \rightarrow \exists v Q(y, v))]$

  1. Forma prenexa: (Cambio de nombre, interiorizar negaciones, *extracción de cuantificadores*)

     $|$— $\exists x A \wedge C \leftrightarrow \exists x(A \wedge C)$

     - $\forall x \forall y[\exists z(P(x, y, z) \wedge (\exists u Q(x, u) \rightarrow \exists v Q(y, v)))]$
     - $\forall x \forall y \exists z[P(x, y, z) \wedge (\exists u Q(x, u) \rightarrow \exists v Q(y, v))]$

     $|$— $(\exists x A \rightarrow C) \leftrightarrow \forall x(A \rightarrow C)$

     - $\forall x \forall y \exists z[P(x, y, z) \wedge \forall u(Q(x, u) \rightarrow \exists v Q(y, v))]$

     $|$— $\forall x A \wedge C \leftrightarrow \forall x(A \wedge C)$

     - $\forall x \forall y \exists z[\forall u(P(x, y, z) \wedge (Q(x, u) \rightarrow \exists v Q(y, v)))]$
     - $\forall x \forall y \exists z \forall u[P(x, y, z) \wedge (Q(x, u) \rightarrow \exists v Q(y, v))]$

     $|$— $(A \rightarrow \exists x C) \leftrightarrow \exists x(A \rightarrow C)$

     - $\forall x \forall y \exists z \forall u[P(x, y, z) \wedge \exists v(Q(x, u) \rightarrow Q(y, v))]$

     $|$— $\exists x A \wedge C \leftrightarrow \exists x(A \wedge C)$

     - $\forall x \forall y \exists z \forall u[\exists v (P(x, y, z) \wedge (Q(x, u) \rightarrow Q(y, v)))]$
     - $\forall x \forall y \exists z \forall u \exists v[P(x, y, z) \wedge (Q(x, u) \rightarrow Q(y, v))]$

Forma Normal de Skolem

- Ejemplo 4: $\forall x \forall y[\exists z P(x, y, z) \wedge (\exists u Q(x, u) \rightarrow \exists v Q(y, v))]$

2. Cierre existencial: No hay variables libres, ya tenemos el cierre:
   - $\forall x \forall y \exists z \forall u \exists v[P(x, y, z) \wedge (Q(x, u) \rightarrow Q(y, v))]$
3. FNC (eliminar $\leftrightarrow$ y $\rightarrow$, interiorizar $\neg$ y distribuir $\vee$ y $\wedge$)
   - $\forall x \forall y \exists z \forall u \exists v[P(x, y, z) \wedge (\neg Q(x, u) \vee Q(y, v))]$

$\models (A \rightarrow B) \leftrightarrow (\neg A \vee B)$

4. Eliminación de cuantificadores existenciales:

z se sustituye por f(x,y)
v se sustituye por g(x,y,u)

   - $\forall x \forall y \forall u[P(x, y, f(x,y)) \wedge (\neg Q(x, u) \vee Q(y, g(x,y,u)))]$

Forma Normal de Skolem

- Ejercicio: Obtener la FNS de la siguiente fórmula.

$$\neg\exists x\forall y(\neg C(x) \vee (B(a) \wedge D(x, y))) \wedge \neg\exists x A(x)$$

- Ejercicio: Obtener la FNS de la siguiente fórmula.

$$\exists x(\exists y A(x, y) \rightarrow \neg\forall z(B(x, z) \wedge C(z)))$$

Forma Normal de Skolem

- **Teorema: Una fórmula F es satisfacible sii FNS(F) es satisfacible.**

1. F satisfacible sii Prenex(F) satisfacible (por lemas 1 y 5)
2. Prenex(F) satisfacible sii Cierre(Prenex(F)) satisfacible (por lema 2)
3. Sea Cierre(Prenex(F)) satisfacible de la forma Q.M, donde Q son los cuantificadores y M es la matriz de la fórmula. Entonces M satisfacible sii FNC(M) satisfacible (por lemas 3 y 5).
4. Pero también Q.M satisfacible sii Q.FNC(M) satisfacible, puesto que cuantifican las mismas variables en idéntica forma
5. Q.FNC(M) satisfacible sii Skolem(Q.FNC(M)) satisfacible (por lema 4). Pero Skolem(Q.FNC(M)) es precisamente FNS(F)
6. F satisfacible sii FNS(F) satisfacible (por silogismo 1, 2, 3, 4, 5)

Lema 1: Para toda fórmula A, $\vdash A \leftrightarrow Prenex(A)$
Lema 2: Una fórmula A(x) es satisfacible sii $\exists x A(x)$ es satisfacible
Lema 3: Para toda fórmula A, $\vdash A \leftrightarrow FNC(A)$
Lema 4: Una fórmula A es satisfacible sii Skolem(A) es satisfacible.
Lema 5: Si $\vdash F \leftrightarrow F'$ entonces: F satisfacible sii F' satisfacible

Forma Clausular

- Sabemos que “Una fórmula F es satisfacible sii FNS(F) es satisfacible” y que “FNS(F) existe siempre para cualquier fórmula F”, luego **podemos trabajar exclusivamente con fórmulas en forma normal de Skolem**.
- Para trabajar más cómodamente con fórmulas en FNS utilizaremos la forma clausular
- La mayoría de procedimientos de prueba operan por refutación. Estas pruebas se aplican a formulas en una forma denominada ***forma clausal***.

- **Cláusula:** Una cláusula <u>es una disyunción finita de cero o más literales</u>. Cuando la cláusula está compuesta de un solo literal diremos que es una cláusula unitaria.

$$C = L_1 \vee \ldots \vee L_n, \qquad L_i (1 \leq i \leq n) \text{ literal}$$

- La **forma clausular de una fórmula F, FC(F)**, es el conjunto de cláusulas de la FNS(F).
- La **forma clausular** se entiende como la conjunción de las cláusulas, cuyas variables están todas ellas ligadas universalmente.

Forma Clausular

- Una vez que se tiene la Forma Normal de Skolem, se eliminan los cuantificadores universales y se pasa a FNC (como si estuviésemos trabajando en lógica proposicional).

- Finalmente se escribe como conjuntos.

- Ejemplo 1:

$$\forall x(G(H,H) \wedge (\neg F(x) \vee G(x,U(x)) \vee \neg G(H, x)))$$

Se eliminan los $\forall$ y se pasa a FNC (en este caso ya está en FNC):

$$G(H,H) \wedge (\neg F(x) \vee G(x,U(x)) \vee \neg G(H, x))$$

luego se escribe como conjuntos:

$$\{\{G(H,H)\}, \{\neg F(x),G(x,U(x)),\neg G(H, x)\}\}$$

Forma Clausular

Ejemplo 2: $\forall x \exists y (A(x,y) \rightarrow B(x) \wedge C(y))$

Eliminación de la implicación y propiedad distributiva

- $\forall x \exists y (A(x,y) \rightarrow B(x) \wedge C(y)) \leftrightarrow \forall x \exists y (\neg A(x,y) \vee (B(x) \wedge C(y)))$
  $\leftrightarrow \forall x \exists y ((\neg A(x,y) \vee B(x)) \wedge (\neg A(x,y) \vee C(y)))$

Forma Normal de Skolem

- $FNS(\forall x \exists y ((\neg A(x,y) \vee B(x)) \wedge (\neg A(x,y) \vee C(y))))$

  $= \forall x ((\neg A(x,f(x)) \vee B(x)) \wedge (\neg A(x,f(x)) \vee C(f(x))))$

Forma Clausular

- $\{ \{\neg A(x,f(x)) \vee B(x)\}, \{\neg A(x,f(x) \vee C(f(x)) \} \}$

Forma Clausular

- Ejemplo 3:
  - A: $\forall x \forall y(p(x) \wedge (q(y) \vee r(a,x)))$

    $FC(A)$: $\{\{p(x)\}, \{q(y) \vee r(a,x)\}\}$

Teorema: Una fórmula F es satisfacible sii FC(F) es satisfacible

Forma Clausular

- **Forma Clausular de una deducción.**
- Una deducción [P1, P2, ..., Pn] $\vdash$ C es correcta sii P1 $\wedge$ P2 $\wedge$ ... $\wedge$ Pn $\wedge$ $\neg$C es insatisfacible

Dada una deducción: [P1, P2, ..., Pn] $\vdash$ C

1)Obtener la forma clausular de cada Pi, $1 \leq i \leq n$

2)Obtener la forma clausular de $\neg$C

3)Realizar la unión de todos los conjuntos de cláusulas

4)Comprobar la satisfacibilidad

Una deducción [P1, P2, ..., Pn] $\vdash$ C es correcta sii

FC(P1 $\wedge$ P2 $\wedge$ ... $\wedge$ Pn $\wedge$ $\neg$C) es insatisfacible

Ejercicio 1

Ejercicio 1: Dado el conjunto de fórmulas $\{A_1, A_2, A_3, A_4\}$, siendo:

A_1: $\forall z \exists x((\neg P(z) \vee \exists y Q(x, y, z)) \wedge (\neg R(z) \vee \neg P(x)))$

A_2: $\exists z S(z) \rightarrow \forall z T(z)$

A_3: $\forall z \forall x(T(z) \rightarrow P(x))$

A_4: $\exists z \forall y(\neg \exists x Q(z, x, y) \wedge S(y) \wedge R(z))$

donde x, y, z son símbolos de variable, expresarlas en forma clausular.

Ejercicio 1

A_1: $\forall z \exists x((\neg P(z) \vee \exists y Q(x, y, z)) \wedge (\neg R(z) \vee \neg P(x)))$

$\forall z \exists x(\exists y(\neg P(z) \vee Q(x, y, z)) \wedge (\neg R(z) \vee \neg P(x)))$

$\forall z \exists x \exists y((\neg P(z) \vee Q(x, y, z)) \wedge (\neg R(z) \vee \neg P(x)))$

$\forall z((\neg P(z) \vee Q(f(z), g(z), z)) \wedge (\neg R(z) \vee \neg P(f(z))))$

Prenex: ***$A \vee \exists xB) \leftrightarrow \exists x(A \vee B)$***

$(\exists xA \wedge B) \leftrightarrow \exists x(A \wedge B)$

Cierre $\exists$

FNC

Eliminación $\exists$ $\{x/f(z), y/g(z)\}$

A_2: $\exists z S(z) \rightarrow \forall z T(z)$

$\exists z' S(z') \rightarrow \forall z T(z)$

$\forall z' \forall z(S(z') \rightarrow T(z))$

$\forall z' \forall z(\neg S(z') \vee T(z))$

Prenex: ***Cambio de nombre***

$(\exists xA \rightarrow B) \leftrightarrow \forall x(A \rightarrow B)$

$(A \rightarrow \forall xB) \leftrightarrow \forall x(A \rightarrow B)$

FNC: ***$(A \rightarrow B) \leftrightarrow (\neg A \vee B)$***

Cierre $\exists$

Eliminación $\exists$

A_3: $\forall z \forall x(T(z) \rightarrow P(x))$

$\forall z \forall x(\neg T(z) \vee P(x))$

Prenex, Cierre $\exists$,

FNC: $(A \rightarrow B) \leftrightarrow (\neg A \vee B)$

Eliminación $\exists$

A_4: $\exists z \forall y(\neg \exists x Q(z, x, y) \wedge S(y) \wedge R(z))$

$\exists z \forall y(\forall x \neg Q(z, x, y) \wedge S(y) \wedge R(z))$

$\exists z \forall y \forall x(\neg Q(z, x, y) \wedge S(y) \wedge R(z))$

$\forall y \forall x(\neg Q(a, x, y) \wedge S(y) \wedge R(a))$

Prenex: ***$(\neg \exists xA) \leftrightarrow (\forall x \neg A)$***

Cierre $\exists$

FNC: $(A \rightarrow B) \leftrightarrow (\neg A \vee B)$

Eliminación $\exists$ $\{z/a\}$

$\{\{\neg P(z) \vee Q(f(z), g(z), z))\}, \{\neg R(z) \vee \neg P(f(z))\}, \{\neg S(z') \vee T(z)\}, \{\neg T(z) \vee P(x)\}, \{\neg Q(a, f(y), y)\}, \{S(y) \wedge R(a)\}\}$

Ejercicio 2

Ejercicio 2: Obtener la forma clausular de cada una de las fórmulas siguientes:

A_1: $\exists x(\exists y A(x, y) \rightarrow \neg \forall z(B(x, z) \wedge C(z)))$

A_2: $\neg(\exists x A(x) \wedge \neg \exists y \forall x C(x, y))$

Ejercicio 2, solución

A_1: $\exists x(\exists y A(x, y) \rightarrow \neg\forall z(B(x, z) \wedge C(z)))$

$\exists x\forall y(A(x, y) \rightarrow \neg\forall z(B(x, z) \wedge C(z)))$

$\exists x\forall y\, (A(x, y) \rightarrow \exists z\neg(B(x, z) \wedge C(z)))$

$\exists x\forall y\exists z(A(x, y) \rightarrow \neg(B(x, z) \wedge C(z)))$

$\exists x\forall y\exists z(\neg A(x, y) \vee \neg(B(x, z) \wedge C(z)))$

$\exists x\forall y\exists z(\neg A(x, y) \vee \neg B(x, z) \vee \neg C(z))$

$\forall y(\neg A(a, y) \vee \neg B(a, f(y)) \vee \neg C(f(y)))$

Prenex: $(\exists xA \rightarrow B) \leftrightarrow \forall x(A \rightarrow B)$

$(\neg\forall xA) \leftrightarrow (\exists x\neg A)$

$(A \rightarrow \exists xB) \leftrightarrow \exists x(A \rightarrow B)$ *Cierre ∃*

FNC: $(A \rightarrow B) \leftrightarrow (\neg A \vee B)$

$\neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$

Eliminación ∃ $\{x/a, z/f(y)\}$

A_2: $\neg(\exists xA(x) \wedge \neg\exists y\forall xC(x, y))$

$\neg\exists x(A(x) \wedge \neg\exists y\forall zC(z, y))$

$\forall x\neg(A(x) \wedge \neg\exists y\forall zC(z, y))$

$\forall x\neg(A(x) \wedge \forall y\neg\forall zC(z, y))$

$\forall x\neg(A(x) \wedge \forall y\exists z\neg C(z, y))$

$\forall x\neg\forall y(A(x) \wedge \exists z\neg C(z, y))$

$\forall x\neg\forall y\exists z(A(x) \wedge \neg C(z, y))$

$\forall x\exists y\neg\exists z(A(x) \wedge \neg C(z, y))$

$\forall x\exists y\forall z\neg(A(x) \wedge \neg C(z, y))$

$\forall x\exists y\forall z\, (\neg A(x) \vee C(z, y))$

$\forall x\forall z\, (\neg A(x) \vee C(z, g(x)))$

Prenex: Cambio de nombre,

$(\exists xA \wedge B) \leftrightarrow \exists x(A \wedge B)$

$(\neg\exists xA) \leftrightarrow (\forall x\neg A)$

$(\neg\exists xA) \leftrightarrow (\forall x\neg A)$

$(\neg\forall xA) \leftrightarrow (\exists x\neg A)$

$(A \wedge \forall xB) \leftrightarrow \forall x(A \wedge B)$

$(A \wedge \exists xB) \leftrightarrow \exists x(A \wedge B)$

$(\neg\forall xA) \leftrightarrow (\exists x\neg A)$

$(\neg\exists xA) \leftrightarrow (\forall x\neg A)$ *Cierre ∃*

FNC: $\neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$

Eliminación ∃ $\{y/g(x)\}$

Nota: g(x) porque f ya se usa en A1

Forma Clausular: $\{\neg A(a, y) \vee \neg B(a, f(y)) \vee \neg C(f(y)), \neg A(x) \vee C(z, f(x))\}$

Ejercicio 3

Ejercicio 3: Sea $\{A_1,A_2,A_3,A_4\}$ el conjunto de fórmulas siguiente:

A_1: $\exists x B(x)$
A_2: $\forall x \forall y (C(x, y) \rightarrow D(x) \wedge B(x))$
A_3: $\exists x \forall y C(x, y)$
A_4: $\neg \exists x \exists y (D(x) \wedge \neg A(y))$

Construir el conjunto de cláusulas correspondiente a las fórmulas anteriores.

Ejercicio 3, solución

A_1: ∃xB(x)

B(a)

Prenex, Cierre ∃, FNC
Eliminación ∃ {x/a}

A_2: ∀x∀y(C(x, y) → D(x) ∧ B(x))

∀x∀y(¬C(x, y) ∨ (D(x) ∧ B(x)))
∀x∀y((¬C(x, y) ∨ (D(x)) ∧ (¬C(x, y) ∨ B(x)))
(¬C(x, y) ∨ (D(x)) ∧ (¬C(x, y) ∨ B(x))

Prenex, Cierre ∃
FNC: (A → B) ↔ (¬A ∨ B)
(A ∨ (B ∧ C)) ↔ (A ∨ B) ∧ (A ∨ C)
Eliminación ∃

A_3: ∃x∀yC(x, y)

∃x∀yC(x, y)
C(b, y)

Prenex, Cierre ∃, FNC
Eliminación ∃
Nota: constante b porque a ya se usa en A1

A_4: ¬∃x∃y(D(x) ∧ ¬A(y))

∀x¬∃y(D(x) ∧ ¬A(y))
∀x∀y¬(D(x) ∧ ¬A(y))
∀x∀y(¬D(x) ∨ ¬¬A(y)))
¬D(x) ∨ A(y)

Prenex: (¬∃xA) ↔ (∀x¬A)
(¬∃xA) ↔ (∀x¬A)
Cierre ∃
FNC: ¬(A ∧ B) ↔ (¬A ∨ ¬B)
Eliminación ∃

{{B(a)}, {¬C(x, y) ∨ (D(x)}, {¬C(x, y) ∨ B(x)}, {C(a, y)}, {¬D(x) ∨ A(y)}}

Ejercicio 4

Ejercicio 4: Sea $\{A_1, A_2, A_3, A_4\}$ el conjunto de fórmulas siguiente:

A_1: $R(b) \wedge G(b) \wedge R(a)$
A_2: $\forall x(M(x) \rightarrow (G(x) \rightarrow P(x)) \wedge (\neg G(x) \rightarrow \neg P(x)))$
A_3: $\forall x(R(x) \rightarrow M(x))$
A_4: $\neg \exists x(R(x) \wedge \neg D(x))$

Construir el conjunto de cláusulas correspondientes a las fórmulas anteriores.

Ejercicio 4, solución

A_1: $R(b) \wedge G(b) \wedge R(a)$

Prenex, Cierre ∃ , FNC, Eliminación ∃

A_2: $\forall x(M(x) \rightarrow (G(x) \rightarrow P(x)) \wedge (\neg G(x) \rightarrow \neg P(x)))$

$\forall x(\neg M(x) \vee (G(x) \rightarrow P(x)) \wedge (\neg G(x) \rightarrow \neg P(x)))$

$\forall x((\neg M(x) \vee \neg G(x) \vee P(x)) \wedge (\neg G(x) \rightarrow \neg P(x)))$

$\forall x((\neg M(x) \vee \neg G(x) \vee P(x)) \wedge (\neg\neg G(x) \vee \neg P(x)))$

$\forall x((\neg M(x) \vee \neg G(x) \vee P(x)) \wedge (G(x) \vee \neg P(x)))$

Prenex, Cierre ∃
FNC : $(A \rightarrow B) \leftrightarrow (\neg A \vee B)$

Eliminación ∃

A_3: $\forall x(R(x) \rightarrow M(x))$

$\forall x(\neg R(x) \vee M(x))$

Prenex, Cierre ∃
FNC : $(A \rightarrow B) \leftrightarrow (\neg A \vee B)$

Eliminación ∃

A_4: $\neg \exists x(R(x) \wedge \neg D(x))$

$\forall x \neg(R(x) \wedge \neg D(x))$

$\forall x\ (\neg R(x) \vee \neg\neg D(x))$

$\forall x\ (\neg R(x) \vee D(x))$

Prenex: $(\neg \exists x A) \leftrightarrow (\forall x \neg A)$

Cierre ∃
FNC: $\neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$

Eliminación ∃

Forma Clausular:
$\{\{R(b)\},\{G(b)\},\{R(a)\},\{\neg M(x) \vee \neg G(x) \vee P(x)\},\{G(x) \vee \neg P(x)\},\{\neg R(x) \vee M(x)\},\{\neg R(x) \vee D(x)\}\}$

Lógica
Tema 12: Sustitución y Unificación

Profesor: Emilio Serrano
emilioserra@fi.upm.es

Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingenieros Informáticos

Temas

Bloque II

Tema 7
Lenguajes de primer orden: sintaxis y uso en la formalización de argumentos

Tema 8
Semántica formal: Funciones de verdad, tautologicidad, consecuencia lógica

Representación

Tema 9
Razonamiento semántico: definición de modelos y contra-modelos

Tema 10
Cálculo de deducción natural de primer orden

Razonamiento

Tema 11
Forma Normal de Skolem. Forma clausular

Tema 12
Sustitución y Unificación máximamente general

Tema 13
Cálculo de resolución con Unificación

Computación

Sustitución (recordatorio)

- **Sustitución:** Una *sustitución* es una función finita de un conjunto de variables de un lenguaje en el de términos. Se representa como $\alpha = \{x_1/t_1, x_2/t_2, \ldots, x_n/t_n\}$ donde $x_1, \ldots, x_n$ son variables diferentes y $t_1, \ldots, t_n$ son términos tales que, en cada t_i no aparece la variable x_i
- Un par x_i/t_i se denomina *ligadura*
- Una sustitución que no sustituye ninguna variable se llama *sustitución vacía (λ)*
- Una sustitución que sustituye variables por otras variables se denomina *renombrado*
- *Dominio(α)* = $\{x_i \mid x_i/t_i \in \alpha\}$
- *Rango(α)* = $\{y_i \mid y_i$ aparece en t_i y $x_i/t_i \in \alpha\}$

Ejemplos: Ctes = {a, b, c, d}, Var = {x, y, z, w}, Func = {f/1, h/2}

$\alpha1 = \{x/f(a), y/x, z/h(b,y), w/a\}$	Dominio($\alpha1$) = {x, y, z, w}	Rango($\alpha1$) = {x, y}
$\alpha2 = \{x/a, y/a, z/h(b,c), w/f(d)\}$	Dominio($\alpha2$) = {x, y, z, w}	Rango($\alpha2$) = $\emptyset$
$\alpha3 = \{x/y, z/w\}$ (renombrado)	Dominio($\alpha3$) = {x, z}	Rango($\alpha3$) = {y, w}
$\lambda = \{x/x, y/y, z/z\}$		

Sustitución (recordatorio)

- Dada una fórmula F y una sustitución $\alpha = \{x_1/t_1, \ldots, x_n/t_n\}$, se denomina *aplicación de α a F, (Fα)* a la fórmula obtenida reemplazando ***simultáneamente*** cada ocurrencia en F de x_i por t_i, para cada $x_i/t_i \in \alpha$.

  α = {x/f(a), y/x, z/h(b,y), w/a}

  P(x, y, z) α = P(f(a), f(a), h(b,f(a))) *incorrecto*

  P(x, y, z) α = P(f(a), x, h(b,y)) *correcto*

- Una fórmula F' es *instancia* de otra fórmula F si existe una sustitución no vacía α tal que F' = Fα

- Una sustitución α es *idempotente* si Dominio(α) $\cap$ Rango(α) = $\varnothing$

- Si α es una sustitución idempotente entonces (Fα)α = Fα

  α1 = {x/a, y/f(b), z/v} P(x, y, w, z) α1 = P(a, f(b), w, v)

  P(a, f(b), w, v) α1 = P(a, f(b), w, v)

  α2 = {x/a, y/f(b), z/x} P(x, y, w, z) α2 = P(a, f(b), w, x)

  P(a, f(b), w, x) α2 = P(a, f(b), w, a)

  α1 es idempotente, α2 no

Sustitución (recordatorio)

- Dadas dos sustituciones α = $\{x_1/t_1, \ldots, x_n/t_n\}$ y β= $\{y_1/s_1, \ldots, y_m/s_m\}$ su *composición $\alpha\beta$* se define eliminando del conjunto

$$\{x_1/t_1\beta, \ldots, x_n/t_n\beta, y_1/s_1, \ldots, y_m/s_m\}$$

las ligaduras $x_i/t_i\beta$ tales que $x_i \equiv t_i\beta$,

y las ligaduras y_i/s_i tales que $y_i \in \{x_1,\ldots,x_n\}$

- Ejemplo:

si α = $\{x/3, y/f(x,1)\}$ y β= $\{x/4\}$

entonces $\alpha\beta = \{x/3, y/f(4,1)\}$ y

$\beta\alpha = \{x/4, y/f(x,1)\}$

- Propiedades de la composición:

$(F\alpha)\beta = F(\alpha\beta)$

$(\alpha\beta)\gamma = \alpha(\beta\gamma)$

$\alpha \lambda = \lambda \alpha = \alpha$

$\alpha\beta \neq \beta\alpha$

Unificación

- Una sustitución α es un *unificador* de dos fórmulas A y B si $A\alpha = B\alpha$. En este caso se dice que A y B son unificables. Informalmente, unificar es el proceso por el cual dos o mas expresiones se convierten en idénticas mediante una sustitución: la sustitución unificadora.
- Un unificador α de A y B se denomina *unificador de máxima generalidad (umg)* sii para cualquier otro unificador β de A y B existe alguna sustitución γ tal que $\beta=\alpha\gamma$
- Si dos fórmulas son unificables entonces tienen umg
- El umg de dos fórmulas es único (salvo renombrado)
- Ejemplo:

  $A \equiv P(x, f(x, g(y)), z)$ y $B \equiv P(r, f(r, u), a)$

  $\alpha 1 = \{x/r, u/g(y), z/a\}$ y $\alpha 2 = \{x/a, r/a, y/b, u/g(b), z/a\}$

  $A\alpha 1 = B\alpha 1 = P(r, f(r, g(y)), a)$

  $A\alpha 2 = B\alpha 2 = P(a, f(a, g(b)), a)$

  $\alpha 1$ y $\alpha 2$ son unificadores de A y B, pero $\alpha 1$ es el umg de A y B

  $\gamma = \{r/a, y/b\}$, $\alpha 2 = \alpha 1\, \gamma$

Unificación

- **Algoritmo de Unificación:**

Sean A y B dos átomos con el mismo símbolo de predicado:

(1) $\alpha = \lambda$

(2) Mientras $A\alpha \neq B\alpha$:

(2.1) Encontrar el símbolo más a la izquierda en $A\alpha$ tal que el símbolo correspondiente en $B\alpha$ sea diferente

(2.2) Sean t_A y t_B los términos de $A\alpha$ y $B\alpha$ que empiezan con esos símbolos:

(a) Si ni t_A ni t_B no son variables o, si uno de ellos es una variable que aparece en el otro ➔ terminar con fallo (A y B no son unificables)

(b) En otro caso, sea t_A una variable ➔ el nuevo α es el resultado de $\alpha\{t_A/t_B\}$ (se hace composición, no sólo se añade la nueva ligadura)

(3) Terminar, siendo α el umg de A y B

Unificación

- **Algoritmo de Unificación:**

Ejemplo: A ≡ P(x, x) y B ≡ P(f(a), f(b))

α	A α	B α	(t_A, t_B)
λ	P(x, x)	P(f(a), f(b))	(x,f(a))
{x/f(a)}	P(f(a), f(a))	P(f(a), f(b))	(a,b)

Fallo ➔ A y B no son unificables

Ejemplo: A ≡ P(x, f(y)) y B ≡ P(z, x)

α	A α	B α	(t_A, t_B)
λ	P(x, f(y))	P(z, x)	(x,z)
{x/z}	P(z, f(y))	P(z, z)	(z, f(y))
{x/f(y), z/f(y)}	P(f(y), f(y))	P(f(y), f(y))	

➔ A y B son unificables y su umg es {x/f(y), z/f(y)}

Lógica
Tema 13: Cálculo de Resolución con Unificación

Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingenieros Informáticos

Profesor: Emilio Serrano
emilioserra@fi.upm.es

Temas

❑ Bloque II

Tema 7
Lenguajes de primer orden: sintaxis y uso en la formalización de argumentos

Tema 8
Semántica formal: Funciones de verdad, tautologicidad, consecuencia lógica

Representación

Tema 9
Razonamiento semántico: definición de modelos y contra-modelos

Tema 10
Cálculo de deducción natural de primer orden

Razonamiento

Tema 11
Forma Normal de Skolem. Forma clausular

Tema 12
Sustitución y Unificación máximamente general

Tema 13
Cálculo de resolución con Unificación

Computación

Resolución con Variables

- **Regla de resolución con umg:** Sean $L_1 \vee \ldots \vee L_n \vee C_1$ y $\neg L_1' \vee \ldots \vee \neg L_m' \vee C_2$ dos cláusulas, donde todos los L_{ij} son literales con el mismo símbolo de predicado. Puede deducirse una nueva cláusula $(C_1 \rho_1 \vee C_2 \rho_2)\beta$, llamada *resolvente*, donde
  - ρ_1 y ρ_2 son renombrados de todas las variables de cada cláusula que garantizan la no repetición de nombres entre ellas.
  - β es umg de $\{L_1 \rho_1, \ldots, L_n \rho_1, L_1'\rho_2, \ldots, L_m'\rho_2\}$

- La regla de resolución con umg se apoya en una versión de la **regla de factorización** para LPO: Dada una cláusula $L_1 \vee \ldots \vee L_n \vee C$, siendo $L_1, \ldots, L_n$ literales con el mismo símbolo de predicado, puede deducirse una nueva cláusula $L \vee C\beta$ donde
  - β es unificador de $L_1, \ldots, L_n$
  - $L = L_1\beta = \ldots = L_n\beta$

  Observa que la factorización se aplica a una misma clausula, ejemplo
  $\neg M(y_2, z_2, u) \vee \neg P(x_3) \vee \neg N(x_3, u) \vee \mathbf{N(x_3, y_2)} \vee \mathbf{N(x_3, z_2)}$,
  Podría quedar con $\{z_2 / y_2\}$ y factorización como
  $\neg M(y_2, \mathbf{y_2}, u) \vee \neg P(x_3) \vee \neg N(x_3, u) \vee N(x_3, y_2)$

  El literal L se denomina *factor* de $L_1 \vee \ldots \vee L_n \vee C$
  - La factorización nos permite eliminar información, pero es optativa. Los resolventes se pueden obtener con o sin ella.
  - Al factorizar, si los literales tienen las mismas variables, no se pueden cambiar los nombres.
  - En ocasiones aplicar factorización puede hacer la derivación imposible.
  - En ocasiones NO aplicar factorización puede hacer la derivación de □ imposible.

Resolución con Variables

- La regla de resolución con umg es correcta

  Si por su aplicación deducimos □, entonces el conjunto inicial de cláusulas es insatisfacible *(demostración en el anexo al final del tema).*

- La regla de resolución con umg es completa

  Si el conjunto inicial es insatisfacible, entonces podemos asegurar que con la aplicación sucesiva de la regla de resolución llegaremos a deducir la cláusula vacía *(demostración en el anexo al final del tema).*

- **Un conjunto de cláusulas es insatisfacible sii se puede deducir □ a partir de él por resolución con umg** *(demostración en el anexo al final del tema).*

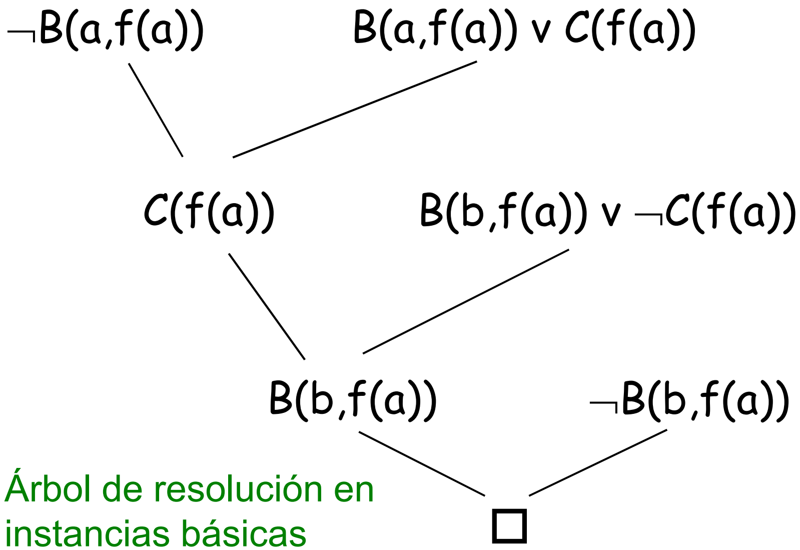
- Por tanto, **el método general de insatisfacibilidad se puede reducir a la búsqueda de □** a partir del conjunto de cláusulas, en lugar de tener que generar conjuntos de instancias básicas

Resolución con Variables

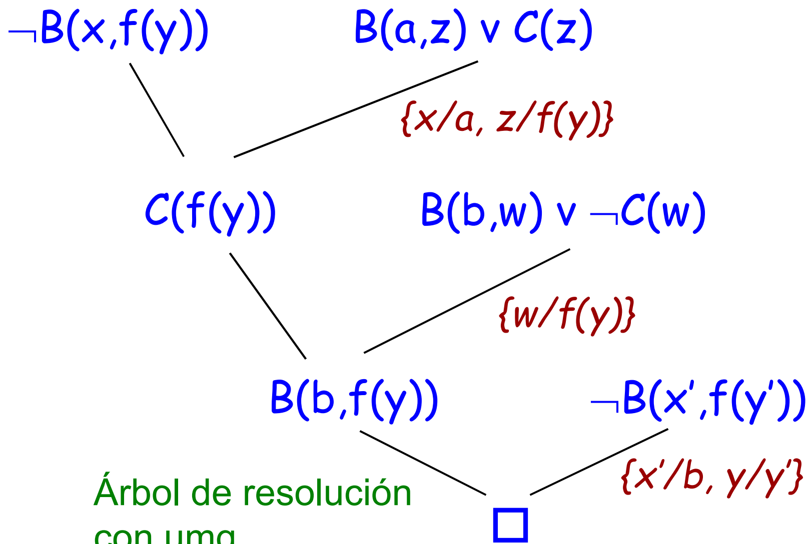
- Pueden construirse árboles de resolución en los que los resolventes de cada dos cláusulas se obtienen en un paso de resolución con umg.
- Por cada paso de resolución en instancias básicas puede definirse un paso de resolución con umg.

Ejemplo: C_1: ¬B(x,f(y)) C_2: B(a,z) v C(z) C_3: B(b,w) v ¬C(w)

I_1: ¬B(a,f(a)) I_1': ¬B(b,f(a)) I_2: B(a,f(a)) v C(f(a)) I_3: B(b,f(a)) v ¬C(f(a))



Árbol de resolución en instancias básicas



Árbol de resolución con umg

Procedimiento de Saturación

- En el ejemplo anterior la búsqueda de una refutación se ha realizado de modo arbitrario, sin embargo, es necesaria una sistematización para esta búsqueda. El modo más simple (conceptualmente hablando) de sistematizar la búsqueda de una refutación o bien la de asegurar que no existe, se basa en la generación sucesiva de todas las posibles resolventes a partir del conjunto de partida.

- **Procedimiento de saturación:** Sea C un conjunto de cláusulas
  1) Sea $S_0 = C$ y $n = 0$
  2) Si $\square \in S_n \Rightarrow$ C es insatisfacible
  3) Construir $S_{n+1} = \{$resolventes de C1 y C2 / $C1 \in (S_0 \cup \ldots \cup S_n)$, $C2 \in S_n\}$
  4) Si $S_{n+1} = \varnothing$ o $S_{n+1} \subset S_0 \cup \ldots \cup S_n \Rightarrow$ C es satisfacible
  5) Hacer $n = n+1$ y repetir desde 2)

- El paso 3) requiere considerar todos los posibles factores F, de predicados distintos, de las cláusulas C1 y C2 sobre los que se pueden resolver ambas cláusulas con el umg que da F.

Procedimiento de Saturación

- Este procedimiento genera *todos* y *sólo* los resolventes posibles a partir de un conjunto de cláusulas.
- Este procedimiento es completo: Un conjunto de cláusulas C es insatisfacible sii el procedimiento de saturación encuentra □ a partir de C.
- El proceso de saturación va a generar multitud de clausulas innecesarias para la obtención de la clausula vacía, por lo que el proceso se puede y se debe depurar.

Procedimiento de Saturación

S_0:	1)	$P \vee Q$	
	2)	$\neg P \vee Q$	
	3)	$P \vee \neg Q$	
	4)	$\neg P \vee \neg Q$	
S_1:	5)	Q	de 1) y 2)
	6)	P	de 1) y 3)
	7)	$Q \vee \neg Q$	de 1) y 4)
	8)	$P \vee \neg P$	de 1) y 4)
	9)	$Q \vee \neg Q$	de 2) y 3)
	10)	$P \vee \neg P$	de 2) y 3)
	11)	$\neg P$	de 2) y 4)
	12)	$\neg Q$	de 3) y 4)
S_2:	13)	$P \vee Q$	de 1) y 7)
	14)	$P \vee Q$	de 1) y 8)
	15)	$P \vee Q$	de 1) y 9)
	16)	$P \vee Q$	de 1) y 10)
	17)	Q	de 1) y 11)
	18)	P	de 1) y 12)
	19)	Q	de 2) y 6)
	20)	$\neg P \vee Q$	de 2) y 7)
	21)	$\neg P \vee Q$	de 2) y 8)
	22)	$\neg P \vee Q$	de 2) y 9)
	23)	$\neg P \vee Q$	de 2) y 10)
	24)	$\neg P$	de 2) y 12)
	25)	P	de 3) y 5)
	26)	$P \vee \neg Q$	de 3) y 7)
	27)	$P \vee \neg Q$	de 3) y 8)
	28)	$P \vee \neg Q$	de 3) y 9)
	29)	$P \vee \neg Q$	de 3) y 10)
	30)	$\neg Q$	de 3) y 11)
	31)	$\neg P$	de 4) y 5)
	32)	$\neg Q$	de 4) y 6)
	33)	$\neg P \vee \neg Q$	de 4) y 7)
	34)	$\neg P \vee \neg Q$	de 4) y 8)
	35)	$\neg P \vee \neg Q$	de 4) y 9)
	36)	$\neg P \vee \neg Q$	de 4) y 10)
	37)	Q	de 5) y 7)
	38)	Q	de 5) y 9)
	39)	$\square$	de 5) y 12)

Se generan muchas cláusulas redundantes e irrelevantes:

- 7), 8), 9) y 10) son tautologías
- Su interacción con otras genera más cláusulas redundantes
- P, Q, $\neg$P y $\neg$Q se generan repetidas veces

En realidad bastaría con generar las cláusulas 5), 12) y 39)

Estrategias

- La aplicación de la regla de resolución en los sistemas de demostración automática de teoremas tiene variantes.

- La aplicación del procedimiento de saturación, sin limitaciones, genera normalmente muchas cláusulas irrelevantes y redundantes.

- Es necesario aplicar criterios selectivos de forma sistemática que simplifiquen el proceso y lo hagan computacionalmente eficiente.

- Se usarán dos tipos de criterios:
  - **Estrategias de simplificación**: con el objetivo de **reducir el número de cláusulas** en el conjunto.
  - **Estrategias de refinamiento**: con el objetivo de **limitar la generación de cláusulas.**

Estrategias de simplificación

1)Eliminación de cláusulas idénticas

Es posible deducir por resolución □ a partir de un conjunto de cláusulas C sii es posible deducir por resolución □ a partir de C tras eliminar cláusulas idénticas (obvio)

Conclusión: si se genera una cláusula que ya está en la demostración, no se incluye nuevamente

2)Eliminación de cláusulas con literales puros

Un literal L de un conjunto de cláusulas es puro si y sólo si no existe ningún otro literal complementario $\neg$L' en el conjunto tal que L y L' son unificables

Es posible deducir por resolución □ a partir de un conjunto de cláusulas C sii es posible deducir por resolución □ a partir de C tras eliminar las cláusulas con literales puros.

Una cláusula con un literal puro es inútil de cara a la refutación porque el literal nunca podrá eliminarse en el proceso de resolución.

Conclusión: esta estrategia sólo es necesario aplicarla una vez, puesto que de un conjunto de cláusulas sin literales puros no pueden generarse cláusulas con literales puros.

Estrategias de simplificación

3)Eliminación de cláusulas tautológicas

Es posible deducir por resolución □ a partir de un conjunto de cláusulas C sii es posible deducir por resolución □ a partir de C tras eliminar las cláusulas tautológicas

Una cláusula tautológica es verdad para cualquier interpretación, por lo que si la borramos de un conjunto de cláusulas insatisfacible, el conjunto seguirá siendo insatisfacible.

Nota: $p(x) \vee \neg p(x) \vee q(y)$ es una cláusula tautológica, pero $p(x) \vee \neg p(y) \vee q(y)$ no

1)	$P \vee Q$	
2)	$\neg P \vee Q$	
3)	$P \vee \neg Q$	
4)	$\neg P \vee \neg Q$	
5)	Q	de 1) y 2)
6)	P	de 1) y 3)
7)	$Q \vee \neg Q$	de 1) y 4)
8)	$P \vee \neg P$	de 1) y 4)
9)	$\neg P$	de 2) y 4)
10)	$\neg Q$	de 3) y 4)
11)	□	de 5) y 10)

Si se eliminan las tautologías, la refutación anterior queda:

1)	$P \vee Q$	
2)	$\neg P \vee Q$	
3)	$P \vee \neg Q$	
4)	$\neg P \vee \neg Q$	
5)	Q	de 1) y 2)
6)	P	de 1) y 3)
7)	$\neg P$	de 2) y 4)
8)	$\neg Q$	de 3) y 4)
9)	□	de 5) y 8)

Refutaciones y Estrategias de refinamiento

- **Derivación** de una cláusula C a partir de un conjunto de cláusulas $\{C_1,...,C_n\}$ es una secuencia $<C_1,...,C_n,R_1,...,R_m>$ tal que:
  - Cada R_i es el resolvente de dos cláusulas anteriores de la secuencia
  - No se realiza el mismo paso de resolución (entre las mismas cláusulas con los mismos factores) más de una vez.
  - $R_m = C$

- **Refutación:** derivación de □ a partir de $\{C_1,...,C_n\}$

- La regla de resolución con umg es correcta ➔ una derivación es una deducción correcta.

- El método de resolución es completo ➔ si {C1,...,Cn} es insatisfacible, hay una derivación de □ mediante resolución.

- El método general de demostración de insatisfacibilidad se puede poner en términos de la búsqueda de una refutación: se consideran todas las posibles derivaciones del conjunto de cláusulas hasta encontrar una que sea una refutación.

- Las estrategias de refinamiento sirven para reducir el espacio de búsqueda de refutaciones.

Resolución Lineal

- Descripción informal: Todos los pasos de resolución (salvo el primero) se dan utilizando el último resolvente generado en la deducción

- **Derivación lineal**: Una derivación lineal de C_m a partir de $\{C_1,...,C_n\}$ es una secuencia $C_1,...,C_n,C_{n+1},...,C_m$ tal que

  C_{n+1} es el resolvente de dos cláusulas $\in \{C_1,...,C_n\}$ (*cláusulas de cabecera*)

  y

  Para todo $i > n+1$, C_i es el resolvente de C_{i-1} con otra cláusula C_j, $j < i$

- **Resolución lineal**: Sólo genera derivaciones lineales

- **La resolución lineal es completa**: Un conjunto de cláusulas C es insatisfacible sii existe una refutación lineal de C
  - Podemos restringir las derivaciones posibles a derivaciones lineales

- Además, no es necesario probar con todas las cláusulas del conjunto inicial como punto de partida de la refutación lineal:

  - Se puede demostrar que si un conjunto de cláusulas S es satisfacible y $S \cup \{C\}$ (C es una cláusula) es insatisfacible, entonces hay un refutación lineal que empieza con C.

Resolución Input

- Descripción informal: Todos los pasos de resolución se dan utilizando al menos una cláusula del conjunto de partida

- **Derivación input**: Una derivación input de C_m a partir de $\{C_1,...,C_n\}$ es una secuencia $C_1,...,C_n,C_{n+1},...,C_m$ tal que :
  - Para todo $i > n$, C_i es resolvente de una cláusula $C_k \in \{C_1,...,C_n\}$ y otra cláusula C_j, $j < i$

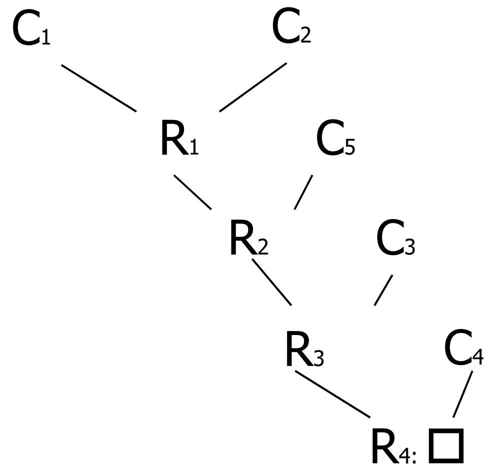
- **Resolución input**: Sólo genera derivaciones input

- Ejemplos:

  C_1: $\neg T(x) \vee L(x)$, C_2: $\neg D(x) \vee \neg L(x)$, C_3: $D(a)$, C_4: $I(a)$, C_5: $\neg I(x) \vee T(x)$

Refutación input (y lineal) partiendo de C_1:

R_1: $\neg T(x) \vee \neg D(x)$	(C_1, C_2)
R_2: $\neg I(x) \vee \neg D(x)$	(R_1, C_5)
R_3: $\neg I(a)$	(R_2, C_3)
R_4: $\square$	(R_3, C_4)



Refutación input (y lineal) partiendo de C_5:

R_1: $T(a)$	(C_5, C_4)
R_2: $L(a)$	(R_1, C_1)
R_3: $\neg D(a)$	(R_2, C_2)
R_4: $\square$	(R_3, C_3)

Resolución Input

- Ejemplo: C_1: p v q, C_2: ¬p v q, C_3: r v ¬q, C_4: ¬r v ¬q

  Refutación *no lineal y no input*:

R_1: q v q	(C_1, C_2)
R_2: ¬q v ¬q	(C_3, C_4)
R_3: □	(R_1, R_2)

  Pero para toda derivación no lineal hay una lineal equivalente:

R_1: q v q	(C_1, C_2)
R_2: r	(R_1, C_3)
R_3: ¬q	(R_2, C_4)
R_4: □	(R_3, R_1)

  - ¿Hay también una resolución input para toda resolución no input?

- **La resolución input no es completa (en el caso general):** Para cualquier conjunto de cláusulas C que sea insatisfacible no es posible afirmar que exista una refutación input de C
  - El ejemplo anterior es la prueba en contrario: no es posible deducir □ por resolución input.

Resolución Dirigida

- <u>Descripción informal</u>: no se pueden realizar resolventes de dos cláusulas de un subconjunto S del conjunto de partida (típicamente las premisas)

- **Derivación dirigida:** Una derivación dirigida de C_m a partir de $\{C_1,\ldots,C_n\}$, con conjunto soporte $S \subset C$, es una secuencia $C_1,\ldots,C_n,C_{n+1},\ldots,C_m$ tal que:
  - Para todo $i > n$, C_i es resolvente de dos cláusulas anteriores en la secuencia que no pertenecen ambas a S

- Las cláusulas de S son **cláusulas soporte** y las de C – S son **cláusulas objetivo**

- **Resolución dirigida:** Sólo genera derivaciones dirigidas

- Ejemplo: $C = \{C_1$: s v t, C_2: ¬s v p, C_3: ¬q v r, C_4: q v ¬p, C_5: u v ¬r, C_6: ¬u, C_7: ¬t$\}$, $S = \{C_1, C_2, C_3, C_4, C_5\}$

1.	s	(C_1, $\mathbf{C_7}$)
2.	p	(R_1, C_2)
3.	q	(R_2, C_4)
4.	r	(R_3, C_3)
5.	u	(R_4, C_5)
6.	☐	(R_5, $\mathbf{C_6}$)

refutación dirigida

1.	t v p	(C_1, C_2)
2.	p	(R_1, $\mathbf{C_7}$)
3.	q	(R_2, C_4)
4.	r	(R_3, C_3)
5.	u	(R_4, C_5)
6.	☐	(R_5, $\mathbf{C_6}$)

refutación no dirigida

Resolución Dirigida

- **La resolución dirigida puede ser completa:** Si hay una derivación de ☐ a partir de C, y S ($S \subset C$) es satisfacible, hay una resolución dirigida de ☐ con conjunto soporte S.

- Esta estrategia no es de mucha utilidad si no es posible identificar un conjunto soporte satisfacible

- En la práctica, en la búsqueda de la refutación de una conclusión a partir de un conjunto de premisas, es una *heurística* razonable operar bajo el supuesto de la satisfacibilidad de las premisas:
  - Premisas de la demostración (en forma clausular): conjunto soporte **S**
  - Negación de la conclusión (en forma clausular): conjunto objetivo **C – S**
  - Si las premisas son inconsistentes, entonces cualquier conclusión se deriva: $[A, \neg A] \vdash B$
  - Pero si las premisas son consistentes, entonces ☐ debe derivarse de la negación de la conclusión: $[A, B] \vdash B$, es decir $\{A, B, \neg B\}$ es insatisfacible.

Resolución Ordenada

- <u>Descripción informal</u>: Todos los pasos de resolución tienen en cuenta dos restricciones de orden: 1) sobre la posición en las cláusulas de los literales sobre los que se resuelve y 2) sobre la posición en el resolvente de los literales que lo forman.

- Existen numerosas formas de tener en cuenta el orden de los literales en las clausulas para limitar las demostraciones tomadas en consideración dentro del árbol de las deducciones.

- En la resolución ordenada las cláusulas son sucesiones finitas de literales (con literales ordenados).

- Para poder llevar a cabo la resolución entre dos clausulas se tiene que poder aplicar la regla de resolución, pero además:

    - La resolución tiene que ser practicada con los literales que están en cabeza de las clausulas.

    - El resolvente tiene que estar ordenado.

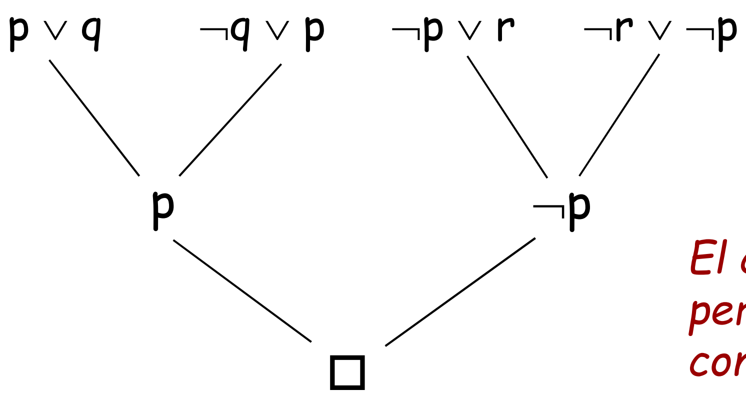
Resolución Ordenada

- **Derivación ordenada:** Una derivación ordenada de C_m a partir de $\{C_1,\ldots,C_n\}$ es una secuencia $C_1,\ldots,C_n,C_{n+1},\ldots,C_m$ tal que:
  - o cada C_i, $i>n$, es resolvente de dos cláusulas anteriores $A_1 \vee L_{11} \vee \ldots \vee L_{1p}$ y $\neg A_2 \vee L_{21} \vee \ldots \vee L_{2q}$ donde A_1 y A_2 son unificables con un umg σ,
  - o Los literales de Ci están ordenados de esta forma: $(L_{11} \vee \ldots \vee L_{1p} \vee L_{21} \vee \ldots \vee L_{2q})\sigma$

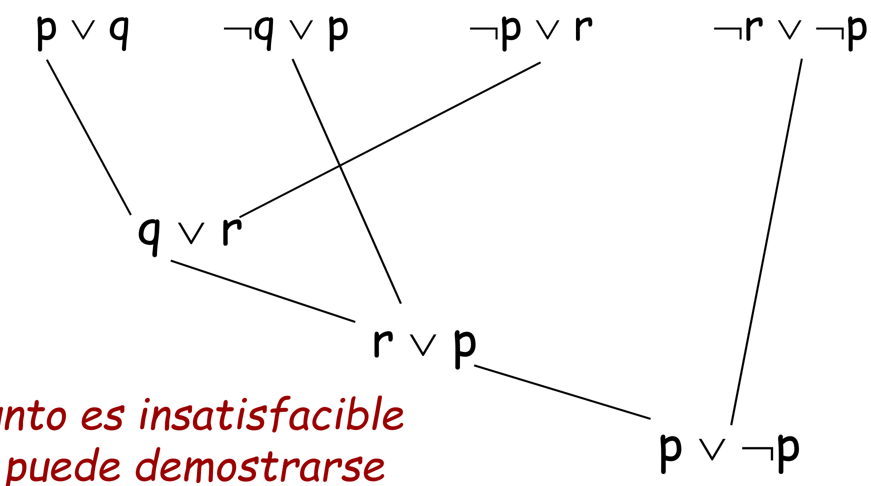
- **La resolución ordenada no es completa**

$\{p \vee q, \neg q \vee p, \neg p \vee r, \neg r \vee \neg p\}$

NO ORDENADA



ORDENADA



El conjunto es insatisfacible pero no puede demostrarse con una refutación ordenada

Estrategias de Refinamiento. Propiedades

- **Corrección:** □ se deduce sólo si el conjunto de cláusulas S es insatisfacible (□ → S insatisfacible)
- **Completitud:** Si el conjunto de cláusulas S es insatisfacible entonces □ se puede deducir (S insatisfacible → □)

	Correcta	Completa
Lineal	Si	Si
Input	Si	No, en el caso general
Dirigida	Si	Sí, si el conjunto soporte es satisfacible
Ordenada	Si	No

❑ Ejercicios:

Ejercicio 1: Demostrar, mediante resolución, que la siguiente deducción es correcta:

A_1 : $\forall x(C(x) \rightarrow (W(x) \wedge R(x)))$
A_2 : $\exists x(C(x) \wedge Q(x))$

C : $\exists x(Q(x) \wedge R(x))$

Para demostrarlo aplicamos el método de resolución al siguiente conjunto :

$A_1 \wedge A_2 \wedge \neg C$

$A_1 = \{\{\neg C(x) \vee W(x)\}, \{\neg C(y) \vee R(y)\}\}$

$A_2 = \{\{C(a)\}, \{Q(a)\}\}$

$\neg C = \{\neg Q(z) \vee \neg R(z)\}$

$C_1 : \neg C(x) \vee W(x)$
$C_2 : \neg C(y) \vee R(y)$
$C_3 : \neg Q(z) \vee \neg R(z)$
$C_4 : C(a)$
$C_5 : Q(a)$

Y comprobamos su satisfacibilidad mediante el método de resolución por UMG.

1. ¬C(x) ∨ W(x)
2. ¬C(y) ∨ R(y)
3. ¬Q(z) ∨ ¬R(z)
4. C(a)
5. Q(a)

C_2: ¬C(y) ∨ R(y) C_4 : C(a)

{y/a}

R_1: R(a) C_3 : ¬Q(z) ∨ ¬R(z)

{z/a}

R_2: ¬Q(a) C_5 : Q(a)

R_3: □

Hemos encontrado la clausula vacía, luego el conjunto de clausulas es insatisfacible, luego la deducción es correcta.

Ejercicio 2: Estudiar si el siguiente conjunto es insatisfacible utilizando el método de resolución:

C1 : $r(x) \vee p(x) \vee \neg q(h(x))$
C2 : $\neg r(x)$
C3 : $p(y) \vee \neg s(y, h(y)) \vee r(y)$
C4 : $\neg s(z, x)$
C5 : $q(y) \vee r(y)$
C6 : $s(f(x), x) \vee \neg p(f(x))$

Comprobamos si $C1 \wedge C2 \wedge C3 \wedge C4 \wedge C5 \wedge C6$ es insatisfacible.

Aplicamos el método de resolución con unificación umg para comprobar la insatisfacibilidad.

*R1 y R2 ---- **NO se puede UNIFICAR con x1/h(x1)}, daría h(h(x1), y no se pueden cambiar variables, ver regla resolución.***

El problema está en querer usar C2 en ambas resoluciones, se puede crear C2' cambiando la variable y usarla con C5

Clausulas renombrando variables:

C1 : r(x) ∨ p(x) ∨ ¬q(h(x))

C2 : ¬r(x1)

C3 : p(y) ∨ ¬s(y, h(y)) ∨ r(y)

C4 : ¬s(z, x2)

C5 : q(y1) ∨ r(y1)

C6 : s(f(x3), x3) ∨ ¬p(f(x3))

C_1: r(x) ∨ p(x) ∨ ¬q(h(x)) C_2: ¬r(x1) C_5: q(y) ∨ r(y)

{x/x1} *{y/x1}*

R_1: p(x1) ∨ ¬q(h(x1)) R_2: q(x1)

R_3: p(x1) C_6: s(f(x3), x3) ∨ ¬p(f(x3))

C_4: ¬s(z, x2)

{x1/f(x3)}

R_4: s(f(x3), x3)

{x2/x3 z/f(x3)}

R_5: □

25

C1 : r(x) ∨ p(x) ∨ ¬q(h(x))
C2 : ¬r(x1)
C3 : p(y) ∨ ¬s(y, h(y)) ∨ r(y)
C4 : ¬s(z, x2)
C5 : q(y1) ∨ r(y1)
C6 : s(f(x3), x3) ∨ ¬p(f(x3))

Puedo hacer un segundo renombrado en C2 para dar C2’ con otra variable x, que me permita hacer unificación entre R1 y R2.

En general, aunque puedo usar los resolventes y clausulas de partida varias veces, **es necesario renombrar sus variables para que no haya dos sustituciones distintas de la misma variable.**

C_1: r(x) ∨ p(x) ∨ ¬q(h(x)) C_2: ¬r(x1) C_5: q(y) ∨ r(y) $C_{2'}$: ¬r(x4)

{x/x1}   *{y/x4}*

R_1: p(x1) ∨ ¬q(h(x1)) R_2: q(x4)

{x4/h(x1)}

R_3: p(x1) C_6: s(f(x3), x3) ∨ ¬p(f(x3))

C_4: ¬s(z, x2)

{x1/f(x3)}

R_4: s(f(x3), x3)

{x2/x3
z/f(x3)}

R_5: □

Examen Julio 2015

Ejercicio 4. Demuestra por resolución UMG (comenzando con la cláusula C7), que el siguiente conjunto de cláusulas es insatisfacible: *(2,5 puntos)*

C1: ¬P(x) ∨ ¬Q(y,z,w) ∨ ¬R(x,w) ∨ R(x,y)

C2: Q(a,f(b),f(c))

C3: Q(x,x,f(x))

C4: ¬Q(x,y,z) ∨ R(x,z)

C5: P(a)

C6: ¬R(a,c) ∨ ¬S(f(x))

C7: S(f(x)) ∨ ¬P(x)

SOLUCIÓN:

R1: S(f(a))	C7, C5 {x7/a}
R2: ¬R(a, c)	R1, C6 {x6/a}
R3: ¬P(a) V ¬Q(c, z1, w1) V ¬R(a, w1)	R2, C1 {x1/a, y1/c}
R4: ¬Q(c, z1, w1) V ¬R(a, w1)	R3, C5 {}
R5: ¬R(a, f(c))	R4, C3 {x3/c, z1/c, w1/f(c)}
R6: ¬Q(a, y4, f(c))	R5, C4 {x4/a, z4/f(c)}
R7: □	R6, C2 {y4/f(b)}

Más consejos:

- Al pasar a forma clausular, no es necesario incluir formulas para cada paso como en deducción natural, pero sí se recomienda: etiquetar las sustituciones, y seguir los pasos en el orden que se han dado en clase: prenex (renombrado,…), cierre, FNC,FNS.
- Recuerda que, en el paso de FNS, se deben usar símbolos de funciones y constantes nuevos. Así que si en una formula (A1) ya has usado "f" o "a", en otra formula A2 no podrás usarlos (sino que serán "g" o "b" asumiendo que no hayan aparecido ya).
- En resolución, no olvides negar la conclusión si la hay.
- Renombra todas las variables con el índice de la clausula: x7 es la variable x en la clausula 7.
- Aunque puedo usar los resolventes y clausulas de partida varias veces, es necesario renombrar sus variables para que no haya dos sustituciones distintas de la misma variable (dando C2′ con x2′).
- Puede ser necesario factorizar o aplicar idempotencia, aunque no es muy típico.
- Una solución bien indexada y con un resolvente por línea (como en este ejercicio), puede ser más cómoda que un árbol de resolución. En cualquier caso, **hay que etiquetar las clausulas y resolventes que se van usando** (C1, C2,C2′, R1…) para facilitar la corrección.

Anexo, Programación lógica

Programación lógica

- Las técnicas de demostración automática de teoremas, como la resolución, pueden aplicarse para diseñar sistemas de extracción de respuestas, sistemas expertos, y resolución de problemas (Green, Raphael, 1968).

- Un programa lógico en Prolog (1972) se compone exclusivamente de cláusulas de Horn.
  - Clausula con máximo un literal sin negar (que hace de consecuente).
- Una cláusula de Horn que no tiene literales negados se denomina *hecho.*
- Una cláusula de Horn que tiene un literal afirmado (y el resto negados) se denomina *regla*. El literal afirmado se llama *cabeza* y la secuencia de literales negados se llama *cuerpo.*
  - *Sentencia Prolog: "A :- B,C,D."*
    - *Equivale a* $(A \vee \neg B \vee \neg C \vee \neg D)$ *que es* $(B \wedge C \wedge D \rightarrow A)$

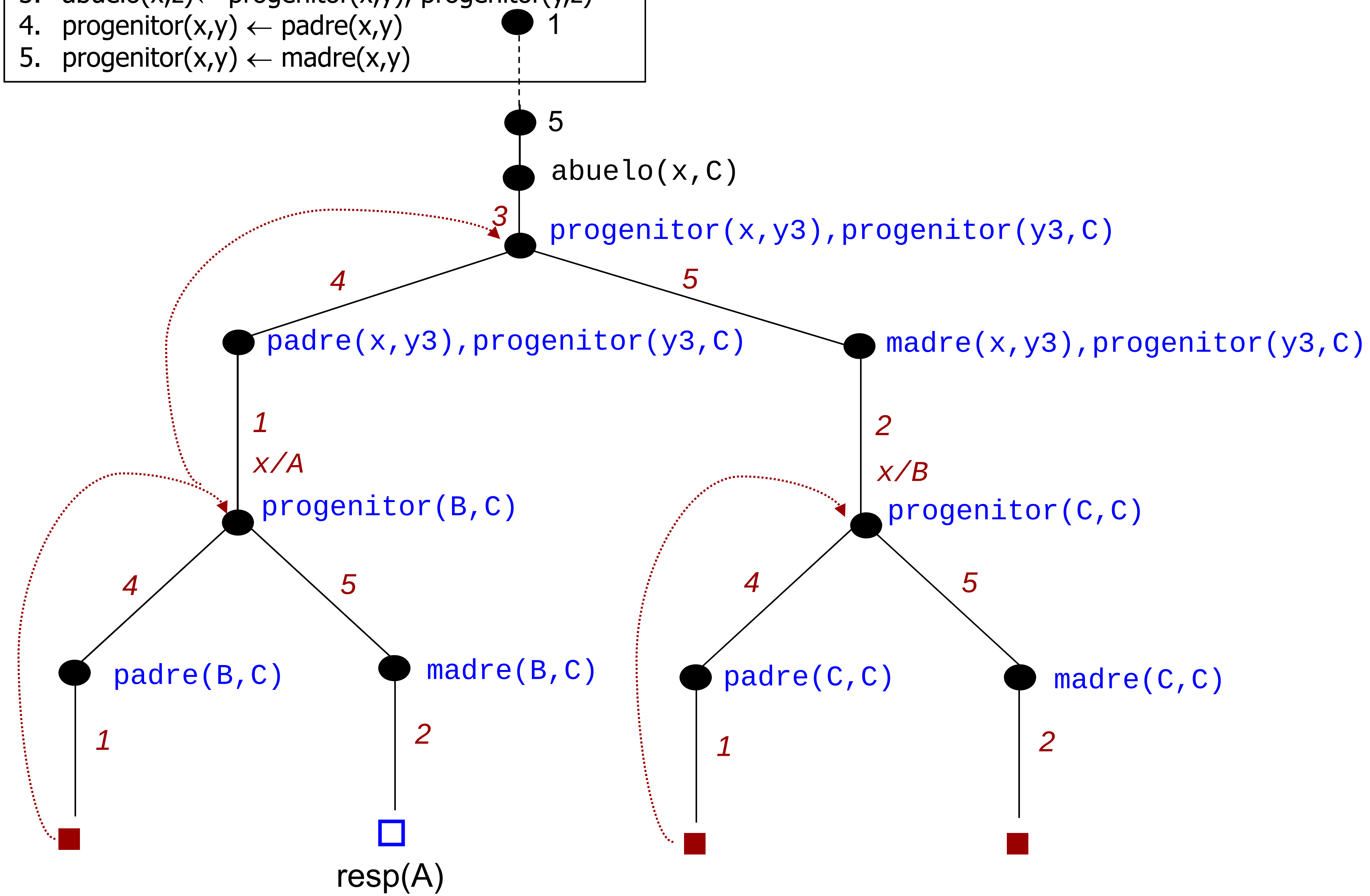
- Un conjunto de reglas con la misma cabeza es un *procedimiento*, cuyo nombre es el del símbolo de predicado correspondiente a la cabeza.
- Un conjunto de definiciones de procedimientos es un *programa* lógico.

Programación lógica: Consultas

- Una cláusula de Horn que no tiene literal afirmado se denomina *objetivo.*
- Un objetivo se escribe como si fuera un cuerpo y se considera una pregunta
- La deducción cuya corrección se debe comprobar es la que tiene como premisas al programa y como conclusión al objetivo.
- La *ejecución* de un programa lógico con un objetivo dado consiste en determinar si el objetivo es deducible del programa, y caso de que lo sea, los valores de las variables del objetivo que dan una respuesta al mismo
- Para la ejecución se aplica resolución *SLD* (Selective Linear Definite clause resolution) con el objetivo como cláusula objetivo inicial.
    - Resolución SLD: Estrategia que combina las estrategias lineal, input, dirigida y ordenada sobre clausulas de Horn.
- La búsqueda de la refutación se hace en profundidad con vuelta atrás (*backtraking*).

Ejemplo

1. padre(A,B)
2. madre(B,C)
3. abuelo(x,z)← progenitor(x,y), progenitor(y,z)
4. progenitor(x,y) ← padre(x,y)
5. progenitor(x,y) ← madre(x,y)



Ejemplo

1. padre(A,B)
2. madre(B,C)
3. abuelo(x,z)← progenitor(x,y), progenitor(y,z)
4. progenitor(x,y) ← padre(x,y)
5. progenitor(x,y) ← madre(x,y)

¿Quién es abuelo de C?: abuelo(x,C)

- abuelo(x,C),resp(x)
  [3,x3/x,z3/C]
- progenitor(x,y3),progenitor(y3,C),resp(x)
  [4,x4/x,y4/y3]
- padre(x,y3),progenitor(y3,C),resp(x)
  [1,x/A,y3/B]
- progenitor(B,C),resp(A)
  [4,x4'/B,y4'/C]
- padre(B,C),resp(A)
  <fallo>
  [5,x5/B,y5/C]
- madre(B,C),resp(A)
  [2]
- **resp(A)**

¿Quién es nieto de A?: abuelo(A,x)

- abuelo(A,x),resp(x)
  [3,x3/A,z3/x]
- progenitor(A,y3),progenitor(y3,x),resp(x)
  [4,x4/A,y4/y3]
- padre(A,y3),progenitor(y3,x),resp(x)
  [1,y3/B]
- progenitor(B,x),resp(x)
  [4,x4'/B,y4'/x]
- padre(B,x),resp(x)
  <fallo>
  [5,x5/B,y5/x]
- madre(B,x),resp(x)
  [2,x/C]
- **resp(C)**

¡Prueba Prolog!

- Puedes programar en Prolog y encontrar ejemplos online: http://swish.swi-prolog.org/example/examples.swinb

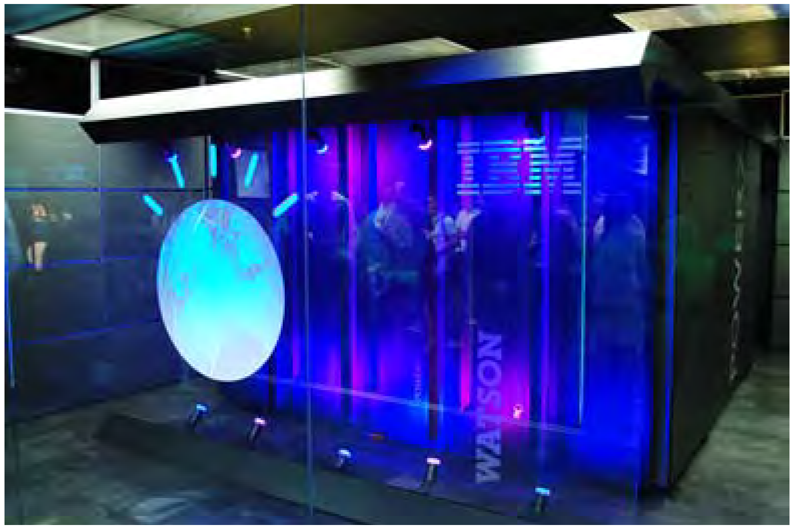
  - Movie database provides a couple of thousands of facts about movies for you to query.
  - Expert system illustrates simple meta-interpretation of rules and asking for missing knowledge.
  - Eliza implements the classical shrink.

Watson usa Prolog

- Watson, sistema de inteligencia artificial que es capaz de responder a preguntas formuladas en lenguaje natural de IBM.

- En 2011, Watson compitió en Jeopardy! derrotando a sus dos oponentes humanos .
  - (Brad Rutter, el mayor ganador de dinero; y Ken Jennings, récord en racha de victorias)
  - Watson tuvo acceso a cuatro terabytes de almacenamiento en disco sin conexión a internet, incluyendo el texto completo de la Wikipedia en inglés.
  - ¡4 TB!, la eficiencia importa.

- La UPM e IBM han llegado a un acuerdo de colaboración para llevar Watson a las aulas, por primera vez en España (noticia).



Departamento de Inteligencia Artificial
Escuela Técnica Superior de
Ingenieros Informáticos

Lógica
Resumen bloque
lógica de primer orden

Profesor: Emilio Serrano
emilioserra@fi.upm.es

Estructuras básicas

- Todos los hombres son mortales
  - $\forall x(H(x) \rightarrow M(x))$ o $\neg\exists x(H(x) \wedge \neg M(x))$
- No todo hombre es mortal
  - $\neg \forall x(H(x) \rightarrow M(x))$ o $\exists x(H(x) \wedge \neg M(x))$
- Al menos un hombre es mortal, hay hombres mortales
  - $\exists x(H(x) \wedge M(x))$
- Ningún hombre es mortal
  - $\neg \exists x(H(x) \wedge M(x))$ o $\forall x(H(x) \rightarrow \neg M(x))$
- Sólo los hombres son mortales
  - $\forall x(M(x) \rightarrow H(x))$ o $\neg\exists x(M(x) \wedge \neg H(x))$
- (Típicamente, $\forall$ se usa con implicación y $\exists$ con conjunción).

Interpretaciones

- **2.- Interpretación de fórmulas moleculares:**
  - Asignación de valor de verdad a las fórmulas moleculares:

1. ***i(¬A) = V*** *sii i(A) = F;* ***i(¬A) = F*** *sii i(A) = V*
2. ***i(A ∧B) = V*** *sii i(A) = V* y *i(B) = V;* ***i(A ∧B) = F*** *siii(A) = F* o *i(B) = F*
3. ***i(A ∨B) = V*** *sii i(A) = V* o *i(B) = V;* ***i(A ∨B) = F*** *siii(A) = F* y *i(B) = F*
4. ***i(A →B) = V*** *sii i(A) = F* o *i(B) = V;* ***i(A →B) = F*** *siii(A) = V* y *i(B) = F*
5. ***i(A ↔B) = V*** *sii i(A) = i(B);* ***i(A ↔B) = F*** *sii i(A) ≠ i(B);*

6. ***i(∃xA) = V*** *sii i(A{x/a}) = V* para al menos una sustitución *x/a, **a** ∈L(D)*
7. ***i(∃xA) = F*** *sii i(A{x/a}) = F* para toda sustitución *x/a*, ***a*** constante de *L(D)*
8. ***i(∀xA) = V*** *sii i(A{x/a}) = V* para toda sustitución *x/a*, ***a*** constante de *L(D)*
9. ***i(∀xA) = F*** *sii i(A{x/a}) = F* para al menos una sustitución *x/a*, ***a*** *∈L(D)*

(En los ejercicios aparece “y además” en todas las sustituciones para i(∀xA) = V y i(∃ xA) = F;
y “o bien” en las sustituciones para i(∃xA) = V y para i(∀xA) = F)

Ejemplo de $\vDash$

¿ $\vDash \exists x \forall y P(x,y) \rightarrow \forall y \exists x P(x,y)$?

D = {1, 2} i(a)=1, i(b)=2

i($\exists x \forall y P(x,y) \rightarrow \forall y \exists x P(x,y)$) **= F** sii

i($\exists x \forall y P(x,y)$) = V sii

{x/a} i($\forall y P(a,y)$) = V sii {y/a} i(P(a,a)) = V y {y/b} i(P(a,b)) = V

o bien

{x/b} i($\forall y P(b,y)$) = V sii {y/a} i(P(b,a)) = V y {y/b} i(P(b,b)) = V

i($\forall y \exists x P(x,y)$) = F sii

o bien {y/a} i($\exists x P(x,a)$) = F sii {x/a} i(P(a,a)) = F y {x/b} i(P(b,a)) = F

{y/b} i($\exists x P(x,b)$) = F sii {x/a} i(P(a,b)) = F y {x/b} i(P(b,b)) = F

La imposibilidad de hacer falsa esta fórmula no es exclusiva de esta interpretación. Verificar el antecedente es incompatible con hacer falso el consecuente ⇨ **La fórmula es válida**

Ejemplo de $\not\models$

¿ $\models \forall y \exists x P(x,y) \rightarrow \exists x \forall y P(x,y)$?

D = {1, 2}   i(a)=1, i(b)=2

$i(\forall y \exists x P(x,y) \rightarrow \exists x \forall y P(x,y)\ )$ **= F** sii

$i(\forall y \exists x P(x,y)) = V$ sii

{y/a} $i(\exists x P(x,a)) = V$ sii   {x/a}   {x/b}

$i(P(a,a)) = V$ o bien $i(P(b,a)) = V$

Y además

{y/b} $i(\exists x P(x,b)) = V$ sii   {x/a}   {x/b}

$i(P(a,b)) = V$ o bien **$i(P(b,b)) = V$**

$i(\exists x \forall y P(x,y)) = F$ sii

{x/a} $i(\forall y P(a,y)) = F$ sii   {y/a}   {y/b}

$i(P(a,a)) = F$ o bien **$i(P(a,b)) = F$**

Y además

{x/b} $i(\forall y P(b,y)) = F$ sii   {y/a}   {y/b}

$i(P(b,a)) = F$ o bien $i(P(b,b)) = F$

Sí es posible definir una interpretación que hace falsa la fórmula ⇨ **La fórmula no es válida, el contramodelo es: $P_D(1,1) = V$, $P_D(1,2) = F$ $P_D(2,1) = F$, $P_D(2,2) = V$**

Ejercicio de examen Julio 2015

Ejercicio 2. Sea L el lenguaje { R , a , b , c }, donde R(x,y) es un símbolo de predicado y a, b, c son constantes. Definir un contramodelo en un dominio de 3 elementos para demostrar lo siguiente:

(2,5 puntos)

$$\forall x \exists y\, R(x,f(y)) \not\models \exists y \forall x\, R(x,f(y))$$

Justificar adecuadamente la respuesta con un análisis semántico.

En la lógica el orden de los cuantificadores es importante. Por ejemplo, que todo el mundo ame a alguien no implica que alguien sea amado por todo el mundo.

Para construir un contramodelo i hay que definir un dominio D, y dar los valores de la función f y el predicado R.

Una solución sencillo sería: dominio de i, D= {1,2,3}.

Interpretación i(a)=1, i(b)= 2, i(c)= 3.

Sea f la función de identidad: f(x) = x; entonces tenemos f(1) =1, f(2)=2, f(3)=3.

Se define R tal que R(x,y) es verdadero en el modelo si y solo si x=y. Entonces los valores de R son <1,1>, <2,2> y <3,3>. Por eso, R(a,a), R(b,b) y R(c,c) son verdaderos en i y otros átomes con R son falsos. Poer ejemplo. piensa en un dominio de egoistas donde todo el mundo se ama a sí mismo y a nadie más.

Ahora es evidente que en el modelo i la formula ∀x ∃y R(x,f(y)) es verdadera. Eso porque i(R(x,f(y)){x/a, y/a} = V, i(R(x,f(y)){x/b, y/b} = V y i(R(x,f(y)){x/c, y/c} = V: para cada sustitución por x se puede elegir una sustitución por y (=f(y)) tal que la fórmula sale verdadero.

Al contrario, la fórmula ∃y ∀x R(x,f(y)) es falsa en i. Con la substitución y/a, la fórmula R(b,a) es falsa; con la substitución y/b, la fórmula R(c,b) es falsa, y con la substitución y/c, la fórmula R(b,c) es falsa. Dado que no hay una substitución por y tal que R(x,f(y)) es verdadero para cada substitución por x, tenemos i(∃y ∀x R(x,f(y))) = F

Tenemos por tanto un contramodelo que demuestra la proposición.

El uso de tablas bidimensionales para la interpretación de predicados de dos argumentos puede ser útil también. Ya sea para comprobar una hipótesis o para desarrollarla.

R_D	1	2	3
1	V	F	F
2	F	V	F
3	F	F	V

La regla para un examen es siempre: tiene que haber (1) un desarrollo claro y justificado de como alcanzar la conclusión, y (2) una conclusión clara y explícita.

En semántica (LP y LPO) se admite el procedimiento inverso: (1) definir un contramodelo al azar (o guiado, o con tablas de verdad en sucio) para demostrar ⊭, e (2) interpretar las formulas EXPLÍCITAMENTE Y PASO A PASO para comprobar que efectivamente es un contramodelo. (Aunque un error por este procedimiento puede penalizar más).

Estrategia general problemas DN

- Identifica la conectiva principal y el tipo de formula de la conclusión y opera según lo siguiente:
  - Si es $A \rightarrow B$, supón A y usa $\mathbf{I}_{\rightarrow}$ cuando encuentres B.
  - Si es $A \wedge B$, prueba A, prueba B, y usa $\mathbf{I}_{\wedge}$.
  - Si es $A \vee B$, prueba A o B (el más fácil), y usa $\mathbf{I}_{\vee}$.
    - La última estrategia, suponer $\neg(A \vee B)$ puede ser útil.
  - Si es $A \leftrightarrow B$, prueba $A \rightarrow B$, $B \rightarrow A$, y aplica $\mathbf{I}_{\leftrightarrow}$
  - Si es $\neg A$, supón A, encuentra una contradicción, y usa $\mathbf{I}_{\neg}$.
  - Si es A, intenta encontrar una solución trivial con los elementos previos de la deducción.
  - Si todo falla, supón la conclusión negada en bloque, encuentra una contradicción, y usa $\mathbf{I}_{\neg}$.
  - Si la demostración no tiene premisas, es necesario empezar con un supuesto, ya sea de A para demostrar $\vdash A \rightarrow B$ o bien de $\neg A$ para demostrar $\vdash A$

Resumen DN

$I_\wedge$

$$\frac{A \quad B}{A \wedge B}$$

$E_\wedge$

$$\frac{A \wedge B}{B} \qquad \frac{A \wedge B}{A}$$

$I_\vee$

$$\frac{A}{B \vee A} \qquad \frac{A}{A \vee B}$$

$E_\vee$

$$\frac{A \vee B \quad A \to C \quad B \to C}{C}$$

$I_\to$

$$\frac{\begin{array}{c} A \text{ (supuesto)} \\ B \end{array}}{A \to B}$$

$E_\to$

$$\frac{A \to B \quad A}{B}$$

$I_\leftrightarrow$

$$\frac{A \to B \quad B \to A}{A \leftrightarrow B}$$

$E_\leftrightarrow$

$$\frac{A \leftrightarrow B}{A \to B} \qquad \frac{A \leftrightarrow B}{B \to A}$$

$I_\neg$

$$\frac{\begin{array}{c} A \text{ (supuesto)} \\ B \wedge \neg B \end{array}}{\neg A} \qquad \frac{A \to B \wedge \neg B}{\neg A}$$

$E_\neg$

$$\frac{\neg\neg A}{A}$$

$I_\exists$

$$\frac{A(t)}{\exists x A(x)}$$

$E_\exists$

$$\frac{\exists x A}{A\{x/a^*\}}$$

$I_\forall$

$$\frac{A(x)}{\forall x A}$$

$E_\forall$

$$\frac{\forall x A}{A\{x/t\}}$$

Derivadas

- $T[A \wedge \neg A] \vdash B$ Ex Contradictione Quodlibet
- $T[A \to B, B \to C] \vdash A \to C$ Transitividad
- $T[A \to B, \neg B] \vdash \neg A$ Modus Tollens
- $T[A \vee B, \neg A] \vdash B$ Corte
- $T[A \vee B, \neg B] \vdash A$ Corte
- $T[A \vee B, \neg A \vee C] \vdash B \vee C$ Corte
- + regla de iteración
- + teorema del intercambio (y equivalencias)

CONDICIONES:

- *siendo a* un nombre temporal nuevo en la demostración*

CONDICIONES:

- *A(x) no incluye nombres temporales (a*)*
- *x no aparece libre en premisa, libre fuera de la subprueba en la que se aplica la regla, ni en ningún supuesto previo no cancelado*

Equivalencias lógicas LPO

Renombrado (si *y* no aparece libre en A)

- $\forall x A(x) \Leftrightarrow \forall y A(x/y)$
- $\exists x A(x) \Leftrightarrow \exists y A(x/y)$

Cuantificadores vs negación

- $\neg\forall x A(x) \Leftrightarrow \exists x \neg A(x)$
- $\neg\exists x A(x) \Leftrightarrow \forall x \neg A(x)$

Conectivas vs cuantificadores (*x* no aparece libre en C)

- $\forall x A \wedge C \Leftrightarrow \forall x(A \wedge C)$
- $\exists x A \wedge C \Leftrightarrow \exists x(A \wedge C)$
- $\forall x A \vee C \Leftrightarrow \forall x(A \vee C)$
- $\exists x A \vee C \Leftrightarrow \exists x(A \vee C)$
- $(\forall x A \rightarrow C) \Leftrightarrow \exists x(A \rightarrow C)$
- $(\exists x A \rightarrow C) \Leftrightarrow \forall x(A \rightarrow C)$
- $(A \rightarrow \forall x C) \Leftrightarrow \forall x(A \rightarrow C)$
- $(A \rightarrow \exists x C) \Leftrightarrow \exists x(A \rightarrow C)$
- $\forall x A \wedge \forall x C \Leftrightarrow \forall x(A \wedge C)$
- $(\exists x A \vee \exists x C) \Leftrightarrow \exists x(A \vee C)$

Ejemplo de uso en DN: “Intercambio 3, $\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$”.

No es válido: “equivalencia”, “de Morgan”, “$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$)”, no incluir el número donde se aplica, no hacer mención al teorema del intercambio, etcétera

+las vistas en LP

- $\neg\neg A \Leftrightarrow A$
- $A \wedge A \Leftrightarrow A$
- $A \vee A \Leftrightarrow A$
- $A \wedge B \Leftrightarrow B \wedge A$
- $A \vee B \Leftrightarrow B \vee A$
- $A \leftrightarrow B \Leftrightarrow B \leftrightarrow A$
- $A \wedge (B \wedge C) \Leftrightarrow (A \wedge B) \wedge C$
- $A \vee (B \vee C) \Leftrightarrow (A \vee B) \vee C$
- $A \rightarrow B \Leftrightarrow \neg A \vee B$
- $A \rightarrow B \Leftrightarrow \neg B \rightarrow \neg A$
- $\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B$
- $\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$
- $A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$
- $A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$
- $A \leftrightarrow B \Leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A)$

Examen DN Julio de 2015

Ejercicio 3. Demostrar la fórmula $\exists x\,(q(x) \rightarrow p(f(a),x))$ a partir del conjunto de premisas

$\{\ \forall x\,(\neg\, q(x) \vee r(x))\ ,\ \ \forall x\, \exists z\,(\neg\, p(x,z) \rightarrow \neg\, r(z))\ \}$

utilizando el cálculo de **Deducción Natural**. *(2,5 puntos)*

1.	$\forall x \exists z(\neg p(x,z) \rightarrow \neg r(z))$	premisa
2.	$\exists z(\neg p(f(a),z) \rightarrow \neg r(z))$	$\forall$-elim (1)
3.	$\neg p(f(a),c) \rightarrow \neg r(c)$	$\exists$-elim(2)
4.	$q(c)$	supuesto
5.	$\forall x(\neg q(x) \vee r(x))$	premisa
6.	$\neg q(c) \vee r(c)$	$\exists$-elim(5)
7.	$\neg\neg q(c)$	intercambio(4)
8.	$r(c)$	corte(6,7)
9.	$\neg\neg r(c)$	intercambio(8)
10.	$\neg\neg p(f(a),c)$	MT(3,9)
11.	$p(f(a),c)$	$\neg$-elim(10)
12.	$q(c) \rightarrow p(f(a),c)$	$\rightarrow$-intro(4-11)
13.	$\exists x(q(x) \rightarrow p(f(a),x))$	$\exists$-intro(12)

Los supuestos son siempre estratégicos (no se hacen sin saber el objetivo) y deben cerrarse cuando se encuentra ese objetivo.

Aquí uno sabe por la conclusión que se hará supuesto de q(una constante "uc"), buscará p(f(a),uc) dentro del supuesto, y así podrá cerrar el supuesto con $q(uc) \rightarrow p(f(a), uc)$, y después introducir el existencial.

La segunda premisa tiene un existencial, interesa eliminarlo antes de un supuesto para poder suponer la constante temporal añadida. (Puede que uno se de cuenta al avanzar en la demostración)

Además, uno puede ver en esa premisa, que interesará sustituir la x por f(a), ya que se busca p(f(a),"una constante". Ver estrategia.

¡Las reglas se aplican a rajatabla!, 7 es necesario para el corte en 8, 9 es necesario para el MT en 10. Hacer una simplificación no explícita penalizaría en el ejercicio.

También es importante anotar las sustituciones realizadas. En 2 se añadiría {x/f(a)}, en 5 se añadiría {z/c}

IMPORTANTE: Siempre que uses una equivalencia, como en puntos 7 y 9, pon:

Intercambio, nº donde se aplica, formula con metavariables. Omitir algo de esto penalizaría. Ejemplo: **Intercambio 3, $\neg\neg A \Leftrightarrow A$**

¡Deja claro cuando empieza y termina un supuesto!, tabulaciones, llaves, etcétera.

Forma Normal de Skolem

- **Procedimiento para hallar FNS(A)**:

1. Dada una fórmula *A*, poner la fórmula en forma Prenex.

   Prenex (A)

2. Realizar el cierre existencial.

   $Cierre_{\exists}(Prenex(A)) = QM$, donde *Q* y *M* son el prefijo y la matriz cuantificacionales de la fórmula, respectivamente.

3. Poner la fórmula en forma normal conjuntiva.

   Obtención de *FNC(M)*

4. Eliminar los cuantificadores existenciales.

   Obtención de *Skolem(QFNC(M)) = FNS(A)*

Forma prenexa

1. **Forma normal prenexa.** Aplicando a una fórmula los siguientes pasos se obtiene otra fórmula equivalente y que está en forma normal prenexa:

1) Rectificar la fórmula usando las equivalencias. Cambio de nombre de variables ligadas:

$$\vdash \forall x A(x) \leftrightarrow \forall y A(x/y)$$
$$\vdash \exists x A(x) \leftrightarrow \exists y A(x/y)$$

donde *y* es una variable que no ocurre libre en *A*.

(no deben haber varios cuantificadores distintos con la misma variable)

2) Interdefinición de cuantificadores (interiorizar negaciones usando equivalencias).

$$\vdash \neg\forall x A(x) \leftrightarrow \exists x \neg A(x)$$
$$\vdash \neg\exists x A(x) \leftrightarrow \forall x \neg A(x)$$

(no puede haber cuantificadores con una negación)

3) Distribución de conectivas respecto a cuantificadores.

si *x* no está libre en la otra subfórmula (en cuyo caso se habría renombrado en 1)

$$\vdash \forall x A \wedge C \leftrightarrow \forall x (A \wedge C)$$
$$\vdash \exists x A \wedge C \leftrightarrow \exists x (A \wedge C)$$
$$\vdash \forall x A \vee C \leftrightarrow \forall x (A \vee C)$$
$$\vdash \exists x A \vee C \leftrightarrow \exists x (A \vee C)$$
$$\vdash (\forall x A \rightarrow C) \leftrightarrow \exists x (A \rightarrow C) \quad \text{(recuerda: } \forall x A \rightarrow C \equiv \neg\forall x A \vee C\text{)}$$
$$\vdash (A \rightarrow \forall x C) \leftrightarrow \forall x (A \rightarrow C)$$
$$\vdash (\exists x A \rightarrow C) \leftrightarrow \forall x (A \rightarrow C)$$
$$\vdash (A \rightarrow \exists x C) \leftrightarrow \exists x (A \rightarrow C)$$

Cierre existencial

2. **Cierre Existencial.** Las variables libres de la fórmula se ligan existencialmente **poniendo el cuantificador correspondiente en cabeza de la fórmula**:

Lema: Una fórmula $A(x)$ es satisfacible sii $\exists x A(x)$ es satisfacible

- Ejemplo:

$$\forall y \exists z (p(x) \wedge q(y) \rightarrow r(f(z), x))$$

se transformaría en

$$\exists x \forall y \exists z (p(x) \wedge q(y) \rightarrow r(f(z), x))$$

Forma Normal Conjuntiva

3. **Forma Normal Conjuntiva - FNC.** Se utilizarán los siguientes teoremas de equivalencia:

   - Interdefinición (Eliminar bicondicionales y condicionales usando la equivalencia):

     $$\vdash (A \rightarrow B) \leftrightarrow (\neg A \vee B)$$
     $$\vdash (A \leftrightarrow B) \leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A)$$

   - Leyes de De Morgan (interiorizar negaciones usando equivalencias):

     $$\vdash \neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$$
     $$\vdash \neg(A \vee B) \leftrightarrow \neg A \wedge \neg B$$

   - Distribución de $\vee$ y $\wedge$:

     $$\vdash A \wedge (B \vee C) \leftrightarrow (A \wedge B) \vee (A \wedge C)$$ *(pueden ir en los dos sentidos)*
     $$\vdash A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C)$$

Lema: Para toda fórmula A, $\vdash A \leftrightarrow FNC(A)$
Lema: La forma normal conjuntiva de una fórmula siempre existe

Eliminación de existenciales

4. **Eliminación de cuantificadores existenciales.** Se elimina el cuantificador existencial sustituyendo la variable que ligaba por una función de Skolem o constante de Skolem.

- La función de Skolem será:
  - Una función nueva en la fórmula, aplicada a todas la variables cuantificadas universalmente que aparecen antes que el cuantificador existencial a eliminar.
  - Si no hay tales variables se utilizará una constante nueva en la fórmula para hacer la sustitución.
- Ejemplos:
  - $\forall x \exists y(p(x) \rightarrow \neg q(y))$ se transformaría en $\forall x(p(x) \rightarrow \neg q(f(x))$
  - $\exists x \forall z(q(x,z) \vee r(a,x))$ se transformaría en $\forall z(q(b,z) \vee r(a,b))$
  - $\exists x \forall y \exists z(p(x) \wedge q(y) \rightarrow r(f(z), x))$ se transformaría en
    $\forall y(p(a) \wedge q(y) \rightarrow r(f(g(y)), a))$

Lema: Una fórmula A es satisfacible sii Skolem(A) es satisfacible.

Ejercicio de forma clausal

- Obtener la forma clausular de la siguiente estructura deductiva   T [A 1 , A 2] |– B
  - A 1 : $\forall x\, P(x) \rightarrow \forall y\, Q(z,y) \vee Q(a)$
  - A 2 : $\exists x\, P(x) \vee \forall y\, \neg Q(y) \rightarrow \exists z\, P(a,z)$
  - B: $\exists x\, \forall y\, \neg P(x,y) \wedge \forall y\, \exists x\, Q(x,y)$

- A 1 $\equiv$ $\forall x\, P(x) \rightarrow \forall y\, Q(z,y) \vee Q(a)$
  - $\exists x\, \forall y\, (\neg P(x) \vee Q(z,y) \vee Q(a))$
  - $\exists z\, \exists x\, \forall y\, (\neg P(x) \vee Q(z,y) \vee Q(a))$
  - $\Rightarrow$ FC(A 1) = $\{ \neg P(b) \vee Q(c,y) \vee Q(a) \}$

- A 2 $\equiv \exists x\, P(x) \vee \forall y\, \neg Q(y) \rightarrow \exists z\, P(a,z)$
  - $\exists z\, \exists y\, \forall x\, ((\neg\, (P(x) \vee \neg\, Q(y)) \vee P(a,z))$
  - $\exists z\, \exists y\, \forall x\, (((\neg P(x) \wedge Q(y)) \vee P(a,z))$
  - $\exists z\, \exists y\, \forall x((\neg P(x) \vee P(a,z)) \wedge (Q(y) \vee P(a,z)))$
  - $\Rightarrow$ FC(A 2) = $\{ \neg P(x) \vee P(a,d)$, $Q(e) \vee P(a,d) \}$

Recuerda que siempre existe una forma prenexa, pero puede haber varias distintas.

Ejemplo, partiendo de:
A2 $\equiv \exists x\, P(x) \vee \forall y\, \neg Q(y) \rightarrow \exists z\, P(a,z)$,
y considerando F $\equiv$ $P(x) \vee \neg Q(y) \rightarrow P(a,z)$.

Podemos obtener: $\forall x \exists y \exists z(F)$, $\exists z \forall x \exists y(F)$, $\exists z \exists y \forall x(F)$, $\exists y \exists z \forall x(F)$… el orden en el que se extraen los cuantificadores afectará a su vez a la eliminación de existenciales.

También puede ser necesario exteriorizar cuantificadores antes de interiorizar negaciones: $\neg(\exists x\, \forall y\, \neg P(x,y) \rightarrow \forall y\, \exists x\, Q(x,y))$, pasa a: $\exists x\, \forall y\, \exists y'\, \forall x'\, \neg(\neg P(x,y) \rightarrow Q(x',y'))$

¡No olvides negar la conclusión si la hay!

- B $\equiv$ $\exists x\, \forall y\, \neg P(x,y) \wedge \forall y\, \exists x\, Q(x,y)$
  - $\neg$ B $\equiv$ $\forall x\, \exists y\, P(x,y) \vee \exists y\, \forall x\, \neg Q(x,y)$
  - $\forall x\, \exists y\, \exists y'\, \forall x'(P(x,y) \vee \neg Q(x',y'))$ (o, $\forall x\, \exists y\, \forall x'(P(x,y) \vee \neg Q(x',y)$ si no aplicas renombrado de y)
  - $\Rightarrow$ FC($\neg$B) = $\{ P(x,f(x)) \vee \neg Q(x',g(x)) \}$

- Forma clausular de la estructura deductiva:
  - FC = $\{ \neg P(b) \vee Q(c,y) \vee Q(a)$, $\neg P(x) \vee P(a,d)$, $Q(e) \vee P(a,d)$, $P(x,f(x)) \vee \neg Q(x',g(x)) \}$

Unificación

- **Algoritmo de Unificación:**

Sean A y B dos átomos con el mismo símbolo de predicado:

(1) $\alpha = \lambda$

(2) Mientras $A\alpha \neq B\alpha$:

(2.1) Encontrar el símbolo más a la izquierda en $A\alpha$ tal que el símbolo correspondiente en $B\alpha$ sea diferente

(2.2) Sean t_A y t_B los términos de $A\alpha$ y $B\alpha$ que empiezan con esos símbolos:

(a) Si ni t_A ni t_B no son variables o, si uno de ellos es una variable que aparece en el otro ➔ terminar con fallo (A y B no son unificables)

(b) En otro caso, sea t_A una variable ➔ el nuevo α es el resultado de $\alpha\{t_A/t_B\}$

(3) Terminar, siendo α el umg de A y B

Unificación

- **Algoritmo de Unificación:**

Ejemplo: $A \equiv P(x, x)$ y $B \equiv P(f(a), f(b))$

α	$A\alpha$	$B\alpha$	(t_A, t_B)
λ	$P(x, x)$	$P(f(a), f(b))$	$(x,f(a))$
$\{x/f(a)\}$	$P(f(a), f(a))$	$P(f(a), f(b))$	(a,b)

Fallo ➔ A y B no son unificables

Ejemplo: $A \equiv P(x, f(y))$ y $B \equiv P(z, x)$

α	$A\alpha$	$B\alpha$	(t_A, t_B)
λ	$P(x, f(y))$	$P(z, x)$	(x,z)
$\{x/z\}$	$P(z, f(y))$	$P(z, z)$	$(z, f(y))$
$\{x/f(y), z/f(y)\}$	$P(f(y), f(y))$	$P(f(y), f(y))$	

➔ A y B son unificables y su umg es $\{x/f(y), z/f(y)\}$

Resolución con Variables

- **Regla de resolución con umg:** Sean $L_1 \vee \ldots \vee L_n \vee C_1$ y $\neg L_1' \vee \ldots \vee \neg L_m' \vee C_2$ dos cláusulas, donde todos los L_{ij} son literales con el mismo símbolo de predicado. Puede deducirse una nueva cláusula $(C_1 \rho_1 \vee C_2 \rho_2)\beta$, llamada *resolvente*, donde
  - ρ_1 y ρ_2 son renombrados de todas las variables de cada cláusula que garantizan la no repetición de nombres entre ellas.
  - β es umg de $\{L_1 \rho_1, \ldots, L_n \rho_1, L_1'\rho_2, \ldots, L_m'\rho_2\}$

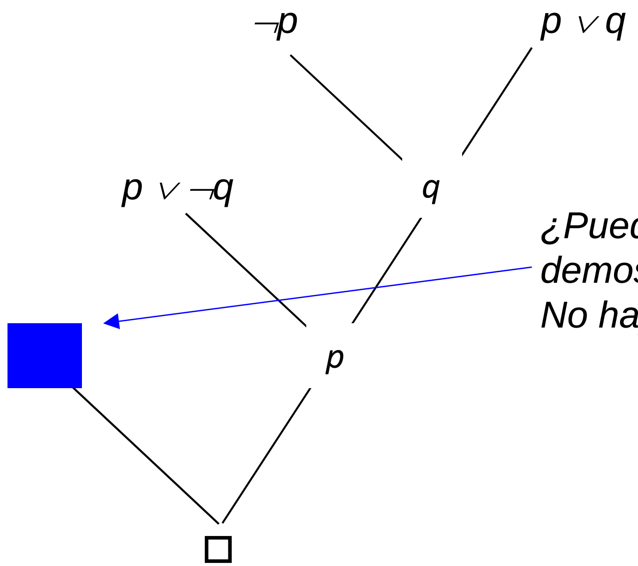
- La regla de resolución con umg se apoya en una versión de la **regla de factorización** para LPO: Dada una cláusula $L_1 \vee \ldots \vee L_n \vee C$, siendo $L_1, \ldots, L_n$ literales con el mismo símbolo de predicado, puede deducirse una nueva cláusula $L \vee C\beta$ donde
  - β es unificador de $L_1, \ldots, L_n$
  - $L = L_1\beta = \ldots = L_n\beta$

  Observa que la factorización se aplica a una misma clausula, ejemplo
  $\neg M(y_2, z_2, u) \vee \neg P(x_3) \vee \neg N(x_3, u) \vee \mathbf{N(x_3, y_2)} \vee \mathbf{N(x_3, z_2)}$,
  Podría quedar con $\{z_2 / y_2\}$ y factorización como
  $\neg M(y_2, \mathbf{y_2}, u) \vee \neg P(x_3) \vee \neg N(x_3, u) \vee N(x_3, y_2)$

  El literal L se denomina *factor* de $L_1 \vee \ldots \vee L_n \vee C$
  - La factorización nos permite eliminar información, pero es optativa. Los resolventes se pueden obtener con o sin ella.
  - Al factorizar, si los literales tienen las mismas variables, no se pueden cambiar los nombres.
  - En ocasiones aplicar factorización puede hacer la derivación imposible.
  - En ocasiones NO aplicar factorización puede hacer la derivación de $\square$ imposible.

Dos errores típicos (recordatorio de LP)

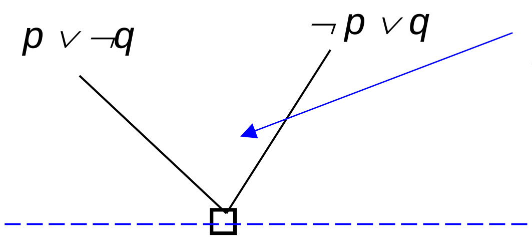
Conjunto de instancias básicas: $\{\neg p, p \vee q, p \vee \neg q\}$



¿Puedo volver a utilizar la clausula de $\neg p$? Claro, no afecta a la demostración como la iteración no afectaba en Deducción natural. No hay que olvidar que hay una conjunción implícita.

$$\neg(A \wedge B) \Leftrightarrow \neg A \vee \neg B$$

Conjunto de instancias básicas: $\{ p \vee \neg q, \neg p \vee q\}$



"…porque parece correcto". Pero no lo es. El resolvente es $p \vee \neg p$ o $q \vee \neg q$… pero no la clausula vacía. Si dudas, puedes usar (en sucio) tablas de verdad, árboles de verdad, o búsqueda de contramodelo.

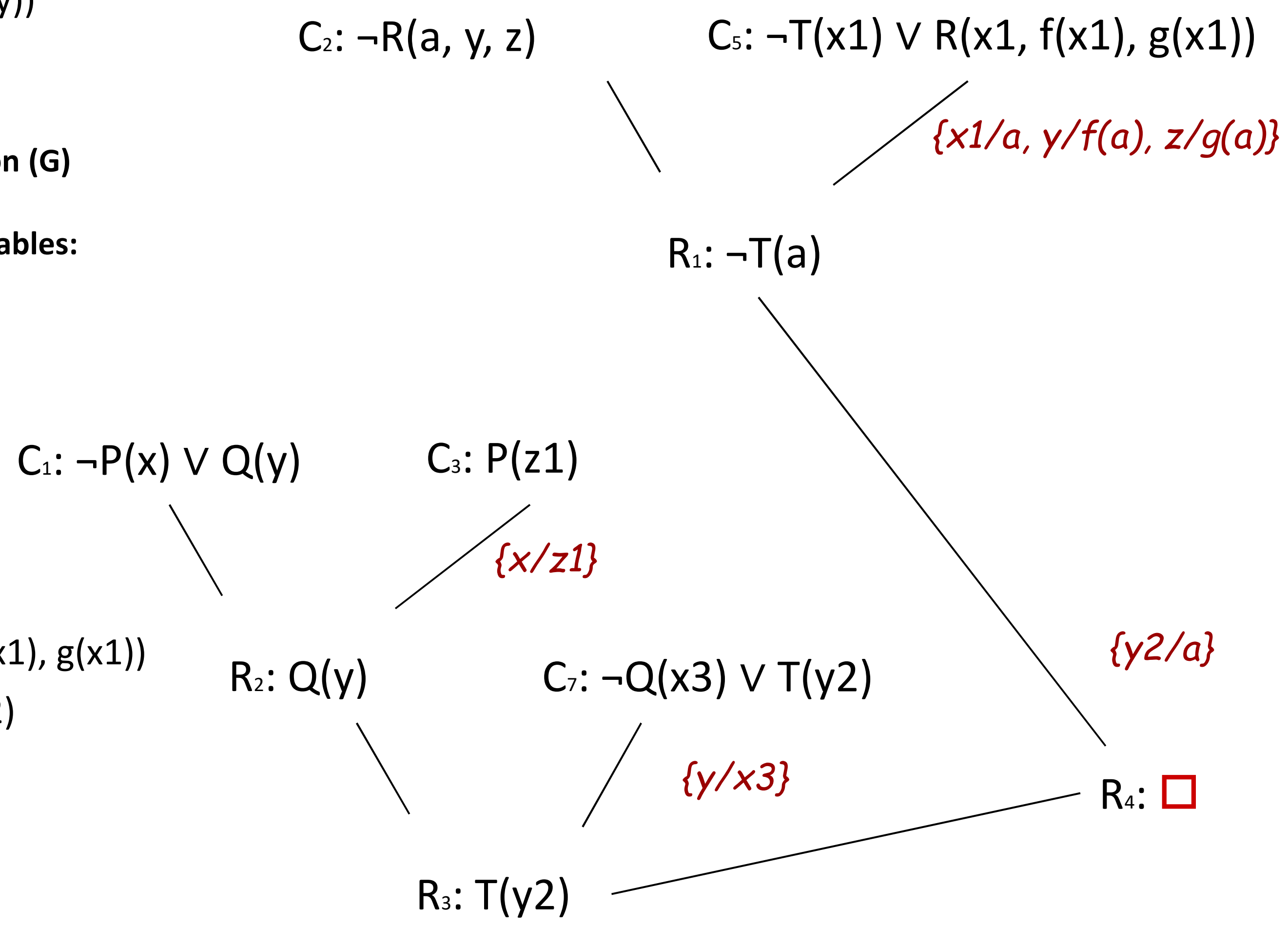
- *No tienes que usar todas las clausulas, lo importante es llegar a la clausula vacía (si es posible).*
- *Se permite el pequeño atajo de asumir que $A \vee \neg L$ y $A \vee L$, de A en lugar de $A \vee A$. También aplicar idempotencia en general: $A \vee A \vee B$ equivale a $A \vee B$*
- *Busca (y anota en sucio) las clausulas de un solo literal (o de un literal menos) que puedes obtener.*
- *El árbol de resolución suele (pero no necesariamente) empezar por las clausulas de la conclusión.*

Ejercicio 5: Comprobar por el Método de resolución la validez de T[A1,A2,A3] |— Q, siendo:

- A1 : $\exists x P(x) \rightarrow \forall x Q(x)$
- A2 : $\exists x \forall z(\neg \exists y R(x, y, z) \wedge P(z) \wedge L(x))$
- A3 : $\forall x \exists z((\neg T(x) \vee \exists y R(x, y, z)) \wedge (\neg L(z) \vee \neg T(x)))$
- Q : $\exists x \exists y \neg(Q(x) \rightarrow T(y))$

- **Tras:**
  - **negar conclusión (G)**
  - **pasar a FNS, y**
  - **renombrar variables:**

- C1: $\neg P(x) \vee Q(y)$
- C2: $\neg R(a, y1, z)$
- C3: $P(z1)$
- C4: $L(a)$
- C5: $\neg T(x1) \vee R(x1, f(x1), g(x1))$
- C6: $\neg L(g(x2)) \vee \neg T(x2)$
- C7: $\neg Q(x3) \vee T(y2)$



Examen Julio 2015

Ejercicio 4. Demuestra por resolución UMG (comenzando con la cláusula C7), que el siguiente conjunto de cláusulas es insatisfacible: *(2,5 puntos)*

C1: ¬P(x) ∨ ¬Q(y,z,w) ∨ ¬R(x,w) ∨ R(x,y)

C2: Q(a,f(b),f(c))

C3: Q(x,x,f(x))

C4: ¬Q(x,y,z) ∨ R(x,z)

C5: P(a)

C6: ¬R(a,c) ∨ ¬S(f(x))

C7: S(f(x)) ∨ ¬P(x)

SOLUCIÓN:

R1: S(f(a))	C7, C5 {x7/a}
R2: ¬R(a, c)	R1, C6 {x6/a}
R3: ¬P(a) V ¬Q(c, z1, w1) V ¬R(a, w1)	R2, C1 {x1/a, y1/c}
R4: ¬Q(c, z1, w1) V ¬R(a, w1)	R3, C5 {}
R5: ¬R(a, f(c))	R4, C3 {x3/c, z1/c, w1/f(c)}
R6: ¬Q(a, y4, f(c))	R5, C4 {x4/a, z4/f(c)}
R7: □	R6, C2 {y4/f(b)}

Más consejos:

- Al pasar a forma clausular, no es necesario incluir formulas para cada paso como en deducción natural, pero sí se recomienda: etiquetar las sustituciones, y seguir los pasos en el orden que se han dado en clase: prenex (renombrado,…), cierre, FNC,FNS.
- Recuerda que, en el paso de FNS, se deben usar símbolos de funciones y constantes nuevos. Así que si en una formula (A1) ya has usado "f" o "a", en otra formula A2 no podrás usarlos (sino que serán "g" o "b" asumiendo que no hayan aparecido ya).
- En resolución, no olvides negar la conclusión si la hay.
- Renombra todas las variables con el índice de la clausula: x7 es la variable x en la clausula 7.
- Aunque puedo usar los resolventes y clausulas de partida varias veces, es necesario renombrar sus variables para que **no haya dos sustituciones distintas de la misma variable** (dando C2' con x2').
- Puede ser necesario factorizar o aplicar idempotencia, aunque no es muy típico.
- Una solución bien indexada y con un resolvente por línea (como en este ejercicio), puede ser más cómoda que un árbol de resolución. En cualquier caso, **hay que etiquetar las clausulas y resolventes que se van usando** (C1, C2,C2', R1…) para facilitar la corrección.