Temas de “Lógica informática” (2005–06)

José A. Alonso Jiménez

Grupo de Lógica Computacional
Dpto. de Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, 1 de Febrero de 2006 (Actualizado el 10 de mayo de 2006)

Índice general

Capítulo 1

Sintaxis y semántica de la lógica proposicional

Lógica informática (2005–06)
Tema 1: Sintaxis y semántica de la lógica proposicional

José A. Alonso Jiménez

Grupo de Lógica Computacional
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

Lógica

- Objetivos de la lógica:
  - La formalización del lenguaje natural.
  - Los métodos de razonamiento.
- Sistemas lógicos:
  - Lógica proposicional.
  - Lógica de primer orden.
  - Lógicas modales.
- Aplicaciones de la lógica en computación:
  - Programación lógica.
  - Verificación y síntesis automática de programas.
  - Representación del conocimiento y razonamiento.
  - Modelización y razonamiento sobre sistemas.

Argumentos y formalización

- Ejemplos de argumentos:
  - Ejemplo 1: Si el tren llega a las 7 y no hay taxis en la estación, entonces Juan llegará tarde a la reunión. Juan no ha llegado tarde a la reunión. El tren llegó a las 7. *Por tanto,* habían taxis en la estación.
  - Ejemplo 2: Si hay corriente y la lámpara no está fundida, entonces está encendida. La lámpara no está encendida. Hay corriente. *Por tanto,* la lámpara está fundida.
- Formalización:
  - Simbolización:

Simb.	Ejemplo 1	Ejemplo 2
p	el tren llega a las 7	hay corriente
q	hay taxis en la estación	la lámpara está fundida
r	Juan llega tarde a la reunión	la lámpara está encendida

  - Si p y no q, entonces r. No r. p. Por tanto, q.
  - $p \wedge \neg q \rightarrow r, \ \neg r, \ p \models q$.

Sintaxis proposicional: Fórmulas proposicionales

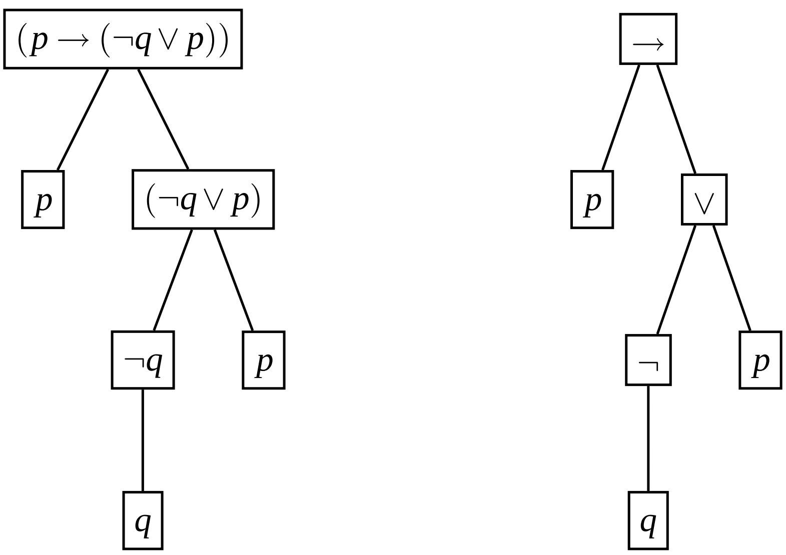
- Alfabeto proposicional:
  - variables proposicionales: $p_0, p_1, \dots; p, q, r$.
  - conectivas lógicas:
    - monaria: $\neg$ (negación),
    - binarias: $\wedge$ (conjunción), $\vee$ (disyunción),
      $\rightarrow$ (condicional), $\leftrightarrow$ (bicondicional).
  - símbolos auxiliares: "(" y ")".
- Fórmulas proposicionales:
  - Definición:
    - Las variables proposicionales son fórmulas.
    - Si F y G son fórmulas, entonces también lo son $\neg F$, $(F \wedge G)$, $(F \vee G)$, $(F \rightarrow G)$ y $(F \leftrightarrow G)$
  - Ejemplos:
    - Fórmulas: p, $(p \vee \neg q)$, $\neg(p \vee p)$, $((p \rightarrow q) \vee (q \rightarrow p))$
    - No fórmulas: (p), $p \vee \neg q$, $(p \vee \wedge q)$

Fórmulas proposicionales (BNF)

- Notaciones:
  - $p, q, r, \dots$ representarán variables proposicionales.
  - $F, G, H, \dots$ representarán fórmulas.
  - VP representa el conjunto de los variables proposicionales.
  - Prop representa el conjunto de las fórmulas.
  - $*$ representa una conectiva binaria.
- Forma de Backus Naur (BNF) de las fórmula proposicionales:
  - $F ::= p \mid \neg G \mid (F \land G) \mid (F \lor G) \mid (F \rightarrow G) \mid (F \leftrightarrow G)$.

Árboles de análisis

- Árboles de análisis (o de formación).



Omisión de paréntesis

- Criterios de reducción de paréntesis:
  - Pueden eliminarse los paréntesis externos.

    $F \wedge G$ es una abreviatura de $(F \wedge G)$.
  - Precedencia de asociación de conectivas: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.

    $F \wedge G \rightarrow \neg F \vee G$ es una abreviatura de $((F \wedge G) \rightarrow (\neg F \vee G))$.
  - Cuando una conectiva se usa repetidamente, se asocia por la derecha.

$F \vee G \vee H$	abrevia	$(F \vee (G \vee H))$
$F \wedge G \wedge H \rightarrow \neg F \vee G$	abrevia	$((F \wedge (G \wedge H)) \rightarrow (\neg F \vee G))$

7

Subfórmulas

- Subfórmulas:
  - Def: El conjunto Subf(F) de las subfórmulas de una fórmula F se define recursivamente por:

$$\text{Subf}(F) = \begin{cases} \{F\}, & \text{si } F \text{ es una variable;} \\ \{F\} \cup \text{Subf}(G), & \text{si } F = \neg G; \\ \{F\} \cup \text{Subf}(G) \cup \text{Subf}(H), & \text{si } F = G * H \end{cases}$$

  - Ejemplos:
    - Subf$(p) = \{p\}$
    - Subf$(q) = \{q\}$
    - Subf$(\neg q) = \{\neg q, q\}$
    - Subf$(\neg q \vee p) = \{\neg q \vee p, \neg q, q, p\}$
    - Subf$(p \rightarrow \neg q \vee p) = \{p \rightarrow \neg q \vee p, p, \neg q \vee p, \neg q, q\}$

8

Semántica proposicional: valores y funciones de verdad

- Valores de verdad ($\mathbb{B}$): 1: verdadero y 0: falso.
- Funciones de verdad:
  - $H_\neg : \{0,1\} \to \{0,1\}$ t.q. $H_\neg(i) = \begin{cases} 1, & \text{si } i = 0; \\ 0, & \text{si } i = 1. \end{cases}$
  - $H_\wedge : \{0,1\}^2 \to \{0,1\}$ t.q. $H_\wedge(i,j) = \begin{cases} 1, & \text{si } i = j = 1; \\ 0, & \text{en otro caso.} \end{cases}$
  - $H_\vee : \{0,1\}^2 \to \{0,1\}$ t.q. $H_\vee(i,j) = \begin{cases} 0, & \text{si } i = j = 0; \\ 1, & \text{en otro caso.} \end{cases}$
  - $H_\to : \{0,1\}^2 \to \{0,1\}$ t.q. $H_\to(i,j) = \begin{cases} 0, & \text{si } i = 1, j = 0; \\ 1, & \text{en otro caso.} \end{cases}$
  - $H_\leftrightarrow : \{0,1\}^2 \to \{0,1\}$ t.q. $H_\leftrightarrow(i,j) = \begin{cases} 1, & \text{si } i = j; \\ 0, & \text{en otro caso.} \end{cases}$

9

Valoración de fórmulas

- Funciones de verdad mediante tablas de verdad:

i	$\neg i$
1	0
0	1

i	j	$i \wedge j$	$i \vee j$	$i \to j$	$i \leftrightarrow j$
1	1	1	1	1	1
1	0	0	1	0	0
0	1	0	1	1	0
0	0	0	0	1	1

- Valoración de verdad:
  - Def.: Una valoración de verdad es una aplicación $v : \mathrm{VP} \to \mathbb{B}$.
  - Prop: Para cada valoración de verdad v existe una única aplicación $\hat{v} : \mathrm{Prop} \to \mathbb{B}$ tal que:

    $$\hat{v}(F) = \begin{cases} v(F), & \text{si } F \text{ es una variable;} \\ H_\neg(\hat{v}(G)), & \text{si } F = \neg G; \\ H_*(\hat{v}(G), \hat{v}(H)), & \text{si } F = G * H \end{cases}$$

    Se dice que $\hat{v}(F)$ es el valor de verdad de F respecto de v.

10

Valoración de fórmulas

- Ejemplo: Sea $F = (p \lor q) \land (\neg q \lor r)$
  - valor de F en una valoración v_1 tal que $v_1(p) = v_1(r) = 1, v_1(q) = 0$

$$\begin{array}{ccccccc} (p & \lor & q) & \land & (\neg q & \lor & r) \\ (1 & \lor & 0) & \land & (\neg 0 & \lor & 1) \\ & 1 & & \land & (1 & \lor & 1) \\ & 1 & & \land & & 1 & \\ & & & 1 & & & \end{array}$$

  - valor de F en una valoración v_2 tal que $v_2(r) = 1, v_2(p) = v_2(q) = 0$

$$\begin{array}{ccccccc} (p & \lor & q) & \land & (\neg q & \lor & r) \\ 0 & 0 & 0 & 0 & 10 & 1 & 1 \end{array}$$

- Prop.: Sea F una fórmula y v, v' dos valoraciones. Si $v(p) = v'(p)$ para todos las variables proposicionales de F, entonces $\hat{v}(F) = \widehat{v'}(F)$.
- Notación: Se escribe $v(F)$ en lugar de $\hat{v}(F)$.

11

Modelos y satisfacibilidad

- Modelo de una fórmula
  - Def.: v es modelo de F si $v(F) = 1$.
  - Notación: $v \models F$.
  - Ejemplo (continuación del anterior):
    – si $v_1(p) = v_1(r) = 1, v_1(q) = 0$, entonces $v_1 \models (p \lor q) \land (\neg q \lor r)$
    – si $v_2(r) = 1, v_2(p) = v_2(q) = 0$, entonces $v_2 \not\models (p \lor q) \land (\neg q \lor r)$.
- Fórmulas satisfacibles e insatisfacibles
  - Def.: F es satisfacible si F tiene algún modelo.
  - Ejemplo: $(p \rightarrow q) \land (q \rightarrow r)$ es satisfacible
    $v(p) = v(q) = v(r) = 0$.
  - Def.: F es insatisfacible si F no tiene ningún modelo.
  - Ejemplo: $p \land \neg p$ es insatisfacible

p	$\neg p$	$p \land \neg p$
1	0	0
0	1	0

12

Tautologías y contradicciones

- Tautologías y contradicciones:
  - Def.: F es una tautología (o válida) si toda valoración es modelo de F. Se representa por $\models F$.
  - Def.: F es una contradicción si ninguna valoración es modelo de F.
  - Def.: F es contingente si no es tautología ni contradicción.
  - Ejemplos:
    1. $(p \to q) \lor (q \to p)$ es una tautología.
    2. $(p \to q) \land \neg(p \to q)$ es una contradicción.
    3. $p \to q$ es contingente.

p	q	$p \to q$	$q \to p$	$(p \to q) \lor (q \to p)$	$\neg(p \to q)$	$(p \to q) \land \neg(p \to q)$
1	1	1	1	1	0	0
1	0	0	1	1	1	0
0	1	1	0	1	0	0
0	0	1	1	1	0	0

Clasificaciones de fórmulas

- Clasificaciones de fórmulas:

Todas las fórmulas		
Tautologías	*Contigentes*	*Contradicciones*
Verdadera en todas las valoraciones (ej. $p \lor \neg p$)	Verdadera en algunas valoraciones y falsa en otras (ej. $p \to q$)	Falsa en todas las valoraciones (ej. $p \land \neg p$)
Safisfacibles		*Insatisfacibles*
Todas las fórmulas		

Satisfacibilidad y validez

- Los problemas SAT y TAUT:
  - Problema SAT: Dada F determinar si es satisfacible.
  - Problema TAUT: Dada F determinar si es una tautología.
- Relaciones entre satisfacibilidad y tautologicidad:
  - F es tautología $\iff$ $\neg F$ es insatisfacible..
  - F es tautología $\implies$ F es satisfacible..
  - F es satisfacible $\not\Longrightarrow$ $\neg F$ es insatisfacible.

$p \rightarrow q$ es satisfacible.
$$v(p) = v(q) = 1$$
$\neg(p \rightarrow q)$ es satisfacible.
$$v(p) = 1, v(q) = 0.$$

Algoritmos para SAT y TAUT

- Tablas de verdad para $\models (p \rightarrow q) \lor (q \rightarrow p)$
  - Tabla de verdad:

p	q	$(p \rightarrow q)$	$(q \rightarrow p)$	$(p \rightarrow q) \lor (q \rightarrow p)$
1	1	1	1	1
1	0	0	1	1
0	1	1	0	1
0	0	1	1	1

  - Tabla de verdad simplificada:

p	q	$(p$	$\rightarrow$	$q)$	$\lor$	$(q$	$\rightarrow$	$p)$
1	1	1	1	1	1	1	1	1
1	0	1	0	0	1	0	1	1
0	1	0	1	1	1	1	0	0
0	0	0	1	0	1	0	1	0

Algoritmos para SAT y TAUT

- Método de Quine.
  - Método de Quine para $\models (p \rightarrow q) \vee (q \rightarrow p)$

$(p$	$\rightarrow$	$q)$	$\vee$	$(q$	$\rightarrow$	$p)$
			0			
	0				0	
				1		0
0		1				
	1					

  - Método de Quine para $\models (p \rightarrow q) \vee (q \rightarrow p)$

$(p$	$\rightarrow$	$q)$	$\vee$	$(q$	$\rightarrow$	$p)$
0	0	1	0	1	0	0
	$1*$					

Algoritmos para SAT y TAUT

- Algoritmos de decisión para SAT y TAUT:
  - Tablas de verdad para $\not\models (p \leftrightarrow q) \vee (q \leftrightarrow p)$

p	q	$(p \leftrightarrow q)$	$(q \leftrightarrow p)$	$(p \leftrightarrow q) \vee (q \leftrightarrow p)$
1	1	1	1	1
1	0	0	0	0
0	1	0	0	0
0	0	1	1	1

  - Método de Quine para $\not\models (p \leftrightarrow q) \vee (q \leftrightarrow p)$

$(p$	$\leftrightarrow$	$q)$	$\vee$	$(q$	$\leftrightarrow$	$p)$
0	0	1	0	1	0	0
1	0	0	0	0	0	1

Selección de tautologías

- Selección de tautologías
  - 1. $F \to F$ (ley de identidad).
  - 2. $F \vee \neg F$ (ley del tercio excluso).
  - 3. $\neg(F \wedge \neg F)$ (principio de no contradicción).
  - 4. $(\neg F \to F) \to F$ (ley de Clavius).
  - 5. $\neg F \to (F \to G)$ (ley de Duns Scoto).
  - 6. $((F \to G) \to F) \to F$ (ley de Peirce).
  - 7. $(F \to G) \wedge F \to G$ (modus ponens).
  - 8. $(F \to G) \wedge \neg G \to \neg F$ (modus tollens).

Modelo de conjuntos de fórmulas

- Notación:
  - $S, S_1, S_2, \dots$ representarán conjuntos de fórmulas.
- Modelo de un conjunto de fórmulas:
  - Def.: v es modelo de S si para toda $F \in S$ se tiene que $v \models F$.
  - Representación: $v \models S$.
  - Ejemplo: Sea $S = \{(p \vee q) \wedge (\neg q \vee r), q \to r\}$

    La valoración v_1 tal que $v_1(p) = 1, v_1(q) = 0, v_1(r) = 1$ es modelo de S ($v_1 \models S$).

$\{(p$	$\vee$	$q)$	$\wedge$	$(\neg$	q	$\vee$	$r),$	q	$\to$	$r\}$
1	1	0	1	1	0	1	1	0	1	1

    La valoración v_2 tal que $v_2(p) = 0, v_2(q) = 1, v_2(r) = 0$ no es modelo de S ($v_2 \not\models S$).

$\{(p$	$\vee$	$q)$	$\wedge$	$(\neg$	q	$\vee$	$r),$	q	$\to$	$r\}$
0	1	0	0	0	1	0	0	1	0	0

Consistencia

- Conjunto consistente de fórmulas:
  - Def.: S es consistente si S tiene algún modelo.
  - Def.: S es inconsistente si S no tiene ningún modelo.
  - Ejemplos:
    – $\{(p \lor q) \land (\neg q \lor r), p \to r\}$ es consistente (con modelos v_4, v_6, v_8)
    – $\{(p \lor q) \land (\neg q \lor r), p \to r, \neg r\}$ es inconsistente

	p	q	r	$(p \lor q)$	$(\neg q \lor r)$	$(p \lor q) \land (\neg q \lor r)$	$p \to r$	$\neg r$
v_1	0	0	0	0	1	0	1	1
v_2	0	0	1	0	1	0	1	0
v_3	0	1	0	1	0	0	1	1
v_4	0	1	1	1	1	1	1	0
v_5	1	0	0	1	1	1	0	1
v_6	1	0	1	1	1	1	1	0
v_7	1	1	0	1	0	0	0	1
v_8	1	1	1	1	1	1	1	0

Consecuencia lógica

- Consecuencia lógica:
  - Def.: F es consecuencia de S si todos los modelos de S son modelos de F.
  - Representación: $S \models F$.
  - Ejemplos: $\{p \to q, q \to r\} \models p \to r$ y $\{p\} \not\models p \land q$

	p	q	r	$p \to q$	$q \to r$	$p \to r$
v_1	0	0	0	1	1	1
v_2	0	0	1	1	1	1
v_3	0	1	0	1	0	1
v_4	0	1	1	1	1	1
v_5	1	0	0	0	1	0
v_6	1	0	1	0	1	1
v_7	1	1	0	1	0	0
v_8	1	1	1	1	1	1

p	q	$p \land q$
1	1	1
1	0	0
0	1	0
0	0	0

Propiedades de la consecuencia

- Propiedades básicas de la relación de consecuencia:
  - Reflexividad: $S \models S$.
  - Monotonía: Si $S_1 \models F$ y $S_1 \subseteq S_2$, entonces $S_2 \models F$.
  - Transitividad: Si $S \models F$ y $\{F\} \models G$, entonces $S \models G$.
- Relación entre consecuencia, validez, satisfacibilidad y consistencia:
  - Las siguientes condiciones son equivalentes:
    1. $\{F_1, \dots, F_n\} \models G$
    2. $\models F_1 \wedge \dots \wedge F_n \rightarrow G$
    3. $\neg(F_1 \wedge \dots \wedge F_n \rightarrow G)$ es insatisfacible
    4. $\{F_1, \dots, F_n, \neg G\}$ es inconsistente

Argumentaciones

- Ejemplo de argumentación:
  - Problema de los animales: Se sabe que
    1. Los animales con pelo que dan leche son mamíferos.
    2. Los mamíferos que tienen pezuñas o que rumian son ungulados.
    3. Los ungulados de cuello largo son jirafas.
    4. Los ungulados con rayas negras son cebras.

    Se observa un animal que tiene pelos, pezuñas y rayas negras. Por consiguiente, se concluye que el animal es una cebra.
  - Formalización:

$$\begin{array}{l}\{\ \text{tiene\_pelos} \vee \text{da\_leche} \rightarrow \text{es\_mamífero}, \\ \quad \text{es\_mamífero} \wedge (\text{tiene\_pezuñas} \vee \text{rumia}) \rightarrow \text{es\_ungulado}, \\ \quad \text{es\_ungulado} \wedge \text{tiene\_cuello\_largo} \rightarrow \text{es\_jirafa}, \\ \quad \text{es\_ungulado} \wedge \text{tiene\_rayas\_negras} \rightarrow \text{es\_cebra}, \\ \quad \text{tiene\_pelos} \wedge \text{tiene\_pezuñas} \wedge \text{tiene\_rayas\_negras}\} \\ \models \text{es\_cebra}\end{array}$$

Problemas lógicos: veraces y mentirosos

- Enunciado: En una isla hay dos tribus, la de los veraces (que siempre dicen la verdad) y la de los mentirosos (que siempre mienten). Un viajero se encuentra con tres isleños A, B y C y cada uno le dice una frase
  1. A dice "B y C son veraces syss C es veraz"
  2. B dice "Si A y C son veraces, entonces B y C son veraces y A es mentiroso"
  3. C dice "B es mentiroso syss A o B es veraz"

  Determinar a qué tribu pertenecen A, B y C.
- Simbolización: a: "A es veraz", b: "B es veraz", c: "C es veraz".
- Formalización:
  $F_1 = a \leftrightarrow (b \wedge c \leftrightarrow c)$, $F_2 = b \leftrightarrow (a \wedge c \rightarrow b \wedge c \wedge \neg a)$ y $F_3 = c \leftrightarrow (\neg b \leftrightarrow a \vee b)$.
- Modelos de $\{F_1, F_2, F_3\}$:
  Si v es modelo de $\{F_1, F_2, F_3\}$, entonces $v(a) = 1, v(b) = 1, v(c) = 0$.
- Conclusión: A y B son veraces y C es mentiroso.

Bibliografía

1. C. Badesa, I. Jané y R. Jansana *Elementos de lógica formal.* (Ariel, 2000)

   Cap. 0 (Introducción), 6 (Sintaxis de la lógica proposicional), 7 (Semántica de la lógica proposicional), 9 (Consecuencia lógica) y 11 (Lógica proposicional y lenguaje natural).
2. M. Ben–Ari, *Mathematical logic for computer science (2nd ed.).* (Springer, 2001)

   Cap. 1 (Introduction) y 2 (Propositional calculus: formulas, models, tableaux).
3. M. Huth y M. Ryan *Logic in computer science: modelling and reasoning about systems.* (Cambridge University Press, 2000)

   Cap. 1 (Propositional logic).
4. E. Paniagua, J.L. Sánchez y F. Martín *Lógica computacional* (Thomson, 2003)

   Cap. 1 (La sintaxis de la Lógica) y Cap. 2 (La semántica de la Lógica).

Capítulo 2

Deducción natural proposicional

Lógica informática (2005–06)
Tema 2: Deducción natural proposicional

José A. Alonso Jiménez

Grupo de Lógica Computacional
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

DN: Reglas de la conjunción

- Regla de introducción de la conjunción: $\dfrac{F \quad G}{F \land G}\land i$
- Reglas de eliminación de la conjunción: $\dfrac{F \land G}{F}\land e \qquad \dfrac{F \land G}{G}\land e$
- Ejemplo: $p \land q, r \vdash q \land r$:

1	$p \land q$	premisa
2	r	premisa
3	q	$\land$e 1
4	$q \land r$	$\land$i 2,3

- Adecuación de las reglas de la conjunción:
  - $\land$i : $\{F, G\} \models F \land G$
  - $\land$e : $F \land G \models F$
  - $\land$e : $F \land G \models G$

DN: Reglas de la doble negación

- Regla de eliminación de la doble negación: $\dfrac{\neg\neg F}{F}\ \neg\neg e$
- Regla de introducción de la doble negación: $\dfrac{F}{\neg\neg F}\ \neg\neg i$
- Ejemplo: $p, \neg\neg(q \land r) \vdash \neg\neg p \land r$:

1	p	premisa
2	$\neg\neg(q \land r)$	premisa
3	$\neg\neg p$	$\neg\neg$i 1
4	$q \land r$	$\neg\neg$e 2
5	r	$\land$e 4
6	$\neg\neg p \land r$	$\land$i 3,5

- Adecuación de las reglas de la doble negación:
  - $\neg\neg$e : $\{\neg\neg F\} \models F$
  - $\neg\neg$i : $\{F\} \models \neg\neg F$

DN: Regla de eliminación del condicional

- Regla de eliminación del condicional: $\dfrac{F \quad F \to G}{G} \to e$
- Ejemplo: $\neg p \land q, \neg p \land q \to r \lor \neg p \vdash r \lor \neg p$:

1	$\neg p \land q$	premisa
2	$\neg p \land q \to r \lor \neg p$	premisa
3	$r \lor \neg p$	$\to$e 1,2

- Ejemplo: $p, p \to q, p \to (q \to r) \vdash r$:

1	p	premisa
2	$p \to q$	premisa
3	$p \to (q \to r)$	premisa
4	q	$\to$e 1,2
5	$q \to r$	$\to$e 1,3
6	r	$\to$e 4,5

- Adecuación de la eliminación del condicional: $\{F, F \to G\} \models G$

DN: Regla derivada de modus tollens (MT)

- Regla derivada de modus tollens: $\dfrac{F \to G \quad \neg G}{\neg F}\, MT$
- Ejemplo: $p \to (q \to r), p, \neg r \vdash \neg q$:

1	$p \to (q \to r)$	premisa
2	p	premisa
3	$\neg r$	premisa
4	$q \to r$	$\to$e $1,2$
5	$\neg q$	MT $3,4$

- Ejemplo: $\neg p \to q, \neg q \vdash p$:

1	$\neg p \to q$	premisa
2	$\neg q$	premisa
3	$\neg\neg p$	MT $1,2$
4	p	$\neg\neg$e 3

DN: Regla de introducción del condicional

- Regla de introducción del condicional: $\dfrac{\boxed{\begin{array}{c}F\\ \vdots\\ G\end{array}}}{F \to G} \to \text{i}$
- Ejemplo: $p \to q \vdash \neg q \to \neg p$:

1	$p \to q$	premisa
2	$\neg q$	supuesto
3	$\neg p$	MT $1,2$
4	$\neg q \to \neg p$	$\to$i $2-3$

- Adecuación de la regla de introducción del condicional:
  Si $F \models G$, entonces $\models F \to G$.

DN: Regla de introducción del condicional

- Ejemplo: $\neg q \to \neg p \vdash p \to \neg\neg q$:

1	$\neg q \to \neg p$	premisa
2	p	supuesto
3	$\neg\neg p$	$\neg\neg$i 2
4	$\neg\neg q$	MT $1,3$
5	$p \to \neg\neg q$	$\to$i $2-4$

- Ejemplo (de teorema): $\vdash p \to p$:

1	p	supuesto
2	$p \to p$	$\to$i $1-1$

DN: Regla de introducción del condicional

- Ejemplo: $\vdash (q \to r) \to ((\neg q \to \neg p) \to (p \to r))$:

1	$q \to r$	supuesto
2	$\neg q \to \neg p$	supuesto
3	p	supuesto
4	$\neg\neg p$	$\neg\neg$i 3
5	$\neg\neg q$	MT $2,4$
6	q	$\neg\neg$e 5
7	r	$\to$e $1,6$
8	$p \to r$	$\to$i $3-7$
9	$(\neg q \to \neg p) \to (p \to r)$	$\to$i $2-8$
10	$(q \to r) \to ((\neg q \to \neg p) \to (p \to r))$	$\to$i $1-9$

DN: Reglas de la disyunción

- Reglas de introducción de la disyunción: $\dfrac{F}{F \lor G}\lor i_1 \qquad \dfrac{G}{F \lor G}\lor i_2$

- Regla de eliminación de la disyunción: $$\dfrac{F \lor G \quad \boxed{\begin{array}{c} F \\ \vdots \\ H \end{array}} \quad \boxed{\begin{array}{c} G \\ \vdots \\ H \end{array}}}{H}\lor e$$

- Ejemplo: $p \lor q \vdash q \lor p$:

1	$p \lor q$	premisa
2	p	supuesto
3	$q \lor p$	$\lor$i 2
4	q	supuesto
5	$q \lor p$	$\lor$i 4
6	$q \lor p$	$\lor$e $1, 2-3, 4-5$

DN: Reglas de la disyunción

- Ejemplo: $q \rightarrow r \vdash p \lor q \rightarrow p \lor r$:

1	$p \rightarrow r$	premisa
2	$p \lor q$	supuesto
3	p	supuesto
4	$p \lor r$	$\lor$i 3
5	q	supuesto
6	r	$\rightarrow$e $1, 5$
7	$p \lor r$	$\lor$i 6
8	$p \lor r$	$\lor$e $2, 3-4, 5-7$
9	$p \lor q \rightarrow p \lor r$	$\rightarrow$i $2-8$

DN: Regla de copia

- Ejemplo (usando la regla hyp): $\vdash p \to (q \to p)$:

1	p	supuesto
2	q	supuesto
3	p	hyp 1
4	$q \to p$	$\to$i $2-3$
5	$p \to (q \to p)$	$\to$i $1-4$

DN: Reglas de la negación

- Extensiones de la lógica para usar falso:
  - Extensión de la sintaxis: $\perp$ es una fórmula proposicional.
  - Extensión de la semántica: $v(\perp) = 0$ en cualquier valoración.
- Reglas de la negación:
  - Regla de eliminación de lo falso: $\dfrac{\perp}{F}\ \perp e$
  - Regla de eliminación de la negación: $\dfrac{F \quad \neg F}{\perp}\ \neg e$
- Adecuación de las reglas de la negación:
  - $\perp \models F$
  - $\{F, \neg F\} \models \perp$

DN: Reglas de la negación

- Ejemplo: $\neg p \vee q \vdash p \rightarrow q$:

1	$\neg p \vee q$	premisa
2	p	supuesto
3	$\neg p$	supuesto
4	$\bot$	$\neg$e $2,3$
5	q	$\bot$e 4
6	q	supuesto
7	q	$\vee$e $1,3-5,6-6$
8	$p \rightarrow q$	$\rightarrow$i $2-7$

DN: Reglas de la negación

- Regla de introducción de la negación: $$\frac{\boxed{\begin{array}{c}F\\ \vdots\\ \bot\end{array}}}{\neg F}\ \neg\mathrm{i}$$
- Adecuación: Si $F \models \bot$, entonces $\models \neg F$.
- Ejemplo: $p \rightarrow q, p \rightarrow \neg q \vdash \neg p$:

1	$p \rightarrow q$	premisa
2	$p \rightarrow \neg q$	premisa
3	p	supuesto
4	q	$\rightarrow$e $1,3$
5	$\neg q$	$\rightarrow$e $2,3$
6	$\bot$	$\neg$e $4,5$
7	$\neg p$	$\neg$i $2-5$

DN: Reglas del bicondicional

- Regla de introducción del bicondicional: $\dfrac{F \rightarrow G \quad G \rightarrow F}{F \leftrightarrow G} \leftrightarrow \mathrm{i}$
- Ejemplo: $p \land q \leftrightarrow q \land p$:

1	$p \land q$	supuesto
2	p	$\land$e 1
3	q	$\land$e 1
4	$q \land p$	$\land$i 2, 3
5	$p \land q \rightarrow q \land p$	$\rightarrow$i 1 – 4
6	$q \land p$	supuesto
7	q	$\land$e 6
8	p	$\land$e 6
9	$p \land q$	$\land$i 7, 8
10	$q \land p \rightarrow p \land q$	$\rightarrow$i 6 – 9
11	$p \land q \leftrightarrow q \land p$	$\leftrightarrow$i 5, 10

15

DN: Reglas del bicondicional

- Reglas de eliminación del bicondicional: $\dfrac{F \leftrightarrow G}{F \rightarrow G} \leftrightarrow \mathrm{e}_1 \qquad \dfrac{F \leftrightarrow G}{G \rightarrow F} \leftrightarrow \mathrm{e}_2$
- Ejemplo: $p \leftrightarrow q, p \lor q \vdash p \land q$:

1	$p \leftrightarrow q$	premisa
2	$p \lor q$	premisa
3	p	supuesto
4	$p \rightarrow q$	$\leftrightarrow$e 1
5	q	$\rightarrow$e 4, 3
6	$p \land q$	$\land$i 3, 5
7	q	supuesto
8	$q \rightarrow p$	$\leftrightarrow$e 1
9	p	$\rightarrow$e 8, 7
10	$p \land q$	$\land$i 7, 9
11	$p \land q$	$\lor$e 2, 3 – 6, 7 – 10

16

DN: Reglas derivadas: modus tollens

- Regla derivada de modus tollens (MT):

$$\frac{F \rightarrow G \quad \neg G}{\neg F} MT$$

- Derivación

1	$F \rightarrow G$	premisa
2	$\neg G$	premisa
3	F	supuesto
4	G	$\rightarrow$e $1,3$
5	$\perp$	$\neg$e $2,4$
6	$\neg F$	$\neg$i $2-4$

17

DN: Reglas derivadas: introducción de doble negación

- Regla de introducción de la doble negación:

$$\frac{F}{\neg\neg F} \neg\neg i$$

- Derivación:

1	F	premisa
2	$\neg F$	supuesto
3	$\perp$	$\neg$e $1,2$
4	$\neg\neg F$	$\neg$i $2-3$

18

DN: Reglas derivadas: reducción al absurdo (RAA)

- Regla de reducción al absurdo:

$$\dfrac{\boxed{\begin{array}{c}\neg F\\ \vdots \\ \bot\end{array}}}{F}\ \mathit{RAA}$$

- Derivación:

1	$\neg F \to \bot$	premisa
2	$\neg F$	supuesto
3	$\bot$	$\to$e $1,2$
4	$\neg\neg F$	$\neg$i $2-3$
5	F	$\neg$e $\neg$4

DN: Reglas derivadas: ley del tercio excluido (LEM)

- Ley del tercio excluido (LEM):

$$\dfrac{}{F \lor \neg F}\ \mathit{LEM}$$

- Derivación:

1	$\neg(F \lor \neg F)$	supuesto
2	F	supuesto
3	$F \lor \neg F$	$\lor$i 2
4	$\bot$	$\neg$e $1,3$
5	$\neg F$	$\neg$i $2-4$
6	$F \lor \neg F$	$\lor$i 5
7	$\bot$	$\neg$e $1,6$
8	$F \lor \neg F$	RAA $1-7$

DN: Reglas derivadas: ley del tercio excluido (LEM)

- Ejemplo: $p \rightarrow q \vdash \neg p \vee q$:

1	$p \rightarrow q$	premisa
2	$p \vee \neg p$	LEM
3	p	supuesto
4	q	$\rightarrow$e $1,3$
5	$\neg p \vee q$	$\vee$i 4
6	$\neg p$	supuesto
7	$\neg p \vee q$	$\vee$i 6
8	$\neg p \vee q$	$\vee$e $2, 3-5, 6-7$

DN: Reglas de deducción natural

- Reglas de deducción natural:

	Introducción	Eliminación
$\wedge$	$\dfrac{F \quad G}{F \wedge G}\wedge\text{i}$	$\dfrac{F \wedge G}{F}\wedge\text{e}_1$ $\quad \dfrac{F \wedge G}{G}\wedge\text{e}_2$
$\vee$	$\dfrac{F}{F \vee G}\vee\text{i}_1$ $\quad \dfrac{G}{F \vee G}\vee\text{i}_2$	$\dfrac{F \vee G \quad \boxed{\begin{array}{c}F\\ \vdots\\ H\end{array}} \quad \boxed{\begin{array}{c}G\\ \vdots\\ H\end{array}}}{H}\vee\text{e}$
$\rightarrow$	$\dfrac{\boxed{\begin{array}{c}F\\ \vdots\\ G\end{array}}}{F \rightarrow G}\rightarrow\text{i}$	$\dfrac{F \quad F \rightarrow G}{G}\rightarrow\text{e}$

DN: Reglas de deducción natural

- Reglas de deducción natural:

	Introducción	Eliminación
$\neg$	$\dfrac{\boxed{\begin{array}{c}F\\ \vdots\\ \bot\end{array}}}{\neg F}\neg i$	$\dfrac{F \quad \neg F}{\bot}\neg e$
$\bot$		$\dfrac{\bot}{F}\bot e$
$\neg\neg$		$\dfrac{\neg\neg F}{F}\neg\neg e$
$\leftrightarrow$	$\dfrac{F \rightarrow G \quad G \rightarrow F}{F \leftrightarrow G}\leftrightarrow i$	$\dfrac{F \leftrightarrow G}{F \rightarrow G}\leftrightarrow e_1 \quad \dfrac{F \leftrightarrow G}{G \rightarrow F}\leftrightarrow e_2$

- Adecuación y completitud del cálculo de deducción natural.

23

Bibliografía

1. C. Badesa, I. Jané y R. Jansana *Elementos de lógica formal.* (Ariel, 2000).
   Cap. 16: Cálculo deductivo.
2. R. Bornat *Using ItL Jape with X* (Department of Computer Science, QMW, 1998).
3. J.A. Díez *Iniciación a la Lógica*, (Ariel, 2002).
   Cap. 4: Cálculo deductivo. Deducibilidad.
4. M. Huth y M. Ryan *Logic in computer science: modelling and reasoning about systems.* (Cambridge University Press, 2000)
   Cap. 1: Propositional logic.
5. E. Paniagua, J.L. Sánchez y F. Martín *Lógica computacional* (Thomson, 2003)
   Cap. 3.6: El método de la deducción natural.

24

Capítulo 3

Formas normales

Lógica informática (2005–06)
Tema 3: Formas normales

José A. Alonso Jiménez

Grupo de Lógica Computacional
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

Equivalencia lógica: Fórmulas equivalentes.

- Def.: F y G son equivalentes si $v(F) = v(G)$ para toda valoración v.
  Representación: $F \equiv G$.
- Ejemplos de equivalencias notables:
  1. Idempotencia: $F \lor F \equiv F$; $F \land F \equiv F$.
  2. Conmutatividad: $F \lor G \equiv G \lor F$; $F \land G \equiv G \land F$.
  3. Asociatividad: $F \lor (G \lor H) \equiv (F \lor G) \lor H$; $F \land (G \land H) \equiv (F \land G) \land H$.
  4. Absorción: $F \land (F \lor G) \equiv F$; $F \lor (F \land G) \equiv F$.
  5. Distributividad: $F \land (G \lor H) \equiv (F \land G) \lor (F \land H)$;
     $F \lor (G \land H) \equiv (F \lor G) \land (F \lor H)$.
  6. Doble negación: $\neg\neg F \equiv F$.
  7. Leyes de De Morgan: $\neg(F \land G) \equiv \neg F \lor \neg G$; $\neg(F \lor G) \equiv \neg F \land \neg G$.
  8. Leyes de tautologías: Si F es una tautología, $F \land G \equiv G$; $F \lor G \equiv F$.
  9. Leyes de contradicciones: Si F es una contradicción $F \land G \equiv F$; $F \lor G \equiv G$.

Equivalencia lógica: propiedades

- Relación entre equivalencia y bicondicional:
  - $F \equiv G$ syss $\models F \leftrightarrow G$ syss $\vdash F \leftrightarrow G$.
- Propiedades básicas de la equivalencia lógica:
  - Reflexiva: $F \equiv F$.
  - Simétrica: Si $F \equiv G$, entonces $G \equiv F$.
  - Transitiva: Si $F \equiv G$ y $G \equiv H$, entonces $F \equiv H$.
- Principio de sustitución de fórmulas equivalentes:
  - Prop.: Si en la fórmula F se sustituye una de sus subfórmulas G por una fórmula G' lógicamente equivalente a G, entonces la fórmula obtenida, F', es lógicamente equivalente a F.
  - Ejemplo: $$\begin{aligned} F &= \neg(p \wedge q) \rightarrow \neg(p \wedge \neg\neg r) \\ G &= \neg(p \wedge q) \\ G' &= \neg p \vee \neg q \\ F' &= (\neg p \vee \neg q) \rightarrow \neg(p \wedge \neg\neg r) \end{aligned}$$

Formas normales: Forma normal conjuntiva

- Átomos y literales:
  - Def.: Un átomo es un variable proposicional (p.e. $p, q, \dots$).
  - Def.: Un literal es un átomo o su negación (p.e. $p, \neg p, q, \neg q, \dots$).
  - Notación: $L, L_1, L_2, \dots$ representarán literales.
- Forma normal conjuntiva:
  - Def.: Una fórmula está en forma normal conjuntiva (FNC) si es una conjunción de disyunciones de literales; es decir, es de la forma $(L_{1,1} \vee \dots \vee L_{1,n_1}) \wedge \dots \wedge (L_{m,1} \vee \dots \vee L_{m,n_m})$.
  - Ejemplos: $(\neg p \vee q) \wedge (\neg q \vee p)$ está en FNC.
    $(\neg p \vee q) \wedge (q \rightarrow p)$ no está en FNC.
  - Def.: Una fórmula G es una forma normal conjuntiva (FNC) de la fórmula F si G está en forma normal conjuntiva y es equivalente a F.
  - Ejemplo: Una FNC de $\neg(p \wedge (q \rightarrow r))$ es $(\neg p \vee q) \wedge (\neg p \vee \neg r)$.

Formas normales: Cálculo de forma normal conjuntiva

- Algoritmo de cálculo de forma normal conjuntiva:
  - Algoritmo: Aplicando a una fórmula F los siguientes pasos se obtiene una forma normal conjuntiva de F, $\text{FNC}(F)$:
    1. Eliminar los bicondicionales usando la equivalencia
       $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$ (1)
    2. Eliminar los condicionales usando la equivalencia
       $A \rightarrow B \equiv \neg A \vee B$ (2)
    3. Interiorizar las negaciones usando las equivalencias
       $\neg(A \wedge B) \equiv \neg A \vee \neg B$ (3)
       $\neg(A \vee B) \equiv \neg A \wedge \neg B$ (4)
       $\neg\neg A \equiv A$ (5)
    4. Interiorizar las disyunciones usando las equivalencias
       $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$ (6)
       $(A \wedge B) \vee C \equiv (A \vee C) \wedge (B \vee C)$ (7)

5

Formas normales: Cálculo de forma normal conjuntiva

- Ejemplo de cálculo de una FNC de $\neg(p \wedge (q \rightarrow r))$:

$$\begin{array}{lll} & \neg(p \wedge (q \rightarrow r)) & \\ \equiv & \neg(p \wedge (\neg q \vee r)) & \text{[por (2)]} \\ \equiv & \neg p \vee \neg(\neg q \vee r) & \text{[por (3)]} \\ \equiv & \neg p \vee (\neg\neg q \wedge \neg r) & \text{[por (4)]} \\ \equiv & \neg p \vee (q \wedge \neg r) & \text{[por (5)]} \\ \equiv & (\neg p \vee q) \wedge (\neg p \vee \neg r) & \text{[por (6)]} \end{array}$$

- Ejemplo de cálculo de una FNC de $(p \rightarrow q) \vee (q \rightarrow p)$:

$$\begin{array}{lll} & (p \rightarrow q) \vee (q \rightarrow p) & \\ \equiv & (\neg p \vee q) \vee (\neg q \vee p) & \text{[por (2)]} \\ \equiv & \neg p \vee q \vee \neg q \vee p & \end{array}$$

6

Formas normales: Cálculo de forma normal conjuntiva

- Ejemplo de cálculo de una FNC de $(p \leftrightarrow q) \rightarrow r$:

$$
\begin{array}{lll}
 & (p \leftrightarrow q) \rightarrow r & \\
\equiv & (p \rightarrow q) \wedge (q \rightarrow p) \rightarrow r & [(1)] \\
\equiv & \neg((p \rightarrow q) \wedge (q \rightarrow p)) \vee r & [(2)] \\
\equiv & \neg((\neg p \vee q) \wedge (\neg q \vee p)) \vee r & [(2)] \\
\equiv & (\neg(\neg p \vee q) \vee \neg(\neg q \vee p)) \vee r & [(3)] \\
\equiv & ((\neg\neg p \wedge \neg q) \vee (\neg\neg q \wedge \neg p)) \vee r & [(4)] \\
\equiv & ((p \wedge \neg q) \vee (q \wedge \neg p)) \vee r & [(5)] \\
\equiv & (((p \wedge \neg q) \vee q) \wedge ((p \wedge \neg q) \vee \neg p)) \vee r & [(6)] \\
\equiv & (((p \vee q) \wedge (\neg q \vee q)) \wedge ((p \vee \neg p) \wedge (\neg q \vee \neg p))) \vee r & [(7)] \\
\equiv & (((p \vee q) \wedge (\neg q \vee q)) \vee r) \wedge (((p \vee \neg p) \wedge (\neg q \vee \neg p)) \vee r) & [(7)] \\
\equiv & (((p \vee q) \vee r) \wedge ((\neg q \vee q) \vee r)) \wedge (((p \vee \neg p) \vee r) \wedge ((\neg q \vee \neg p) \vee r)) & [(7)] \\
\equiv & (p \vee q \vee r) \wedge (\neg q \vee q \vee r) \wedge (p \vee \neg p \vee r) \wedge (\neg q \vee \neg p \vee r) & \\
\equiv & (p \vee q \vee r) \wedge (\neg q \vee \neg p \vee r) &
\end{array}
$$

Formas normales: Forma normal disyuntiva

- Forma normal disyuntiva:
  - Def.: Una fórmula está en forma normal disyuntiva (FND) si es una disyunción de conjunciones de literales; es decir, es de la forma $(L_{1,1} \wedge \ldots \wedge L_{1,n_1}) \vee \ldots \vee (L_{m,1} \wedge \ldots \wedge L_{m,n_m})$.
  - Ejemplos: $(\neg p \wedge q) \vee (\neg q \wedge p)$ está en FND.
    $(\neg p \wedge q) \vee (q \rightarrow p)$ no está en FND.
  - Def.: Una fórmula G es una forma normal disyuntiva (FND) de la fórmula F si G está en forma normal disyuntiva y es equivalente a F.
  - Ejemplo: Una FND de $\neg(p \wedge (q \rightarrow r))$ es $\neg p \vee (q \wedge \neg r)$.

Formas normales: Cálculo de forma normal disyuntiva

- Algoritmo de cálculo de forma normal disyuntiva:
  - Algoritmo: Aplicando a una fórmula F los siguientes pasos se obtiene una forma normal disyuntiva de F, FND(F):
    1. Eliminar los bicondicionales usando la equivalencia
       $A \leftrightarrow B \equiv (A \rightarrow B) \land (B \rightarrow A)$ (1)
    2. Eliminar los condicionales usando la equivalencia
       $A \rightarrow B \equiv \neg A \lor B$ (2)
    3. Interiorizar las negaciones usando las equivalencias
       $\neg(A \land B) \equiv \neg A \lor \neg B$ (3)
       $\neg(A \lor B) \equiv \neg A \land \neg B$ (4)
       $\neg\neg A \equiv A$ (5)
    4. Interiorizar las conjunciones usando las equivalencias
       $A \land (B \lor C) \equiv (A \land B) \lor (A \land C)$ (6)
       $(A \lor B) \land C \equiv (A \land C) \lor (B \land C)$ (7)

Formas normales: Cálculo de forma normal disyuntiva

- Ejemplo de cálculo de una FND de $\neg(p \land (q \rightarrow r))$:

$$
\begin{array}{lll}
 & \neg(p \land (q \rightarrow r)) & \\
\equiv & \neg(p \land (\neg q \lor r)) & \text{[por (2)]} \\
\equiv & \neg p \lor \neg(\neg q \lor r) & \text{[por (3)]} \\
\equiv & \neg p \lor (\neg\neg q \land \neg r) & \text{[por (4)]} \\
\equiv & \neg p \lor (q \land \neg r) & \text{[por (5)]}
\end{array}
$$

- Ejemplo de cálculo de una FND de $\neg(\neg p \lor \neg q \rightarrow \neg(p \land q))$:

$$
\begin{array}{lll}
 & \neg(\neg p \lor \neg q \rightarrow \neg(p \land q)) & \\
\equiv & \neg(\neg(\neg p \lor \neg q) \lor \neg(p \land q)) & \text{[por (2)]} \\
\equiv & \neg\neg(\neg p \lor \neg q) \land \neg\neg(p \land q) & \text{[por (4)]} \\
\equiv & (\neg p \lor \neg q) \land (p \land q) & \text{[por (5)]} \\
\equiv & (\neg p \land (p \land q)) \lor (\neg q \land (p \land q)) & \text{[por (7)]} \\
\equiv & (\neg p \land p \land q) \lor (\neg q \land p \land q) &
\end{array}
$$

Decisión de validez mediante FNC

- Literales complementarios:
  - El complementario de un literal L es $L^c = \begin{cases} \neg p & \text{si } L = p; \\ p & \text{si } L = \neg p. \end{cases}$
- Propiedades de reducción de tautologías:
  - $F_1 \land \ldots \land F_n$ es una tautología syss $F_1, \ldots, F_n$ lo son.
  - $L_1 \lor \ldots \lor L_n$ es una tautología syss $\{L_1, \ldots, L_n\}$ contiene algún par de literales complementarios (i.e. existen i, j tales que $L_i = L_j^c$).
- Algoritmo de decisión de tautologías mediante FNC
  - Entrada: Una fórmula F.
  - Procedimiento:
    1. Calcular una FNC de F.
    2. Decidir si cada una de las conjunciones de la FNC tiene algún par de literales complementarios.

Decisión de validez mediante FNC

- Ejemplos de decisión de tautologías mediante FNC
  - $\neg(p \land (q \to r))$ no es tautología:
    FNC$(\neg(p \land (q \to r))) = (\neg p \lor q) \land (\neg p \lor \neg r)$
    Contramodelos de $\neg(p \land (q \to r))$:
    v_1 tal que $v_1(p) = 1$ y $v_1(q) = 0$
    v_2 tal que $v_2(p) = 1$ y $v_2(r) = 1$
  - $(p \to q) \lor (q \to p)$ es tautología:
    FNC$((p \to q) \lor (q \to p)) = \neg p \lor q \lor \neg q \lor p$
  - $(p \leftrightarrow q) \to r$ no es tautología:
    FNC$((p \leftrightarrow q) \to r) = (p \lor q \lor r) \land (\neg q \lor \neg p \lor r)$
    Contramodelos de $(p \leftrightarrow q) \to r$:
    v_1 tal que $v_1(p) = 0, v_1(q) = 0$ y $v_1(r) = 0$
    v_2 tal que $v_2(p) = 1, v_2(q) = 1$ y $v_2(r) = 0$

Decisión de satisfacibilidad mediante FND

- Propiedades de reducción de satisfacibilidad:
  - $F_1 \vee \ldots \vee F_n$ es satisfacible syss alguna de las fórmulas $F_1, \ldots, F_n$ lo es.
  - $L_1 \wedge \ldots \wedge L_n$ es satisfacible syss $\{L_1, \ldots, L_n\}$ no contiene ningún par de literales complementarios.
- Algoritmo de decisión de satisfacibilidad mediante FND:
  - Entrada: Una fórmula F.
  - Procedimiento:
    1. Calcular una FND de F.
    2. Decidir si alguna de las disyunciones de la FND no tiene un par de literales complementarios.

Decisión de satisfacibilidad mediante FND

- Ejemplos de decisión de satisfacibilidad mediante FND:
  - $\neg(p \wedge (q \rightarrow r))$ es satisfacible:
    FND$(\neg(p \wedge (q \rightarrow r))) = \neg p \vee (q \wedge \neg r)$
    Modelos de $\neg(p \wedge (q \rightarrow r))$:
    v_1 tal que $v_1(p) = 0$
    v_2 tal que $v_2(q) = 1$ y $v_2(r) = 0$
  - $\neg(\neg p \vee \neg q \rightarrow \neg(p \wedge q))$ es insatisfacible:
    FND$(\neg(\neg p \vee \neg q \rightarrow \neg(p \wedge q))) = (\neg p \wedge p \wedge q) \vee (\neg q \wedge p \wedge q)$

Bibliografía

1. C. Badesa, I. Jané y R. Jansana *Elementos de lógica formal.* (Ariel, 2000)
   Cap. 8 (Equivalencia lógica) y 10 (Formas normales).
2. M. Ben–Ari, *Mathematical logic for computer science (2nd ed.).* (Springer, 2001)
   Cap. 2 (Propositional calculus: formulas, models, tableaux).
3. J.A. Díez *Iniciación a la Lógica*, (Ariel, 2002)
   Cap. 3 (Semántica formal. Consecuencia lógica).
4. M. Huth y M. Ryan *Logic in computer science: modelling and reasoning about systems.* (Cambridge University Press, 2000)
   Cap. 1 (Propositional logic).
5. E. Paniagua, J.L. Sánchez y F. Martín *Lógica computacional* (Thomson, 2003)
   Cap. 4.4 (Formas normales).

Capítulo 4

Tableros semánticos

Lógica informática (2005–06)
Tema 4: Tableros semánticos

José A. Alonso Jiménez

Grupo de Lógica Computacional
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

1

Demostración de fórmula tautológica

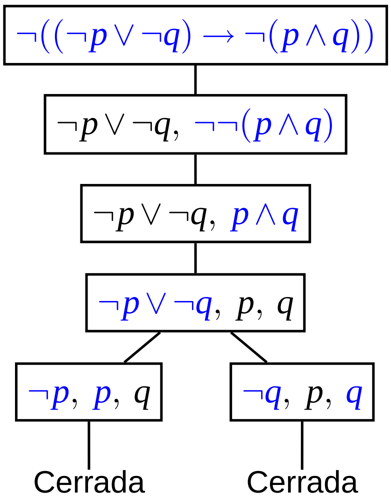
- Demostración de $\models \neg p \lor \neg q \to \neg(p \land q)$.

$\neg p \lor \neg q \to \neg(p \land q)$ es una tautología
syss $\{\neg(\neg p \lor \neg q \to \neg(p \land q))\}$ es inconsistente
syss $\{\neg p \lor \neg q, \neg\neg(p \land q)\}$ es inconsistente
syss $\{\neg p \lor \neg q, p \land q\}$ es inconsistente
syss $\{p, q, \neg p \lor \neg q\}$ es inconsistente
syss $\{p, q, \neg p\}$ es inconsistente y
$\{p, q, \neg q\}$ es inconsistente

2

Demostración por tableros semánticos

- Tablero semántico cerrado:



3

Refutación de fórmula no tautológica

- Refutación $\not\models \neg p \lor \neg q \to \neg(p \land r)$:

  $\neg p \lor \neg q \to \neg(p \land r)$ es una tautología
  syss $\{\neg(\neg p \lor \neg q \to \neg(p \land r))\}$ es inconsistente
  syss $\{\neg p \lor \neg q, \neg\neg(p \land r)\}$ es inconsistente
  syss $\{\neg p \lor \neg q, p \land r\}$ es inconsistente
  syss $\{p, r, \neg p \lor \neg q\}$ es inconsistente
  syss $\{p, r, \neg p\}$ es inconsistente y
  $\{p, r, \neg q\}$ es inconsistente

- Contramodelos de $\neg p \lor \neg q \to \neg(p \land r)$:

  Las valoraciones v tales que $v(p) = 1, v(q) = 0$ y $v(r) = 1$.

- Una forma normal disyuntiva de $\neg(\neg p \lor \neg q \to \neg(p \land r))$:

  $p \land r \land \neg q$

4

Refutación por tableros semánticos

- Tablero semántico:

$$\neg((\neg p \lor \neg q) \to \neg(p \land r))$$
$$\neg p \lor \neg q,\ \neg\neg(p \land r)$$
$$\neg p \lor \neg q,\ p \land r$$
$$\neg p \lor \neg q,\ p,\ r$$
$$\neg p,\ p,\ r \qquad\qquad \neg q,\ p,\ r$$

Cerrada Abierta

Notación uniforme: Literales y dobles negaciones

- Literales
  - Un literal es un átomo o la negación de un átomo (p.e. $p, \neg p, q, \neg q, \dots$).
- Dobles negaciones
  - F es una doble negación si es de la forma $\neg\neg G$.
  - Ley de doble negación: Si F es $\neg\neg G$, entonces $F \equiv G$.

Notación uniforme: fórmulas alfa y beta

- Las fórmulas alfa, junto con sus componentes, son

F	F_1	F_2
$A_1 \wedge A_2$	A_1	A_2
$\neg(A_1 \rightarrow A_2)$	A_1	$\neg A_2$
$\neg(A_1 \vee A_2)$	$\neg A_1$	$\neg A_2$
$A_1 \leftrightarrow A_2$	$A_1 \rightarrow A_2$	$A_2 \rightarrow A_1$

- Si F es una fórmula alfa con componentes F_1 y F_2, entonces $F \equiv F_1 \wedge F_2$.
- Las fórmulas beta, junto con sus componentes, son

F	F_1	F_2
$B_1 \vee B_2$	B_1	B_2
$B_1 \rightarrow B_2$	$\neg B_1$	B_2
$\neg(B_1 \wedge B_2)$	$\neg B_1$	$\neg B_2$
$\neg(B_1 \leftrightarrow B_2)$	$\neg(B_1 \rightarrow B_2)$	$\neg(B_2 \rightarrow B_1)$

- Si F es una fórmula beta y con componentes F_1 y F_2, entonces $F \equiv F_1 \vee F_2$.

7

Tablero del conjunto de fórmulas S

- El árbol cuyo único nodo tiene como etiqueta S es un tablero de S.
- Sea $\mathscr{T}$ un tablero de S y S_1 la etiqueta de una hoja de $\mathscr{T}$.
  1. Si S_1 es cerrado (es decir, que contiene una fórmula y su negación), entonces el árbol obtenido añadiendo como hijo de S_1 el nodo etiquetado con `cerrado` es un tablero de S.
  2. Si S_1 es abierto (es decir, es un conjunto de literales que no es cerrado), entonces el árbol obtenido añadiendo como hijo de S_1 el nodo etiquetado con `abierto` es un tablero de S.
  3. Si S_1 contiene una doble negación $\neg\neg F$, entonces el árbol obtenido añadiendo como hijo de S_1 el nodo etiquetado con $(S_1 \setminus \{\neg\neg F\}) \cup \{F\}$ es un tablero de S.
  4. Si S_1 contiene una fórmula alfa F de componentes F_1 y F_2, entonces el árbol obtenido añadiendo como hijo de S_1 el nodo etiquetado con $(S_1 \setminus \{F\}) \cup \{F_1, F_2\}$ es un tablero de S.
  5. Si S_1 contiene una fórmula beta F de componentes F_1 y F_2, entonces el árbol obtenido añadiendo como hijos de S_1 los nodos etiquetados con $(S_1 \setminus \{F\}) \cup \{F_1\}$ y $(S_1 \setminus \{F\}) \cup \{F_2\}$ es un tablero de S.

8

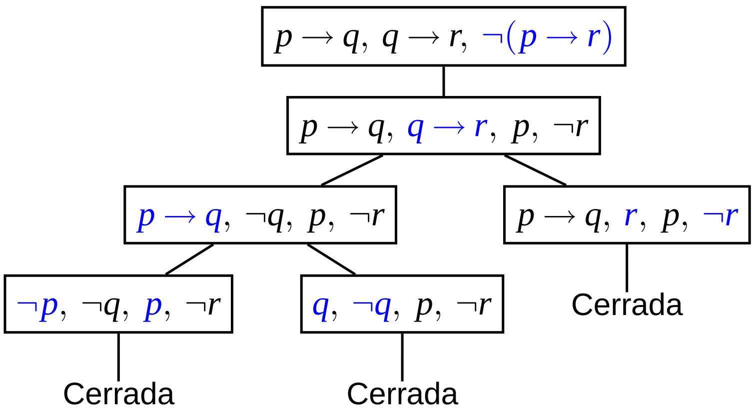
Teorema por tableros

- Def.: Un tablero completo de S es un tablero de S tal que todas sus hojas son abiertas o cerradas.
- Def.: Un tablero es cerrado si todas sus hojas son cerradas.
- Def.: Una fórmula F es un teorema (mediante tableros semánticos) si tiene una prueba mediante tableros; es decir, si $\{\neg F\}$ tiene un tablero completo cerrado. Se representa por $\vdash_{Tab} F$.
- Ejemplos: $\vdash_{Tab} \neg p \vee \neg q \rightarrow \neg(p \wedge q)$
  $\nvdash_{Tab} \neg p \vee \neg q \rightarrow \neg(p \wedge r)$
- Teor.: El cálculo de tableros semánticos es adecuado y completo; es decir,
  Adecuado: $\vdash_{Tab} F \quad \Longrightarrow \quad \models F$
  Completo: $\models F \quad \Longrightarrow \quad \vdash_{Tab} F$
- Si los nodos abiertos de un tablero completo de F son
  $\{L_{1,1},\dots,L_{1,n_1}\},\dots,\{L_{m,1},\dots,L_{m,n_m}\}$,
  entonces una forma normal disyuntiva de F es
  $(L_{1,1} \wedge \dots \wedge L_{1,n_1}) \vee \dots \vee (L_{m,1} \wedge \dots \wedge L_{m,n_m})$.

9

Deducción por tableros

- Def.: La fórmula F es deducible (mediante tableros semánticos) a partir del conjunto de fórmulas S si existe una prueba mediante tableros de F a partir de S; es decir, existe un tablero completo cerrado de $S \cup \{\neg F\}$.
  Se representa por $S \vdash_{Tab} F$.
- Ejemplo: $\{p \rightarrow q, q \rightarrow r\} \vdash_{Tab} p \rightarrow r$

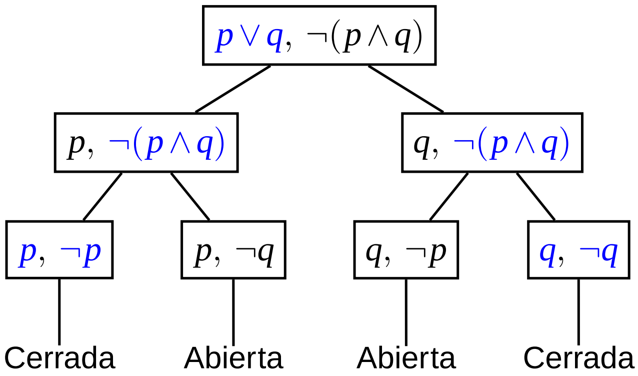


- Teor.: $S \vdash_{Tab} F$ syss $S \models F$.

10

Deducción por tableros

- Ejemplo: $\{p \lor q\} \nvdash_{Tab} p \land q$



- Contramodelos de $\{p \lor q\} \nvdash_{Tab} p \land q$

  las valoraciones v_1 tales que $v_1(p) = 1$ y $v_1(q) = 0$

  las valoraciones v_2 tales que $v_2(p) = 0$ y $v_2(q) = 1$

Bibliografía

1. Ben–Ari, M. *Mathematical Logic for Computer Science (2nd ed.)* (Springer, 2001)

   Cap. 2: Propositional calculus: formulas, models, tableaux

2. Fitting, M. *First-Order Logic and Automated Theorem Proving (2nd ed.)* (Springer, 1995)

   Cap. 3: Semantic tableaux and resolution

3. Hortalá, M.T.; Leach, J. y Rogríguez, M. *Matemática discreta y lógica matemática* (Ed. Complutense, 1998)

   Cap. 7.9: Tableaux semánticos para la lógica de proposiciones

4. Nerode, A. y Shore, R.A. *Logic for Applications* (Springer, 1997)

   Cap. 1.4: Tableau proofs in propositional calculus

5. E. Paniagua, J.L. Sánchez y F. Martín *Lógica computacional* (Thomson, 2003)

   Cap. 4.3: Métodos de las tablas semánticas

Capítulo 5

Resolución proposicional

Lógica informática (2005–06)
Tema 5: Resolución proposicional

José A. Alonso Jiménez

Grupo de Lógica Computacional
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

1

Lógica clausal: sintaxis

- Un átomo es una variable proposicional.
  Variables sobre átomos: $p, q, r, \dots, p_1, p_2, \dots$.
- Un literal es un átomo (p) o la negación de un átomo ($\neg p$).
  Variables sobre literales: $L, L_1, L_2, \dots$.
- Una cláusula es un conjunto finito de literales.
  Variables sobre cláusulas: $C, C_1, C_2, \dots$.
- La cláusula vacía es el conjunto vacío de literales.
  La cláusula vacía se representa por $\square$.
- Conjuntos finitos de cláusulas.
  Variables sobre conjuntos finitos de cláusulas: $S, S_1, S_2, \dots$.

2

Lógica clausal: semántica

- Def.: Una valoración de verdad es una aplicación $v : \mathrm{VP} \to \mathbb{B}$.
- Def.: El valor de un literal positivo p en una valoración v es $v(p)$.
- Def.: El valor de un literal negativo $\neg p$ en una valoración v es

$$v(\neg p) = \begin{cases} 1, & \text{si } v(p) = 0; \\ 0, & \text{si } v(p) = 1. \end{cases}$$

- Def.: El valor de una cláusula C en una valoración v es

$$v(C) = \begin{cases} 1, & \text{si existe un } L \in C \text{ tal que } v(L) = 1; \\ 0, & \text{en caso contrario.} \end{cases}$$

- Def.: El valor de un conjunto de cláusulas S en una valoración v es

$$v(S) = \begin{cases} 1, & \text{si para toda } C \in S, v(C) = 1 \\ 0, & \text{en caso contrario.} \end{cases}$$

- Prop.: En cualquier valoración v, $v(\square) = 0$.

Cláusulas y fórmulas

- Equivalencias entre cláusulas y fórmulas
  - Def.: Una cláusula C y una fórmula F son equivalentes si $v(C) = v(F)$ para cualquier valoración v.
  - Def.: Un conjunto de cláusulas S y una fórmula F son equivalentes si $v(S) = v(F)$ para cualquier valoración v.
  - Def.: Un conjunto de cláusulas S y un conjunto de fórmulas $\{F_1, \dots, F_n\}$ son equivalentes si, para cualquier valoración v, $v(S) = 1$ syss v es un modelo de $\{F_1, \dots, F_n\}$.
- De cláusulas a fórmulas
  - Prop.: La cláusula $\{L_1, L_2, \dots, L_n\}$ es equivalente a la fórmula $L_1 \vee L_2 \vee \dots \vee L_n$.
  - Prop.: El conjunto de cláusulas $\{\{L_{1,1}, \dots, L_{1,n_1}\}, \dots, \{L_{m,1}, \dots, L_{m,n_m}\}\}$ es equivalente a la fórmula $(L_{1,1} \vee \dots \vee L_{1,n_1}) \wedge \dots \wedge (L_{m,1} \vee \dots \vee L_{m,n_m})$.

De fórmulas a cláusulas (forma clausal)

- Def.: Una forma clausal de una fórmula F es un conjunto de cláusulas equivalente a F.
- Prop.: Si $(L_{1,1} \vee \ldots \vee L_{1,n_1}) \wedge \ldots \wedge (L_{m,1} \vee \ldots \vee L_{m,n_m})$ es una forma normal conjuntiva de la fórmula F. Entonces, una forma clausal de F es $\{\{L_{1,1}, \ldots, L_{1,n_1}\}, \ldots, \{L_{m,1}, \ldots, L_{m,n_m}\}\}$.
- Ejemplos:
  - Una forma clausal de $\neg(p \wedge (q \rightarrow r))$ es $\{\{\neg p, q\}, \{\neg p, \neg r\}\}$.
  - Una forma clausal de $p \rightarrow q$ es $\{\{\neg p, q\}\}$.
  - La cláusula $\{\{\neg p, q\}, \{r\}\}$ es una forma clausal de las fórmulas $(p \rightarrow q) \wedge r$ y $\neg\neg r \wedge (\neg q \rightarrow \neg p)$.
- Def.: Una forma clausal de un conjunto de fórmulas S es un conjunto de cláusulas equivalente a S.
- Prop.: Si $S_1, \ldots, S_n$ son formas clausales de $F_1, \ldots, F_n$, entonces $S_1 \cup \ldots \cup S_n$ es una forma clausal de $\{F_1, \ldots, F_n\}$.

5

Modelos, consistencia y consecuencia

- Def.: Una valoración v es modelo de un conjunto de cláusulas S si $v(S) = 1$.
- Ej.: La valoración v tal que $v(p) = v(q) = 1$ es un modelo de $\{\{\neg p, q\}, \{p, \neg q\}\}$.
- Def.: Un conjunto de cláusulas es consistente si tiene modelos e inconsistente, en caso contrario.
- Ejemplos:
  - $\{\{\neg p, q\}, \{p, \neg q\}\}$ es consistente.
  - $\{\{\neg p, q\}, \{p, \neg q\}, \{p, q\}, \{\neg p, \neg q\}\}$ es inconsistente.
- Prop.: Si $\square \in S$, entonces S es inconsistente.
- Def.: $S \models C$ si para todo modelo v de S, $v(C) = 1$.

6

Reducción de consecuencia a inconsistencia de cláusulas

- Prop: Sean $S_1, \dots, S_n$ formas clausales de las fórmulas $F_1, \dots, F_n$.
  - $\{F_1, \dots, F_n\}$ es consistente syss $S_1 \cup \dots \cup S_n$ es consistente.
  - Si S es una forma clausal de $\neg G$, entonces son equivalentes
    1. $\{F_1, \dots, F_n\} \models G$.
    2. $\{F_1, \dots, F_n \neg G\}$ es inconsistente.
    3. $S_1 \cup \dots \cup S_n \cup S$ es inconsistente.
- Ejemplo: $\{p \rightarrow q, q \rightarrow r\} \models p \rightarrow r$ syss
  $\{\{\neg p, q\}, \{\neg q, r\}, \{p\}, \{\neg r\}\}$ es inconsistente.

7

Regla de resolución

- Reglas habituales:

Modus Ponens: $\dfrac{p \rightarrow q, \quad p}{q}$ $\qquad \dfrac{\{\neg p, q\}, \quad \{p\}}{\{q\}}$

Modus Tollens: $\dfrac{p \rightarrow q, \quad \neg q}{\neg p}$ $\qquad \dfrac{\{\neg p, q\}, \quad \{\neg q\}}{\{\neg p\}}$

Encadenamiento: $\dfrac{p \rightarrow q, \quad q \rightarrow r}{p \rightarrow r}$ $\qquad \dfrac{\{\neg p, q\}, \quad \{\neg q, r\}}{\{\neg p, r\}}$

- Regla de resolución proposicional:

$$\frac{\{p_1, \dots, r, \dots, p_m\}, \quad \{q_1, \dots, \neg r, \dots, q_n\}}{\{p_1, \dots, p_m, q_1, , \dots, q_n\}}$$

8

Regla de resolución

- Def.: Sean C_1 una cláusula, L un literal de C_1 y C_2 una cláusula que contiene el complementario de L. La resolvente de C_1 y C_2 respecto de L es
$$\text{Res}_L(C_1,C_2) = (C_1 \setminus \{L\}) \cup (C_2 \setminus \{L^c\})$$
- Ejemplos:
$$\begin{aligned}
&\text{Res}_q(\{p,q\},\{\neg q,r\}) &&= \{p,r\}\\
&\text{Res}_q(\{q,\neg p\},\{p,\neg q\}) &&= \{p,\neg p\}\\
&\text{Res}_p(\{q,\neg p\},\{p,\neg q\}) &&= \{q,\neg q\}\\
&\text{Res}_p(\{q,\neg p\},\{q,p\}) &&= \{q\}\\
&\text{Res}_p(\{p\},\{\neg p\}) &&= \square
\end{aligned}$$
- Def.: $\text{Res}(C_1,C_2)$ es el conjunto de las resolventes entre C_1 y C_2
- Ejemplos:
$$\begin{aligned}
&\text{Res}(\{\neg p,q\},\{p,\neg q\}) &&= \{\{p,\neg p\},\{q,\neg q\}\}\\
&\text{Res}(\{\neg p,q\},\{p,q\}) &&= \{\{q\}\}\\
&\text{Res}(\{\neg p,q\},\{q,r\}) &&= \emptyset
\end{aligned}$$
- Nota: $\square \not\in \text{Res}(\{p,q\},\{\neg p,\neg q\})$

9

Ejemplo de refutación por resolución

- Refutación de $\{\{p,q\},\{\neg p,q\},\{p,\neg q\},\{\neg p,\neg q\}\}$:

1	$\{p,q\}$	Hipótesis
2	$\{\neg p,q\}$	Hipótesis
3	$\{p,\neg q\}$	Hipótesis
4	$\{\neg p,\neg q\}$	Hipótesis
5	$\{q\}$	Resolvente de 1 y 2
6	$\{\neg q\}$	Resolvente de 3 y 4
7	$\square$	Resolvente de 5 y 6

10

Demostraciones por resolución

- Sea S un conjunto de cláusulas.
- La sucesión $(C_1,\dots,C_n)$ es una demostración por resolución de la cláusula C a partir de S si $C = C_n$ y para todo $i \in \{1,...,n\}$ se verifica una de las siguientes condiciones:
  - $C_i \in S$;
  - existen $j,k < i$ tales que C_i es una resolvente de C_j y C_k
- La cláusula C es demostrable por resolución a partir de S si existe una demostración por resolución de C a partir de S.
- Una refutación por resolución de S es una demostración por resolución de la cláusula vacía a partir de S.
- Se dice que S es refutable por resolución si existe una refutación por resolución a partir de S.

Demostraciones por resolución

- Def.: Sean $S_1,\dots,S_n$ formas clausales de las fórmulas $F_1,\dots,F_n$ y
  S una forma clausal de $\neg F$
  Una demostración por resolución de F a partir de $\{F_1,\dots,F_n\}$ es una refutación por resolución de $S_1 \cup \dots \cup S_n \cup S$.
- Def.: La fórmula F es demostrable por resolución a partir de $\{F_1,\dots,F_n\}$ si existe una demostración por resolución de F a partir de $\{F_1,\dots,F_n\}$.
  Se representa por $\{F_1,\dots,F_n\} \vdash_{Res} F$.
- Ejemplo: Demostración por resolución de $p \wedge q$ a partir de $\{p \vee q, p \leftrightarrow q\}$

1	$\{p,q\}$	Hipótesis
2	$\{\neg p,q\}$	Hipótesis
3	$\{p,\neg q\}$	Hipótesis
4	$\{\neg p,\neg q\}$	Hipótesis
5	$\{q\}$	Resolvente de 1 y 2
6	$\{\neg q\}$	Resolvente de 3 y 4
7	$\square$	Resolvente de 5 y 6

Adecuación y completitud de la resolución

- Si C es una resolvente de C_1 y C_2, entonces $\{C_1, C_2\} \models C$.
- Si $\square \in S$, entonces S es inconsistente.
- Si el conjunto de cláusulas S es refutable, entonces S es inconsistente.
- Teor.: El cálculo de resolución es adecuado y completo; es decir,

Adecuado:	$S \vdash_{Res} F$	$\Longrightarrow$	$S \models F$
Completo:	$S \models F$	$\Longrightarrow$	$S \vdash_{Res} F$

13

Argumentación y resolución

- Problema de los animales: Se sabe que
  1. Los animales con pelo y los que dan leche son mamíferos.
  2. Los mamíferos que tienen pezuñas o que rumian son ungulados.
  3. Los ungulados de cuello largo son jirafas.
  4. Los ungulados con rayas negras son cebras.

  Se observa un animal que tiene pelos, pezuñas y rayas negras. Por consiguiente, se concluye que el animal es una cebra.
- Formalización:

$$
\begin{array}{l}
\{ \quad \text{tiene\_pelos} \lor \text{da\_leche} \to \text{es\_mamífero}, \\
\quad\;\; \text{es\_mamífero} \land (\text{tiene\_pezuñas} \lor \text{rumia}) \to \text{es\_ungulado}, \\
\quad\;\; \text{es\_ungulado} \land \text{tiene\_cuello\_largo} \to \text{es\_jirafa}, \\
\quad\;\; \text{es\_ungulado} \land \text{tiene\_rayas\_negras} \to \text{es\_cebra}, \\
\quad\;\; \text{tiene\_pelos} \land \text{tiene\_pezuñas} \land \text{tiene\_rayas\_negras} \; \} \\
\vdash_{Res} \text{es\_cebra}
\end{array}
$$

14

Argumentación y resolución

1	{¬ tiene_pelos, es_mamífero}	Hipótesis
2	{¬ da_leche, es_mamífero}	Hipótesis
3	{¬es_mamífero, ¬tiene_pezuñas, es_ungulado}	Hipótesis
4	{¬es_mamífero, ¬rumia, es_ungulado}	Hipótesis
5	{¬es_ungulado, ¬tiene_cuello_largo, es_jirafa}	Hipótesis
6	{¬es_ungulado, ¬tiene_rayas_negras, es_cebra}	Hipótesis
7	{tiene_pelos}	Hipótesis
8	{tiene_pezuñas}	Hipótesis
9	{tiene_rayas_negras}	Hipótesis
10	{¬es_cebra}	Hipótesis
11	{es_mamífero}	Resolvente de 1 y 7
12	{¬tiene_pezuñas, es_ungulado}	Resolvente de 11 y 3
13	{es_ungulado}	Resolvente de 12 y 8
14	{¬tiene_rayas_negras, es_cebra}	Resolvente de 13 y 6
15	{es_cebra}	Resolvente de 14 y 9
16	$\square$	Resolvente de 15 y 10

Bibliografía

1. M. Ben–Ari, *Mathematical logic for computer science (2nd ed.).* (Springer, 2001).
   Cap. 4: Propositional calculus: resolution and BDDs.
2. C.–L. Chang y R.C.–T. Lee *Symbolic Logic and Mechanical Theorem Proving* (Academic Press, 1973).
   Cap. 5.2: The resolution principle for the proposicional logic.
3. N.J. Nilsson *Inteligencia artificial (Una nueva síntesis)* (McGraw–Hill, 2001).
   Cap. 14: La resolución en el cálculo proposicional.
4. E. Paniagua, J.L. Sánchez y F. Martín *Lógica computacional* (Thomson, 2003).
   Cap. 5.7: El principio de resolución en lógica proposicional.
5. U. Schöning *Logic for Computer Scientists* (Birkäuser, 1989).
   Cap. 1.5: Resolution.

Capítulo 6

Sintaxis y semántica de la lógica de primer orden

Lógica informática (2005–06)

Tema 6: Sintaxis y semántica de la lógica de primer orden

José A. Alonso Jiménez

Grupo de Lógica Computacional
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

1

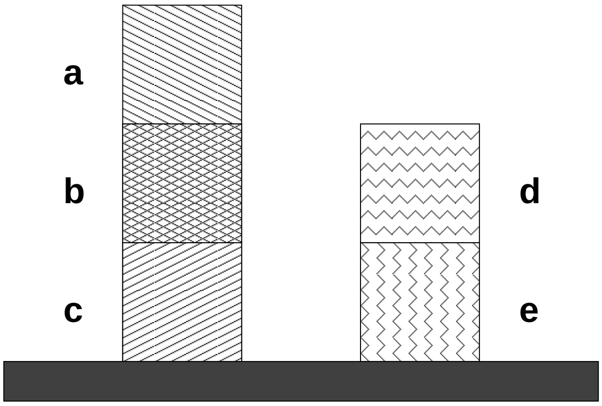
Limitación expresiva de la lógica proposicional

- Ejemplo 1: *Si Sevilla es vecina de Cádiz, entonces Cádiz es vecina de Sevilla. Sevilla es vecina de Cádiz. Por tanto, Cádiz es vecina de Sevilla*
  - Representación en lógica proposicional:
    $\{SvC \to CvS, \quad SvC\} \models CvS$
- Ejemplo 2: *Si una ciudad es vecina de otra, entonces la segunda es vecina de la primera. Sevilla es vecina de Cádiz. Por tanto, Cádiz es vecina de Sevilla*
  - Representación en lógica proposicional: Imposible
  - Representación en lógica de primer orden:
    $\{(\forall x)(\forall y)[vecina(x,y) \to vecina(y,x)], \quad vecina(Sevilla,Cadiz)\}$
    $\models vecina(Cadiz,Sevilla)$

2

Potencia expresiva de la lógica de primer orden

Mundo de los bloques



- Simbolización:
  - $\text{sobre}(x,y)$ se verifica si el bloque x está colocado sobre el bloque y
  - $\text{sobre\_mesa}(x)$ se verifica si el bloque x está sobre la mesa
- Situación del ejemplo:

  $\text{sobre}(a,b), \text{sobre}(b,c), \text{sobre\_mesa}(c), \text{sobre}(d,e), \text{sobre\_mesa}(e)$

Potencia expresiva de la lógica de primer orden

- Definiciones:
  - $\text{bajo}(x,y)$ se verifica si el bloque x está debajo del bloque y

    $(\forall x)(\forall y)[\text{bajo}(x,y) \leftrightarrow \text{sobre}(y,x)]$
  - $\text{encima}(x,y)$ se verifica si el bloque x está encima del bloque y pudiendo haber otros bloques entre ellos

    $(\forall x)(\forall y)[\text{encima}(x,y) \leftrightarrow \text{sobre}(x,y) \vee (\exists z)[\text{sobre}(x,z) \wedge \text{encima}(z,y)]]$
  - $\text{libre}(x)$ se verifica si el bloque x no tiene bloques encima

    $(\forall x)[\text{libre}(x) \leftrightarrow \neg(\exists y)\,\text{sobre}(y,x)]$
  - $\text{pila}(x,y,z)$ se verifica si el bloque x está sobre el y, el y sobre el z y el z sobre la mesa

    $(\forall x)(\forall y)(\forall z)[\text{pila}(x,y,z) \leftrightarrow \text{sobre}(x,y) \wedge \text{sobre}(y,z) \wedge \text{sobre\_mesa}(z)]$
- Propiedades:
  - Si z,y,z es una pila entonces y no está libre

    $(\forall x)(\forall y)(\forall z)[\text{pila}(x,y,z) \rightarrow \neg\,\text{libre}(y)]$

Potencia expresiva de la lógica de primer orden

Representación del mundo de los bloques con funciones e igualdad:

- Simbolización:
  - $\text{es\_bloque}(x)$ se verifica si x es un bloque.
  - $\text{superior}(x)$ es el bloque que está sobre el bloque x.
- Situación del ejemplo:
  $\text{es\_bloque}(a), \text{es\_bloque}(b), \text{es\_bloque}(c), \text{es\_bloque}(d), \text{es\_bloque}(e)$
  $\text{superior}(b) = a, \text{superior}(c) = b, \text{superior}(e) = d$
- Definiciones:
  - $\text{sobre\_mesa}(x)$ se verifica si el bloque x está sobre la mesa
    $(\forall x)[\text{sobre\_mesa}(x) \leftrightarrow \text{es\_bloque}(x) \land \neg(\exists y)\, \text{superior}(y) = x]$
  - $\text{libre}(x)$ se verifica si el bloque x no tiene bloques encima
    $(\forall x)[\text{libre}(x) \leftrightarrow \neg(\exists y)\, \text{superior}(x) = y]$
  - $\text{tope}(x)$ es el bloque libre que está encima de x

$$
\begin{aligned}
(\forall x)[\ & (\text{libre}(x) \rightarrow \text{tope}(x) = x) \land \\
& (\neg \text{libre}(x) \rightarrow \text{tope}(x) = \text{tope}(\text{superior}(x)))]
\end{aligned}
$$

5

Lenguaje de primer orden

- Símbolos lógicos:
  - Variables: $x, y, z, \dots, x_1, x_2, \dots$.
  - Conectivas: $\neg, \land, \lor, \rightarrow, \leftrightarrow$.
  - Cuantificadores: $\forall, \exists$.
  - Símbolo de igualdad: $=$.
- Símbolos propios:
  - Símbolos de constantes: $a, b, c, \dots, a_1, a_2, \dots$.
  - Símbolos de predicado (con aridad): $P, Q, R, \dots, P_1, P_2, \dots$.
  - Símbolos de función (con aridad): $f, g, h, \dots, f_1, f_2, \dots$.
- Símbolos auxiliares: "(", ")", ",".
- Notación:
  - $L, L_1, L_2, \dots$ representan lenguajes de primer orden.
  - Var representa el conjunto de las variables.
- Los símbolos de predicados de aridad mayor que 1 se llaman de relaciones.

6

Ejemplos de lenguajes de primer orden

- Lenguaje del mundo de los bloques:
  - Símbolos de constantes: a, b, c, d, e
  - Símbolos de predicado (y de relación):
    – de aridad 1: $\text{sobre\_mesa}, \text{libre}, \text{es\_bloque}$
    – de aridad 2: $\text{sobre}, \text{bajo}, \text{encima}$
    – de aridad 3: pila
  - Símbolos de función (de aridad 1): $\text{superior}, \text{tope}$
- Lenguaje de la aritmética:
  - Símbolos de constantes: $0, 1$
  - Símbolos de función:
    – monaria: s (siguiente)
    – binarias: $+, \cdot$
  - Símbolo de predicado binario: $<$

Términos

- Def. de término de un lenguaje de primer orden L:
  - Las variables son términos de L.
  - Las constantes de L son términos de L.
  - Si f es un símbolo de función n–aria de L y $t_1, \dots, t_n$ son términos de L, entonces $f(t_1, \dots, t_n)$ es un término de L.
- Ejemplos:
  - En el lenguaje de la aritmética,
    - $+(\cdot(x, 1), s(y))$ es un término, que se suele escribir como $(x \cdot 1) + s(y)$
    - $+(\cdot(x, <), s(y))$ no es un término
  - En el lenguaje del mundo de los bloques,
    - $\text{superior}(\text{superior}(c))$ es un término.
    - $\text{libre}(\text{superior}(c))$ no es un término.
- Notación:
  - $s, t, t_1, t_2, \dots$ representan términos.
  - $\text{Térm}(L)$ representa el conjunto de los términos de L

Fórmulas atómicas

- Def. de fórmula atómica de un lenguaje de primer orden L:
  - Si t_1 y t_2 son términos de L, entonces $t_1 = t_2$ es una fórmula atómica de L.
  - Si P es un símbolo de relación n–aria de L y $t_1, \dots, t_n$ son términos de L, entonces $P(t_1, \dots, t_n)$ es una fórmula atómica de L.
- Ejemplos:
  - En el lenguaje de la aritmética,
    - $< (\cdot(x,1), s(y))$ es una fórmula atómica que se suele escribir como $x \cdot 1 < s(y)$
    - $+(x,y) = \cdot(x,y)$ es una fórmula atómica que se suele escribir como $x + y = x \cdot y$
  - En el lenguaje del mundo de los bloques,
    - $\text{libre}(\text{superior}(c))$ es una fórmula atómica.
    - $\text{tope}(c) = \text{superior}(b)$ es una fórmula atómica.
- Notación:
  - $A, B, A_1, A_2, \dots$ representan fórmulas atómicas.
  - Atom(L) representa el conjunto de las fórmulas atómicas de L.

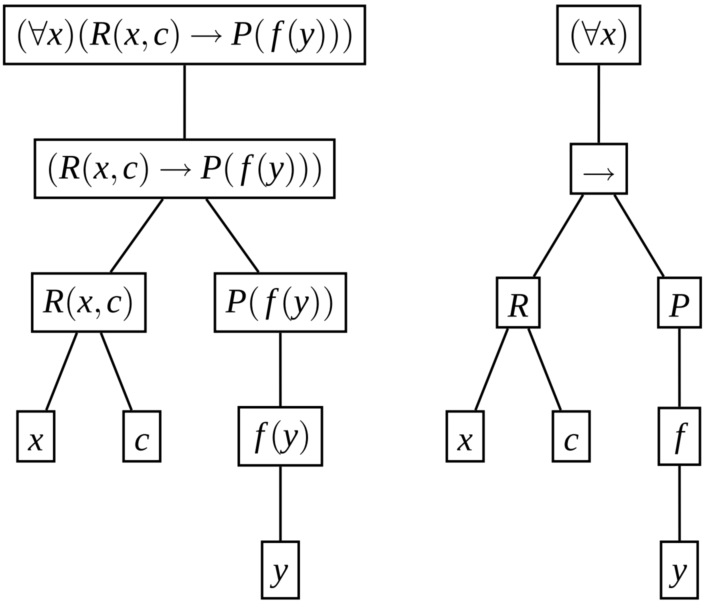
9

Fórmulas

- Definición de las fórmulas de L:
  - Las fórmulas atómicas de L son fórmulas de L.
  - Si F y G son fórmulas de L, entonces $\neg F$, $(F \wedge G)$, $(F \vee G)$, $(F \rightarrow G)$ y $(F \leftrightarrow G)$ son fórmulas de L.
  - Si F es una fórmula de L, entonces $(\forall x)F$ y $(\exists x)F$ son fórmulas de L.
- Ejemplos:
  - En el lenguaje de la aritmética,
    - $(\forall x)(\exists y) < (x,y)$ es una fórmula que se escribe como $(\forall x)(\exists y) x < y$
    - $(\forall x)(\exists y) + (x,y)$ no es una fórmula.
  - En el lenguaje del mundo de los bloques,
    - $(\forall x)(\text{tope}(x) = x \leftrightarrow \text{libre}(x))$ es una fórmula.
- Notación:
  - $F, G, H, F_1, F_2, \dots$ representan fórmulas.
  - Fórm(L) representa el conjunto de las fórmulas de L.

10

Árboles de análisis (o de formación)



Subfórmulas

- Def: El conjunto Subf(F) de las subfórmulas de una fórmula F se define recursivamente por:

$$\text{Subf}(F) = \begin{cases} \{F\}, & \text{si } F \text{ es una fórmula atómica;} \\ \{F\} \cup \text{Subf}(G), & \text{si } F = \neg G; \\ \{F\} \cup \text{Subf}(G) \cup \text{Subf}(H), & \text{si } F = G * H; \\ \{F\} \cup \text{Subf}(G), & \text{si } F = (\forall x)G; \\ \{F\} \cup \text{Subf}(G), & \text{si } F = (\exists x)G \end{cases}$$

- Ejemplo:

$$\begin{aligned} \text{Subf}((\forall x)(R(x,c) \rightarrow P(f(y)))) = \{ \; & (\forall x)(R(x,c) \rightarrow P(f(y))), \\ & (R(x,c) \rightarrow P(f(y))), \\ & R(x,c), \\ & P(f(y))\} \end{aligned}$$

Criterios de reducción de paréntesis

- Pueden eliminarse los paréntesis externos.
  $F \wedge G$ es una abreviatura de $(F \wedge G)$
- Precedencia de asociación de conectivas y cuantificadores: $\forall, \exists, \neg, \wedge, \vee, \rightarrow, \leftrightarrow$.
  $\forall x P(x) \rightarrow Q(x)$ es una abreviatura de $((\forall x) P(x)) \rightarrow Q(x)$
- Cuando una conectiva se usa repetidamente, se asocia por la derecha.

  $F \vee G \vee H$ es una abreviatura de $(F \vee (G \vee H))$

  $F \wedge G \wedge H \rightarrow \neg F \vee G$ es una abreviatura de $((F \wedge (G \wedge H)) \rightarrow (\neg F \vee G))$
- Los símbolos binarios pueden escribirse en notación infija.

  $x + y$ es una abreviatura de $+(x,y)$

  $x < y$ es una abreviatura de $<(x,y)$

13

Conjuntos de variables

- Def.: El conjunto de las variables del término t se define recursivamente por:

$$\mathrm{V}(t) = \begin{cases} \emptyset, & \text{si } t \text{ es una constante;} \\ \{x\}, & \text{si } t \text{ es una variable } x; \\ \mathrm{V}(t_1) \cup \cdots \cup \mathrm{V}(t_n), & \text{si } t \text{ es } f(t_1, \ldots, t_n) \end{cases}$$

- Def.: El conjunto de las variables de la fórmula F se define recursivamente por:

$$\mathrm{V}(F) = \begin{cases} \mathrm{V}(t_1) \cup \mathrm{V}(t_2), & \text{si } F \text{ es } t_1 = t_2; \\ \mathrm{V}(t_1) \cup \cdots \cup \mathrm{V}(t_n), & \text{si } F \text{ es } P(t_1, \ldots, t_n); \\ \mathrm{V}(G), & \text{si } F \text{ es } \neg G; \\ \mathrm{V}(G) \cup \mathrm{V}(H), & \text{si } F \text{ es } G * H; \\ \mathrm{V}(G), & \text{si } F \text{ es } (\forall x) G; \\ \mathrm{V}(G), & \text{si } F \text{ es } (\exists x) G \end{cases}$$

- Ejemplos:
  - El conjunto de las variables de $(\forall x)(R(x,c) \rightarrow P(f(y)))$ es $\{x,y\}$.
  - El conjunto de las variables de $(\forall x)(R(a,c) \rightarrow P(f(y)))$ es $\{y\}$.

14

Apariciones libres y ligadas

- Def.: Una aparición (u ocurrencia) de la variable x en la fórmula F es ligada si es en una subfórmula de F de la forma $(\forall x)G$ ó $(\exists x)G$.
- Def.: Una aparición (u ocurrencia) de la variable x en la fórmula F es libre si no es ligada
- Ejemplo: Las apariciones ligadas son las subrayadas:

  $(\forall x)(P(\underline{x}) \rightarrow R(\underline{x},y)) \rightarrow ((\exists y)P(\underline{y}) \rightarrow R(z,x))$

  $(\exists x)R(\underline{x},y) \vee (\forall y)P(\underline{y})$

  $(\forall x)(P(\underline{x}) \rightarrow (\exists y)R(\underline{x},\underline{y}))$

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

Variables libres y ligadas

- Def.: La variable x es libre en F si tiene una aparición libre en F.
- Def.: La variable x es ligada en F si tiene una aparición ligada en F.
- Prop.: El conjunto de las variables libres de una fórmula F es:

$$\mathrm{VL}(F) = \begin{cases} \mathrm{V}(t_1) \cup \mathrm{V}(t_2), & \text{si } F \text{ es } t_1 = t_2; \\ \mathrm{V}(t_1) \cup \cdots \cup \mathrm{V}(t_n), & \text{si } F \text{ es } P(t_1,\dots,t_n); \\ \mathrm{VL}(G), & \text{si } F \text{ es } \neg G; \\ \mathrm{VL}(G) \cup \mathrm{VL}(H), & \text{si } F \text{ es } G * H; \\ \mathrm{VL}(G) \setminus \{x\}, & \text{si } F \text{ es } (\forall x)G; \\ \mathrm{VL}(G) \setminus \{x\}, & \text{si } F \text{ es } (\exists x)G \end{cases}$$

- Ejemplo:

Fórmula	Ligadas	Libres
$(\forall x)(P(x) \rightarrow R(x,y)) \rightarrow ((\exists y)P(y) \rightarrow R(x,z))$	x,y	x,y,z
$(\forall x)(P(x) \rightarrow (\exists y)R(x,y))$	x,y	
$(\forall z)(P(x) \rightarrow R(x,y))$		x,y

Fórmulas cerradas y abiertas

- Fórmula cerradas:
  - Def.: Una fórmula cerrada (o sentencia) es una fórmula sin variables libres.
  - Ejemplos: $(\forall x)(P(x) \rightarrow (\exists y)R(x,y))$ es cerrada.
    $(\exists x)R(x,y) \vee (\forall y)P(y)$ no es cerrada.
- Fórmulas abiertas:
  - Def.: Una fórmula abierta es una fórmula con variables libres.
  - Ejemplos: $(\forall x)(P(x) \rightarrow (\exists y)R(x,y))$ no es abierta.
    $(\exists x)R(x,y) \vee (\forall y)P(y)$ es abierta.

Semántica: Estructuras, asignaciones e interpretaciones

- Una estructura del lenguaje L es un par $\mathscr{I} = (U,I)$ tal que:
  - U es un conjunto no vacío, denominado universo de la estructura;
  - I es una función con dominio el conjunto de símbolos propios de L tal que
    - si c es una constante de L, entonces $I(c) \in U$;
    - si f es un símbolo de función n–aria de L, entonces $I(f) : U^n \rightarrow U$;
    - si P es un símbolo de relación 0–aria de L, entonces $I(P) \in \{1,0\}$;
    - si R es un símbolo de relación n–aria ($n > 0$) de L, entonces $I(R) \subseteq U^n$;
- Una asignación A en una estructura (U,I) es una función $A : \text{Var} \rightarrow U$ que hace corresponder a cada variable del alfabeto un elemento del universo de la estructura.
- Una interpretación de L es un par $(\mathscr{I},A)$ formado por una estructura $\mathscr{I}$ de L y una asignación A en $\mathscr{I}$.
- Notación: A veces se usa para los valores de verdad V y F en lugar de 1 y 0.

Ejemplos de estructuras

Sea L el lenguaje de la aritmética cuyos símbolos propios son:
constante: 0;
símbolo de función monaria: s;
símbolo de función binaria: $+$ y
símbolo de relación binaria: $\leq$

- Primera estructura de L:
  $U_1 = \mathbb{N}$
  $I_1(0) = 0$
  $I_1(s) = \{(n, n+1) : n \in \mathbb{N}\}$ (sucesor)
  $I_1(+) = \{(a, b, a+b) : a, b \in \mathbb{N}\}$ (suma)
  $I_1(\leq) = \{(n, m) : n, m \in \mathbb{N}, n \leq m\}$ (menor o igual)
- Segunda estructura de L:
  $U_2 = \{0, 1\}^*$ (cadenas de 0 y 1)
  $I_2(0) = \varepsilon$ (cadena vacía)
  $I_2(s) = \{(w, w1) : w \in \{0, 1\}^*\}$ (siguiente)
  $I_2(+) = \{(w_1, w_2, w_1w_2) : w_1, w_2 \in \{0, 1\}^*\}$ (concatenación)
  $I_2(\leq) = \{(w_1, w_2) : w_1, w_2 \in \{0, 1\}^*, w_1$ es prefijo de $w_2\}$ (prefijo)

Ejemplos de estructuras

- Tercera estructura de L:
  $U_3 = \{abierto, cerrado\}$
  $I_3(0) = cerrado$
  $I_3(s) = \{(abierto, cerrado), (cerrado, abierto)\}$

$$I_3(+) = \{ \; (abierto, abierto, abierto), (abierto, cerrado, abierto), \\ (cerrado, abierto, abierto), (cerrado, cerrado, cerrado)\}$$

$$I_3(\leq) = \{ \; (abierto, abierto), (cerrado, abierto), (cerrado, cerrado)\}$$

e	$I_3(s)(e)$
abierto	*cerrado*
cerrado	*abierto*

$I_3(+)$	*abierto*	*cerrado*
abierto	*abierto*	*abierto*
cerrado	*abierto*	*cerrado*

$I_3(\leq)$	*abierto*	*cerrado*
abierto	1	0
cerrado	1	1

Ejemplo de evaluación de términos

- Sean L el lenguaje de la página 19 y t el término $s(x+s(0))$.
  - Si $\mathscr{I}$ es la primera estructura y $A(x)=3$, entonces

$$\begin{array}{lll} \mathscr{I}_A(t) & = \mathscr{I}_A(s(x+s(0))) & = s^I(3+^I s^I(0^I)) = \\ & = s^I(3+^I s^I(0)) & = s^I(3+^I 1) = \\ & = s^I(4) & = 5 \end{array}$$

  - Si $\mathscr{I}$ es la segunda estructura y $A(x)=10$, entonces

$$\begin{array}{lll} \mathscr{I}_A(t) & = \mathscr{I}_A(s(x+s(0))) & = s^I(10+^I s^I(0^I)) = \\ & = s^I(10+^I s^I(\varepsilon)) & = s^I(10+^I 1) = \\ & = s^I(101) & = 1011 \end{array}$$

  - Si $\mathscr{I}$ es la tercera estructura y $A(x)=abierto$, entonces

$$\begin{array}{lll} \mathscr{I}_A(t) & = \mathscr{I}_A(s(x+s(0))) & = s^I(abierto+^I s^I(0^I)) = \\ & = s^I(abierto+^I s^I(cerrado)) & = s^I(abierto+^I abierto) = \\ & = s^I(abierto) & = cerrado \end{array}$$

Evaluación de términos

- Def.: Dada una estructura $\mathscr{I}=(U,I)$ de L y una asignación A en $\mathscr{I}$, se define la función de evaluación de términos $\mathscr{I}_A : \text{Térm}(L) \to U$ por

$$\mathscr{I}_A(t) = \begin{cases} I(c), & \text{si } t \text{ es una constante } c; \\ A(x), & \text{si } t \text{ es una variable } x; \\ I(f)(\mathscr{I}_A(t_1),\dots,\mathscr{I}_A(t_n)), & \text{si } t \text{ es } f(t_1,\dots,t_n) \end{cases}$$

- $\mathscr{I}_A(t)$ se lee "el valor de t en $\mathscr{I}$ respecto de A".
- Ejemplo: Sean L el lenguaje de la página 19, t el término $s(+(x,s(0)))$, $\mathscr{I}$ la primera estructura y $A(x)=3$.

$$\begin{array}{lll} \mathscr{I}_A(t) & = \mathscr{I}_A(s(+(x,s(0)))) & = I(s)(\mathscr{I}_A(+(x,s(0)))) = \\ & = I(s)(I(+)(\mathscr{I}_A(x),\mathscr{I}_A(s(0)))) & = I(s)(I(+)(A(x),\mathscr{I}_A(s(0)))) = \\ & = I(s)(I(+)(3,I(s)(\mathscr{I}_A(0)))) & = I(s)(I(+)(3,I(s)(I(0)))) = \\ & = I(s)(I(+)(3,I(s)(0))) & = I(s)(I(+)(3,1)) = \\ & = I(s)(4) & = 5 \end{array}$$

Evaluación de fórmulas

- Def.: Dada una estructura $\mathscr{I} = (U, I)$ de L y una asignación A sobre $\mathscr{I}$, se define la función de evaluación de fórmulas $\mathscr{I}_A : \text{Fórm}(L) \to \mathbb{B}$ por
  - Si F es $t_1 = t_2$, $\quad \mathscr{I}_A(F) = H_=(\mathscr{I}_A(t_1), I_A(t_2))$
  - Si F es $P(t_1, \dots, t_n)$, $\quad \mathscr{I}_A(F) = H_{I(P)}(\mathscr{I}_A(t_1), \dots, \mathscr{I}_A(t_n))$
  - Si F es $\neg G$, $\quad \mathscr{I}_A(F) = H_\neg(\mathscr{I}_A(G))$
  - Si F es $G * H$, $\quad \mathscr{I}_A(F) = H_*(\mathscr{I}_A(G), \mathscr{I}_A(H))$
  - Si F es $(\forall x)G$, $\quad \mathscr{I}_A(F) = \begin{cases} 1, & \text{si para todo } u \in U \text{ se tiene} \\ & \mathscr{I}_{A[x/u]}(G) = 1; \\ 0, & \text{en caso contrario} \end{cases}$
  - Si F es $(\exists x)G$, $\quad \mathscr{I}_A(F) = \begin{cases} 1, & \text{si existe algún } u \in U \text{ tal que} \\ & \mathscr{I}_{A[x/u]}(G) = 1; \\ 0, & \text{en caso contrario} \end{cases}$
- $\mathscr{I}_A(F)$ se lee "el valor de F en $\mathscr{I}$ respecto de A".

Conceptos auxilares para la evaluación de fórmulas

- La función de verdad de la igualdad en U es la función $H_= : U^2 \to \mathbb{B}$ definida por
$$H_=(u_1, u_2) = \begin{cases} 1, & \text{si } u_1 = u_2; \\ 0, & \text{en caso contrario} \end{cases}$$
- Función de verdad de una relación: Si R es una relación n–aria en U (i.e. $R \subseteq U^n$), entonces la función de verdad de R es la función $H_R : U^n \to \mathbb{B}$ definida por
$$H_R(u_1, \dots, u_n) = \begin{cases} 1, & \text{si } (u_1, \dots, u_n) \in R; \\ 0, & \text{en caso contrario} \end{cases}$$
- Variante de una asignación: Sea A una asignación en la estructura (U, I) y $u \in U$. Mediante $A[x/u]$ se representa la asignación definida por
$$A[x/u](y) = \begin{cases} u, & \text{si } y \text{ es } x; \\ A(y) & \text{si } y \text{ es distinta de } x \end{cases}$$

Ejemplo de evaluación de fórmula

Evaluación de $(\forall x)(\exists y)P(x,y)$ en la estructura $\mathscr{I} = (U,I)$ respecto de la asignación A tales que $U = \{1,2\}$ e $I(P) = \{(1,1),(2,2)\}$

$$\mathscr{I}_A((\forall x)(\exists y)P(x,y)) = \mathsf{V} \Leftrightarrow \mathscr{I}_{A[x/1]}((\exists y)P(x,y)) = \mathsf{V} \text{ y } \mathscr{I}_{A[x/2]}((\exists y)P(x,y)) = \mathsf{V}$$

$$\mathscr{I}_{A[x/1]}((\exists y)P(x,y)) = \mathsf{V} \Leftrightarrow \mathscr{I}_{A[x/1,y/1]}P(x,y) = \mathsf{V} \text{ ó } \mathscr{I}_{A[x/1,y/2]}P(x,y) = \mathsf{V}$$

$$\begin{aligned} \mathscr{I}_{A[x/1,y/1]}P(x,y) &= P^I(1,1) \\ &= \mathsf{V} \end{aligned}$$

Luego, $\mathscr{I}_{A[x/1]}((\exists y)P(x,y)) = \mathsf{V}$.

$$\mathscr{I}_{A[x/2]}((\exists y)P(x,y)) = \mathsf{V} \Leftrightarrow \mathscr{I}_{A[x/2,y/1]}P(x,y) = \mathsf{V} \text{ ó } \mathscr{I}_{A[x/2,y/2]}P(x,y) = \mathsf{V}$$

$$\begin{aligned} \mathscr{I}_{A[x/2,y/2]}P(x,y) &= P^I(2,2) \\ &= \mathsf{V} \end{aligned}$$

Luego, $\mathscr{I}_{A[x/2]}((\exists y)P(x,y)) = \mathsf{V}$.

Por tanto, $\mathscr{I}_A((\forall x)(\exists y)P(x,y)) = \mathsf{V}$

Ejemplo de evaluación de fórmulas

Ejemplo: Evaluación de $(\forall x)g(g(x)) = x$ en la estructura $\mathscr{I} = (U,I)$ respecto de la asignación A tales que $U = \{1,2\}$, $I(a) = 1$, $I(g) = \{(1,2),(2,1)\}$, $I(P) = \{2\}$ e $I(Q) = \{(1,2),(2,2)\}$.

$$\mathscr{I}_A((\forall x)g(g(x)) = x) = \mathsf{V} \Leftrightarrow \mathscr{I}_{A[x/1]}g(g(x)) = x = \mathsf{V} \text{ y } \mathscr{I}_{A[x/2]}g(g(x)) = x = \mathsf{V}$$

$$\begin{aligned} \mathscr{I}_{A[x/1]}(g(g(x)) = x) &= (g^I(g^I(1)) = 1) \\ &= (g^I(2) = 1) \\ &= (1 = 1) \\ &= \mathsf{V} \end{aligned}$$

$$\begin{aligned} \mathscr{I}_{A[x/2]}(g(g(x)) = x) &= (g^I(g^I(2)) = 2) \\ &= (g^I(1) = 2) \\ &= (2 = 2) \\ &= \mathsf{V} \end{aligned}$$

Por tanto, $\mathscr{I}_A((\forall x)g(g(x)) = x) = \mathsf{V}$.

Dependencias en la evaluación de fórmulas

- Ejemplo de dependencia del universo: Sea G la fórmula $(\forall x)(\exists y)R(y,x)$, entonces
  - $\mathscr{I}_A(G) = $ V, siendo $\mathscr{I} = (\mathbb{Z}, I), I(R) = <$ y A una asignación en $\mathscr{I}$.
  - $\mathscr{I}_A(G) = $ F, siendo $\mathscr{I} = (\mathbb{N}, I), I(R) = <$ y A una asignación en $\mathscr{I}$.
- Ejemplo de dependencia de la estructura: Sea G la fórmula $(\exists x)(\forall y)R(x,y)$, entonces
  - $\mathscr{I}_A(G) = $ V, siendo $\mathscr{I} = (\mathbb{N}, I), I(R) = \leq$ y A una asignación en $\mathscr{I}$.
  - $\mathscr{I}_A(G) = $ F, siendo $\mathscr{I} = (\mathbb{N}, I), I(R) = \geq$ y A una asignación en $\mathscr{I}$.
- Ejemplo de dependencia de la asignación: Sea G la fórmula $(\forall y)R(x,y)$, entonces
  - $\mathscr{I}_A(G) = $ V, siendo $\mathscr{I} = (\mathbb{N}, I), I(R) = \leq$ y A una asignación en $\mathscr{I}$ tal que $A(x) = 0$.
  - $\mathscr{I}_A(G) = $ F, siendo $\mathscr{I} = (\mathbb{N}, I), I(R) = \leq$ y A una asignación en $\mathscr{I}$ tal que $A(x) = 5$.

Evaluación y variables libres

- Sea t un término de L e $\mathscr{I}$ una estructura de L.
  - Si A y B son dos asignaciones en $\mathscr{I}$ que coinciden sobre las variables de t, entonces $\mathscr{I}_A(t) = \mathscr{I}_B(t)$.
  - Si t no tiene variables, entonces $\mathscr{I}_A(t) = \mathscr{I}_B(t)$ para cualesquiera asignaciones A y B en $\mathscr{I}$. Se suele escribir simplemente $\mathscr{I}(t)$.
- Sea F una fórmula de L e $\mathscr{I}$ una estructura de L.
  - Si A y B son dos asignaciones en $\mathscr{I}$ que coinciden sobre las variables libres de F, entonces $\mathscr{I}_A(F) = \mathscr{I}_B(F)$.
  - Si F es cerrada, entonces $\mathscr{I}_A(F) = \mathscr{I}_B(F)$ para cualesquiera asignaciones A y B en $\mathscr{I}$. Se suele escribir simplemente $\mathscr{I}(F)$.

Modelo de una fórmula

- Sean F una fórmula de L e $\mathscr{I}$ una estructura de L.
  - $(\mathscr{I},A)$ es una realización de F si A una asignación en $\mathscr{I}$ tal que $\mathscr{I}_A(F) = 1$.
    Se representa por $\mathscr{I}_A \models F$.
  - $\mathscr{I}$ es un modelo de F si, para todo asignación A en $\mathscr{I}$, $\mathscr{I}_A(F) = 1$.
    Se representa por $\mathscr{I} \models F$.
- Ejemplos: Sea $\mathscr{I} = (\mathbb{N}, I)$ una estructura tal que $I(f) = +$ e $I(g) = *$.
  - Si A es una asignación en $\mathscr{I}$ tal que $A(x) = A(y) = 2$. entonces
    $\mathscr{I}_A \models f(x,y) = g(x,y)$,
  - Si B es una asignación en $\mathscr{I}$ tal que $B(x) = 1, B(y) = 2$. entonces
    $\mathscr{I}_B \not\models f(x,y) = g(x,y)$,
  - $I \not\models f(x,y) = g(x,y)$
  - $I \models f(x,y) = f(y,x)$

Satisfacibilidad y validez

- Def.: Sea F una fórmula de L.
  - F es válida si toda estructura de L es modelo de F
    (i.e. para toda estructura $\mathscr{I}$ de L y toda asignación A en $\mathscr{I}$ se tiene que $\mathscr{I}_A(F) = 1$).
    Se representa por $\models F$.
  - F es satisfacible si tiene alguna realización
    (i.e. existe alguna estructura $\mathscr{I}$ de L y alguna asignación A en I tales que $\mathscr{I}_A(F) = 1$).
  - F es insatisfacible si no tiene ninguna realización
    (i.e. para toda estructura $\mathscr{I}$ de L y toda asignación A en I se tiene que $\mathscr{I}_A(F) = 0$).
- Ejemplos:
  - $(\exists x)P(x) \vee (\forall x)\neg P(x)$ es válida.
  - $(\exists x)P(x) \wedge (\exists x)\neg P(x)$ es satisfacible, pero no es válida.
    $\mathscr{I}_A((\exists x)P(x) \wedge (\exists x)\neg P(x)) = 1$, siendo $\mathscr{I} = (\{a,b\}, I)$, $I(P) = \{a\}$.
    $\mathscr{I}_A((\exists x)P(x) \wedge (\exists x)\neg P(x)) = 1$, siendo $\mathscr{I} = (\{a\}, I)$, $I(P) = \{a\}$.
  - $(\forall x)P(x) \wedge (\exists x)\neg P(x)$ es insatisfacible.

Satisfacibilidad y validez

- Prop.: F es válida syss $\neg F$ es insatisfacible.
  F es válida
  $\iff$ para toda estructura $\mathscr{I}$ y toda asignación A se tiene que $\mathscr{I}_A(F) = 1$
  $\iff$ para toda estructura $\mathscr{I}$ y toda asignación A se tiene que $\mathscr{I}_A(\neg F) = 0$
  $\iff$ $\neg F$ es insatisfacible.
- Si F es válida, entonces F es satisfacible.
  F es válida
  $\Longrightarrow$ para toda estructura $\mathscr{I}$ y toda asignación A se tiene que $\mathscr{I}_A(F) = 1$
  $\Longrightarrow$ existe una estructura $\mathscr{I}$ y una asignación A tales que $\mathscr{I}_A(F) = 1$
  $\Longrightarrow$ F es satisfacible.
- F es satisfacible $\;\not\Longrightarrow\;$ $\neg F$ es insatisfacible.
  $(\forall x)P(x)$ y $\neg(\forall x)P(x)$ son satisfacibles.
- Sea F una fórmula de L y $x_1, \dots, x_n$ las variables libres de F.
  - F es válida syss $(\forall x_1)\dots(\forall x_n)F$ es válida. [Cierre universal].
  - F es satisfacible syss $(\exists x_1)\dots(\exists x_n)F$ es satisfacible. [Cierre existencial].

Modelo de un conjunto de fórmulas

- Notación: $S, S_1, S_2, \dots$ representarán conjuntos de fórmulas.
- Def.: Sean S un conjunto de fórmulas de L e $\mathscr{I}$ una estructura de L y A una asignación en $\mathscr{I}$.
  - $(\mathscr{I}, A)$ es una realización de S si A es una asignación en $\mathscr{I}$ tal que para toda $F \in S$ se tiene que $\mathscr{I}_A(F) = 1$.
    Se representa por $\mathscr{I}_A \models S$.
  - $\mathscr{I}$ es un modelo de S si para toda $F \in S$ se tiene que $\mathscr{I} \models F$
    (i.e. para toda $F \in S$ y toda asignación A en $\mathscr{I}$ se tiene $\mathscr{I}_A(F) = 1$).
    Se representa por $\mathscr{I} \models S$.
- Ejemplos:
  - Sea $S = \{(\forall y)R(x,y),\ (\forall y)f(x,y) = y\}$.
    - $(\mathscr{I}, A)$ con $\mathscr{I} = (\mathbb{N}, I), R^I = \leq, f^I = +, A(x) = 0$ es realización de S.
    - $(\mathscr{I}, A)$ con $\mathscr{I} = (\mathbb{N}, I), R^I = <, f^I = +, A(x) = 0$ no es realización de S.
  - Sea $S = \{R(e,y),\ f(e,y) = y\}$.
    - $\mathscr{I} = (\mathbb{N}, I)$ con $R^I = \leq, f^I = +, e^I = 0$ es modelo de S.
    - $\mathscr{I} = (\mathbb{N}, I)$ con $R^I = <, f^I = +, e^I = 0$ no es modelo de S.

Consistencia de un conjunto de fórmulas

- Def.: Sea S un conjunto de fórmulas de L.
  - S es consistente si S tiene alguna realización (i.e. existe alguna estructura $\mathscr{I}$ de L y alguna asignación A en $\mathscr{I}$ tales que, para toda $F \in S$, $I_A(F) = 1$).
  - S es inconsistente si S no tiene ninguna realización (i.e. para toda estructura $\mathscr{I}$ de L y toda asignación A en $\mathscr{I}$, existe alguna $F \in S$, tal que $I_A(F) = 0$).
- Ejemplos:
  - $S = \{(\forall y)R(x,y),\ (\forall y)f(x,y) = y\}$ es consistente .
    $(\mathscr{I},A)$ con $\mathscr{I} = (\mathbb{N},I), R^I = \leq, f^I = +, A(x) = 0$ es realización de S.
  - $S = \{P(x) \rightarrow Q(x),\ (\forall y)P(y),\ \neg Q(x)\}$ es inconsistente.
- Prop.: Sea S un conjunto de fórmulas *cerradas* de L. Entonces S es consistente syss S tiene algún modelo.

33

Consecuencia lógica

- Def.: Sean F una fórmula de L y S un conjunto de fórmulas de L.
  - F es consecuencia lógica de S si todas las realizaciones de S lo son de F.
    (i.e. para toda estructura $\mathscr{I}$ de L y toda asignación A en $\mathscr{I}$,
    si $\mathscr{I}_A \models S$ entonces $\mathscr{I}_A \models F$).
    Se representa por $S \models F$.
  - Se escribe $G \models F$ en lugar de $\{G\} \models F$.
  - Se escribe $G \not\models F$ en lugar de $\{G\} \not\models F$.
- Ejemplos:
  - $(\forall x)P(x) \models P(y)$
  - $P(y) \not\models (\forall x)P(x)$
    $(\mathscr{I},A)$ con $\mathscr{I} = (U,I), U = \{1,2\}, P^I = \{1\}, A(y) = 1$.
  - $\{(\forall x)(P(x) \rightarrow Q(x)),\ P(c)\} \models Q(c)$
  - $\{(\forall x)(P(x) \rightarrow Q(x)),\ Q(c)\} \not\models P(c)$
    $(\mathscr{I},A)$ con $\mathscr{I} = (U,I), U = \{1,2\}, c^I = 1, P^I = \{2\}, Q^I = \{1,2\}$.
  - $\{(\forall x)(P(x) \rightarrow Q(x)),\ \neg Q(c)\} \models \neg P(c)$
  - $\{P(c),\ \neg P(d)\} \models c \neq d$

34

Consecuencia lógica e inconsistencia

- $S \models F$ syss $S \cup \{\neg F\}$ es inconsistente.

  $S \models F$

  $\iff$ para toda estructura $\mathscr{I}$ de L y toda asignación A en $\mathscr{I}$,
  si, para todo $G \in S$, $\mathscr{I}_A(G) = 1$ entonces $\mathscr{I}_A(F) = 1$.

  $\iff$ para toda estructura $\mathscr{I}$ de L y toda asignación A en $\mathscr{I}$,
  si, para todo $G \in S$, $\mathscr{I}_A(G) = 1$ entonces $\mathscr{I}_A(\neg F) = 0$.

  $\iff$ para toda estructura $\mathscr{I}$ de L y toda asignación A en $\mathscr{I}$,
  existe alguna $H \in S \cup \{\neg F\}$ tal que $\mathscr{I}_A(H) = 0$.

  $\iff$ $S \cup \{\neg F\}$ es inconsistente.
- Sean F una fórmula *cerrada* de L y S un conjunto de fórmulas *cerradas* de L. Entonces, son equivalentes
  - F es consecuencia lógica de S
  - todos los modelos de S lo son de F.

Equivalencia lógica

- Def.: Sean F y G fórmulas de L. F y G son equivalentes si para toda estructura $\mathscr{I}$ de L y toda asignación A en $\mathscr{I}$, $\mathscr{I}_A(F) = \mathscr{I}_A(G)$.
  Se representa por $F \equiv G$.
- Ejemplos:
  - $P(x) \not\equiv P(y)$.
    $\mathscr{I} = (\{1,2\}, I)$ con $P^I = \{1\}$ y $A(x) = 1, A(y) = 2$.
  - $(\forall x)P(x) \equiv (\forall y)P(y)$.
  - $(\forall x)(P(x) \wedge Q(x)) \equiv (\forall x)P(x) \wedge (\forall x)Q(x)$.
  - $(\exists x)(P(x) \wedge Q(x)) \not\equiv (\exists x)P(x) \wedge (\exists x)Q(x)$.
    $\mathscr{I} = (\{1,2\}, I)$ con $P^I = \{1\}$ y $Q^I = \{2\}$.
- Propiedades: Sean F y G fórmulas cerradas de L.
  - $F \equiv G$ syss $\models F \leftrightarrow G$.
  - $F \equiv G$ syss $F \models G$ y $G \models F$.

Bibliografía

1. C. Badesa, I. Jané y R. Jansana *Elementos de lógica formal.* (Ariel, 2000) pp. 195–259 y 323–326.
2. M.L. Bonet *Apuntes de LPO.* (Univ. Politécnica de Cataluña, 2003) pp. 17–26.
3. C.–L. Chang y R.C.–T. Lee *Symbolic logic and mechanical theorem proving* (Academic Press, 1973) pp. 26–35.
4. J.L. Fernández, A. Manjarrés y F.J. Díez *Lógica computacional.* (UNED, 2003) pp. 64–87.
5. J.H. Gallier *Logic for computer science (foundations of automatic theorem Proving)* (June 2003) pp. 146–186.
6. M. Huth y M. Ryan *Logic in computer science: modelling and reasoning about systems.* (Cambridge University Press, 2000) pp. 90–109 y 128–140.
7. M. Ojeda e I. Pérez de Guzmán *Lógica para la computación (Vol. 2: Lógica de primer orden)* (Ágora, 1997) pp. 1–37 y 49–51.
8. L. Paulson *Logic and proof* (U. Cambridge, 2002) pp. 22–29.

Capítulo 7

Deducción natural en lógica de primer orden

Lógica informática (2005–06)
Tema 7: Deducción natural en lógica de primer orden

José A. Alonso Jiménez

Grupo de Lógica Computacional
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

1

Sustituciones

- Def.: Una sustitución σ (de L) es una aplicación σ : Var $\to$ Térm(L).
- Notación: $[x_1/t_1, x_2/t_2, \dots, x_n/t_n]$ representa la sustitución σ definida por
  $$\sigma(x) = \begin{cases} t_i, & \text{si } x \text{ es } x_i; \\ x, & \text{si } x \notin \{x_1, \dots, x_n\} \end{cases}$$
- Ejemplo: $[x/s(0), y/x+y]$ la sustitución σ de Var en los términos de la aritmética definida por
  $\sigma(x) = s(0), \sigma(y) = x+y$ y $\sigma(z) = z$ para $z \in$ Var $\setminus \{x,y\}$
- Notación: $\sigma, \sigma_1, \sigma_2, \dots$ representarán sustituciones.

2

Aplicación de sustituciones a términos

- Def.: $t[x_1/t_1,\dots,x_n/t_n]$ es el término obtenido sustituyendo en t las apariciones de x_i por t_i.
- Def.: La extensión de σ a términos es la aplicación σ : Térm$(L) \to$ Térm(L) definida por

$$t\sigma = \begin{cases} c, & \text{si } t \text{ es una constante } c; \\ \sigma(x), & \text{si } t \text{ es una variable } x; \\ f(t_1\sigma,\dots,t_n\sigma), & \text{si } t \text{ es } f(t_1,\dots,t_n) \end{cases}$$

- Ejemplo: Si $\sigma = [x/f(y,a), y/z]$, entonces
  - $a\sigma = a$, donde a es una constante.
  - $w\sigma = w$, donde w es una variable distinta de x e y.
  - $h(a,x,w)\sigma = h(a\sigma,x\sigma,w\sigma) = h(a,f(y,a),w)$
  - $f(x,y)\sigma = f(x\sigma,y\sigma) = f(f(y,a),z)$
  - $h(a,f(x,y),w)\sigma = h(a\sigma,f(x,y)\sigma,w\sigma) = h(a,f(f(y,a),z),w)$

Aplicación de sustituciones a fórmulas

- Def.: $F[x_1/t_1,\dots,x_n/t_n]$ es la fórmula obtenida sustituyendo en F las apariciones libres de x_i por t_i.
- Def.: La extensión de σ a fórmulas es la aplicación σ : Fórm$(L) \to$ Fórm(L) definida por

$$F\sigma = \begin{cases} P(t_1\sigma,\dots,t_n\sigma), & \text{si } F \text{ es la fórmula atómica } P(t_1,\dots,t_n); \\ t_1\sigma = t_2\sigma, & \text{si } F \text{ es la fórmula } t_1 = t_2; \\ \neg(G\sigma), & \text{si } F \text{ es } \neg G; \\ G\sigma * H\sigma, & \text{si } F \text{ es } G * H; \\ (Qx)(G\sigma_x), & \text{si } F \text{ es } (Qx)G \text{ y } Q \in \{\forall,\exists\} \end{cases}$$

donde σ_x es la sustitución definida por

$$\sigma_x(y) = \begin{cases} x, & \text{si } y \text{ es } x; \\ \sigma(y), & \text{si } y \text{ es distinta de } x \end{cases}$$

Ejemplos de aplicación de sustituciones a fórmulas

- Ejemplos: Si $\sigma = [x/f(y), y/b]$, entonces
  - $$\begin{aligned} ((\forall x)(Q(x) \to R(x,y)))\sigma &= (\forall x)((Q(x) \to R(x,y))\sigma_x) \\ &= (\forall x)(Q(x)\sigma_x \to R(x,y)\sigma_x) \\ &= (\forall x)(Q(x) \to R(x,b)) \end{aligned}$$
  - $$\begin{aligned} (Q(x) \to (\forall x)R(x,y))\sigma &= Q(x)\sigma \to ((\forall x)R(x,y))\sigma \\ &= Q(f(y)) \to (\forall x)(R(x,y)\sigma_x) \\ &= Q(f(y)) \to (\forall x)R(x,b) \end{aligned}$$
  - $$\begin{aligned} ((\forall x)(Q(x) \to (\forall y)R(x,y)))\sigma &= (\forall x)((Q(x) \to (\forall y)R(x,y))\sigma_x) \\ &= (\forall x)(Q(x)\sigma_x \to ((\forall y)R(x,y))\sigma_x) \\ &= (\forall x)(Q(x) \to (\forall y)(R(x,y)\sigma_{xy})) \\ &= (\forall x)(Q(x) \to (\forall y)R(x,y)) \end{aligned}$$

5

Sustituciones libres

- Def.: Una sustitución se denomina libre para una fórmula cuando todas las apariciones de variables introducidas por la sustitución en esa fórmula resultan libres.
- Ejemplos:
  - $[y/x]$ no es libre para $(\exists x)(x < y)$
    $(\exists x)(x < y)[y/x] = (\exists x)(x < x)$
  - $[y/g(y)]$ es libre para $(\forall x)(P(x) \to Q(x, f(y)))$
    $(\forall x)(P(x) \to Q(x, f(y)))[y/g(y)] = (\forall x)(P(x) \to Q(x, f(g(y))))$
  - $[y/g(x)]$ no es libre para $(\forall x)(P(x) \to Q(x, f(y)))$
    $(\forall x)(P(x) \to Q(x, f(y)))[y/g(x)] = (\forall x)(P(x) \to Q(x, f(g(x))))$
- Convenio: Al escribir $F\sigma$ supondremos que σ es libre para F.

6

Regla de eliminación del cuantificador universal

- Regla de eliminación del cuantificador universal:

$$\frac{(\forall x)F}{F[x/t]}\ \forall e$$

donde $[x/t]$ es libre para F.

- Nota: Analogía con $\wedge e_1$ y $\wedge e_2$.
- Ejemplo: $P(c), (\forall x)(P(x) \rightarrow \neg Q(x)) \vdash \neg Q(c)$

1	$P(c)$	premisa
2	$(\forall x)(P(x) \rightarrow \neg Q(x))$	premisa
3	$P(c) \rightarrow Q(c)$	$\forall$e 2
4	$Q(c)$	$\rightarrow$e $3, 1$

- Nota: $(\forall x)(\exists y)(x < y) \not\vdash (\exists y)(y < y)$.

7

Regla de introducción del cuantificador universal

$$\frac{\boxed{\begin{array}{l} x_0 \\ \quad\vdots \\ F[x/x_0] \end{array}}}{(\forall x)F}\ \forall i$$

donde x_0 es una variable nueva, que no aparece fuera de la caja.

- Nota: Analogía con $\wedge$i.
- Ejemplo: $(\forall x)(P(x) \rightarrow Q(x)), (\forall x)P(x) \vdash (\forall x)Q(x)$

1	$(\forall x)(P(x) \rightarrow \neg Q(x))$	premisa
2	$(\forall x)P(x)$	premisa
3	actual x_0	supuesto
4	$P(x_0) \rightarrow \neg Q(x_0)$	$\forall$e $1, 3$
5	$P(x_0)$	$\forall$e $2, 3$
6	$Q(x_0)$	$\rightarrow$e $4, 5$
7	$(\forall x)Q(x)$	$\forall$i $3-6$

8

Regla de introducción del cuantificador existencial

- Regla de introducción del cuantificador existencial:

  $$\frac{F[x/t]}{(\exists x)F}\ \exists i$$

  donde $[x/t]$ es libre para F.
- Nota: Analogía con $\vee i_1$ y $\vee i_2$.
- Ejemplo 3: $(\forall x)P(x) \vdash (\exists x)P(x)$

1	$(\forall x)P(x)$	premisa
2	$P(x_0)$	$\forall$e 1
3	$(\exists x)P(x)$	$\exists$i 2

Regla de eliminación del cuantificador existencial

$$\frac{(\exists x)F \qquad \boxed{\begin{array}{ll} x_0 & F[x/x_0] \\ & \vdots \\ & G \end{array}}}{G}\ \exists e$$

donde x_0 es una variable nueva, que no aparece fuera de la caja.

- Nota: Analogía con $\vee$e.
- Ejemplo: $(\forall x)(P(x) \rightarrow Q(x)), (\exists x)P(x) \vdash (\exists x)Q(x)$

1	$(\forall x)(P(x) \rightarrow Q(x))$	premisa
2	$(\exists x)P(x)$	premisa
3	actual $x_0, P(x_0)$	supuesto
4	$P(x_0) \rightarrow Q(x_0)$	$\forall$e 1, 3
5	$Q(x_0)$	$\rightarrow$e 4, 3
6	$(\exists x)Q(x)$	$\exists$i 5
7	$(\exists x)Q(x)$	$\exists$e 2, 3 – 6

Equivalencias

- Sean F y G fórmulas.
  [1(a)] $\neg(\forall x)F \equiv (\exists x)\neg F$
  [1(b)] $\neg(\exists x)F \equiv (\forall x)\neg F$
- Sean F y G fórmulas y x una varible no libre en G.
  [2(a)] $(\forall x)F \land G \equiv (\forall x)(F \land G)$
  [2(b)] $(\forall x)F \lor G \equiv (\forall x)(F \lor G)$
  [2(c)] $(\exists x)F \land G \equiv (\exists x)(F \land G)$
  [2(d)] $(\exists x)F \lor G \equiv (\exists x)(F \lor G)$
- Sean F y G fórmulas.
  [3(a)] $(\forall x)F \land (\forall x)G \equiv (\forall x)(F \land G)$
  [3(b)] $(\exists x)F \lor (\exists x)G \equiv (\exists x)(F \lor G)$
- Sean F y G fórmulas.
  [4(a)] $(\forall x)(\forall y)F \equiv (\forall y)(\forall x)F$
  [4(b)] $(\exists x)(\exists y)F \equiv (\exists y)(\exists x)F$

Equivalencia 1(a) $\rightarrow$

$\neg(\forall x)P(x) \vdash (\exists x)\neg P(x)$

1	$\neg(\forall x)P(x)$	premisa
2	$\neg(\exists x)\neg P(x)$	supuesto
3	actual x_0	supuesto
4	$\neg P(x_0)$	supuesto
5	$(\exists x)\neg P(x)$	$\exists$i $4,3$
6	$\bot$	$\neg$e $2,5$
7	$P(x_0)$	RAA $4-6$
8	$(\forall x)P(x)$	$\forall$i $3-7$
9	$\bot$	$\neg$e $1,8$
10	$(\exists x)\neg P(x)$	RAA $2-9$

Equivalencia 1(a) $\leftarrow$

$(\exists x)\neg P(x) \vdash \neg(\forall x)P(x)$

1	$(\exists x)\neg P(x)$	premisa
2	$\neg\neg(\forall x)P(x)$	supuesto
3	actual $x_0, \neg P(x_0)$	supuesto
4	$(\forall x)P(x)$	$\neg\neg$e 2
5	$P(x_0)$	$\forall$e 4
6	$\bot$	$\neg$e $3,5$
7	$\bot$	$\exists$e $1,3-6$
8	$\neg(\forall x)P(x)$	RAA $2-7$

13

Equivalencia 1(a) $\leftrightarrow$

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

1	$\neg(\forall x)P(x)$	supuesto
2	$(\exists x)\neg P(x)$	Lema 1(a) $\rightarrow$
3	$\neg(\forall x)P(x) \rightarrow (\exists x)\neg P(x)$	$\rightarrow$i $1-2$
4	$(\exists x)\neg P(x)$	supuesto
5	$\neg(\forall x)P(x)$	Lema 1(a) $\leftarrow$
6	$(\exists x)\neg P(x) \rightarrow \neg(\forall x)P(x)$	$\rightarrow$i $4-5$
7	$\neg(\forall x)P(x) \leftrightarrow (\exists x)\neg P(x)$	$\leftrightarrow$i $3,6$

14

Equivalencia 3(a) $\to$

$(\forall x)(P(x) \land Q(x)) \vdash (\forall x)P(x) \land (\forall x)Q(x)$

1	$(\forall x)(P(x) \land Q(x))$	premisa
2	actual x_0	supuesto
3	$P(x_0) \land Q(x_0)$	$\forall$e $1,2$
4	$P(x_0)$	$\land$e 3
5	$(\forall x)P(x)$	$\forall$i $2-4$
6	actual x_1	supuesto
7	$P(x_1) \land Q(x_1)$	$\forall$e $1,6$
8	$Q(x_1)$	$\land$e 7
9	$(\forall x)Q(x)$	$\forall$i $6-8$
10	$(\forall x)P(x) \land (\forall x)Q(x)$	$\land$i $5,9$

Equivalencia 3(a) $\leftarrow$

$(\forall x)P(x) \land (\forall x)Q(x) \vdash (\forall x)(P(x) \land Q(x))$

1	$(\forall x)P(x) \land (\forall x)Q(x)$	premisa
2	actual x_0	supuesto
3	$(\forall x)P(x)$	$\land$e 1
4	$P(x_0)$	$\forall$e $3,2$
5	$(\forall x)Q(x)$	$\land$e 1
6	$Q(x_0)$	$\land$e 5
7	$P(x_0) \land Q(x_0)$	$\land$i $4,6$
8	$(\forall x)(P(x) \land Q(x))$	$\forall$i $2-7$

Equivalencia 3(a) $\leftrightarrow$

$(\forall x)(P(x) \wedge Q(x)) \equiv (\forall x)P(x) \wedge (\forall x)Q(x)$

1	$(\forall x)(P(x) \wedge Q(x))$	supuesto
2	$(\forall x)P(x) \wedge (\forall x)Q(x)$	Lema 3(a) $\rightarrow$
3	$(\forall x)(P(x) \wedge Q(x)) \rightarrow (\forall x)P(x) \wedge (\forall x)Q(x)$	$\rightarrow$i $1-2$
4	$(\forall x)P(x) \wedge (\forall x)Q(x)$	supuesto
5	$(\forall x)(P(x) \wedge Q(x))$	Lema 3(a) $\leftarrow$
6	$(\forall x)P(x) \wedge (\forall x)Q(x) \rightarrow (\forall x)(P(x) \wedge Q(x))$	$\rightarrow$i $4-5$
7	$(\forall x)(P(x) \wedge Q(x)) \leftrightarrow (\forall x)P(x) \wedge (\forall x)Q(x)$	$\leftrightarrow$i $3,6$

17

Equivalencia 3(b) $\rightarrow$

$(\exists x)P(x) \vee (\exists x)Q(x) \vdash (\exists x)(P(x) \vee Q(x))$

1	$(\exists x)P(x) \vee (\exists x)Q(x)$	premisa
2	$(\exists x)P(x)$	supuesto
3	actual $x_0, P(x_0)$	supuesto
4	$P(x_0) \vee Q(x_0)$	$\vee$i 3
5	$(\exists x)(P(x) \vee Q(x))$	$\exists$i $4,3$
6	$(\exists x)(P(x) \vee Q(x))$	$\exists$e $2,3-5$
7	$(\exists x)Q(x)$	supuesto
8	actual $x_1, Q(x_1)$	supuesto
9	$P(x_1) \vee Q(x_1)$	$\vee$i 8
10	$(\exists x)(P(x) \vee Q(x))$	$\exists$i $9,8$
11	$(\exists x)(P(x) \vee Q(x))$	$\exists$e $7,8-10$
12	$(\exists x)(P(x) \vee Q(x))$	$\vee$1e $,2-6,7-11$

18

Equivalencia 3(b) $\leftarrow$

$(\exists x)(P(x) \vee Q(x)) \vdash (\exists x)P(x) \vee (\exists x)Q(x)$

1	$(\exists x)(P(x) \vee Q(x))$	premisa
2	actual x_0, $P(x_0) \vee Q(x_0)$	supuesto
3	$P(x_0)$	supuesto
4	$(\exists x)P(x)$	$\exists$i $3, 2$
5	$(\exists x)P(x) \vee (\exists x)Q(x)$	$\vee$i 4
6	$Q(x_0)$	supuesto
7	$(\exists x)Q(x)$	$\exists$i $6, 2$
8	$(\exists x)P(x) \vee (\exists x)Q(x)$	$\vee$i 7
9	$(\exists x)P(x) \vee (\exists x)Q(x)$	$\vee$e $2, 3-5, 6-8$
10	$(\exists x)P(x) \vee (\exists x)Q(x)$	$\exists$e $1, 2-9$

19

Equivalencia 3(b) $\leftrightarrow$

$(\exists x)P(x) \vee (\exists x)Q(x) \equiv (\exists x)(P(x) \vee Q(x))$

1	$(\exists x)P(x) \vee (\exists x)Q(x)$	supuesto
2	$(\exists x)(P(x) \vee Q(x))$	Lema 3(b) $\rightarrow$
3	$(\exists x)P(x) \vee (\exists x)Q(x) \rightarrow (\exists x)(P(x) \vee Q(x))$	$\rightarrow$i $1-2$
4	$(\exists x)(P(x) \vee Q(x))$	supuesto
5	$(\exists x)P(x) \vee (\exists x)Q(x)$	Lema 3(b) $\leftarrow$
6	$(\exists x)(P(x) \vee Q(x)) \rightarrow (\exists x)P(x) \vee (\exists x)Q(x)$	$\rightarrow$i $4-5$
7	$(\exists x)P(x) \vee (\exists x)Q(x) \leftrightarrow (\exists x)(P(x) \vee Q(x))$	$\leftrightarrow$i $3, 6$

20

Equivalencia 4(b) $\longrightarrow$

$(\exists x)(\exists y)P(x,y) \vdash (\exists y)(\exists x)P(x,y)$

1	$(\exists x)(\exists y)P(x,y)$	premisa
2	actual x_0, $(\exists y)P(x_0,y)$	supuesto
3	actual y_0, $P(x_0,y_0)$	supuesto
4	$(\exists x)P(x,y_0)$	$\exists$i $3.2, 2.1$
5	$(\exists y)(\exists x)P(x,y)$	$\exists$i $4, 3.1$
6	$(\exists y)(\exists x)P(x,y)$	$\exists$e $2.2, 3-5$
7	$(\exists y)(\exists x)P(x,y)$	$\exists$e $1, 2-6$

21

Equivalencia 4(b) $\longleftrightarrow$

$(\exists x)(\exists y)P(x,y) \equiv (\exists y)(\exists x)P(x,y)$

1	$(\exists x)(\exists y)P(x,y)$	supuesto
2	$(\exists y)(\exists x)P(x,y)$	Lema 4(b) $\longrightarrow$
3	$(\exists x)(\exists y)P(x,y) \rightarrow (\exists y)(\exists x)P(x,y)$	$\rightarrow$i $1-2$
4	$(\exists y)(\exists x)P(x,y)$	supuesto
5	$(\exists x)(\exists y)P(x,y)$	Lema 4(b) $\longrightarrow$
6	$(\exists y)(\exists x)P(x,y) \rightarrow (\exists x)(\exists y)P(x,y)$	$\rightarrow$i $4-5$
7	$(\exists x)(\exists y)P(x,y) \leftrightarrow (\exists y)(\exists x)P(x,y)$	$\leftrightarrow$i $3, 6$

22

Regla de eliminación de la igualdad

- Regla de eliminación de la igualdad:

  $$\frac{t_1 = t_2 \quad F[x/t_1]}{F[x/t_2]} = e$$

  donde $[x/t_1]$ y $[x/t_2]$ son libres para F.
- Ejemplo:

1	$(x+1) = (1+x)$	premisa
2	$(x+1>1) \rightarrow (x+1>0)$	premisa
3	$(1+x>1) \rightarrow (1+x>0)$	=e $1,2$

- Ejemplo: $t_1 = t_2, t_2 = t_3 \vdash t_1 = t_3$

1	$t_1 = t_2$	premisa
2	$t_2 = t_3$	premisa
3	$t_1 = t_3$	=e $2,1$

23

Regla de introducción de la igualdad

- Regla de introducción de la igualdad:

  $$\frac{}{t = t} = i$$
- Ejemplo: $t_1 = t_2 \vdash t_2 = t_1$

1	$t_1 = t_2$	premisa
2	$t_1 = t_1$	=i
3	$t_2 = t_1$	=e $1,2$

24

Tableros semánticos: Fórmulas gamma y delta

- Las fórmulas gamma, junto con sus componentes, son

$(\forall x)F$	$F[x/t]$	(con t un término básico)
$\neg(\exists x)F$	$\neg F[x/t]$	(con t un término básico)

- Las fórmulas delta, junto con sus componentes, son

$(\exists x)F$	$F[x/a]$	(con a una nueva constante)
$\neg(\forall x)F$	$\neg F[x/a]$	(con a una nueva constante)

Ejemplo de consecuencia mediante tablero semánticos

$$\{(\forall x)[P(x) \to Q(x)], (\exists x)P(x)\} \vdash_{Tab} (\exists x)Q(x)$$

1 $(\forall x)[P(x) \to Q(x)]$
2 $(\exists x)P(x)$
3 $\neg(\exists x)Q(x)$
4 $P(a)$ (2)
5 $P(a) \to Q(a)$ (1)

6 $\neg P(a)$ (5)
Cerrada
(6 y 4)

7 $Q(a)$ (5)
8 $\neg Q(a)$ (3)
Cerrada
(7 y 8)

Ejemplo de consecuencia mediante tableros semánticos

$\{(\forall x)[P(x) \rightarrow Q(x)], (\forall x)[Q(x) \rightarrow R(x)]\} \vdash_{Tab} (\forall x)[P(x) \rightarrow R(x)]$

1 $(\forall x)[P(x) \rightarrow Q(x)]$
2 $(\forall x)[Q(x) \rightarrow R(x)]$
3 $\neg(\forall x)[P(x) \rightarrow R(x)]$
4 $\neg(P(a) \rightarrow R(a))$ (3)
5 $P(a)$ (4)
6 $\neg R(a)$ (4)
7 $P(a) \rightarrow Q(a)$ (1)
8 $Q(a) \rightarrow R(a)$ (2)

9 $\neg P(a)$ (7) 10 $Q(a)$ (7)
Cerrada (5–9)

11 $\neg Q(a)$ (8) 12 $R(a)$ (8)
Cerrada (10–11) Cerrada (6–12)

Ejemplo de no consecuencia mediante tablero

$(\forall x)[P(x) \vee Q(x)] \not\models (\forall x)P(x) \vee (\forall x)Q(x)$

1 $(\forall x)[P(x) \vee Q(x)]$
2 $\neg((\forall x)P(x) \vee (\forall x)Q(x))$
3 $\neg(\forall x)P(x)$ (2)
4 $\neg(\forall x)Q(x)$ (2)
5 $\neg P(a)$ (3)
6 $\neg Q(b)$ (4)
7 $P(a) \vee Q(a)$ (1)
8 $P(b) \vee Q(b)$ (1)

9 $P(a)$ (7) 10 $Q(a)$ (7)
Cerrada (5–9)

11 $P(b)$ (8) 12 $Q(b)$ (8)
Abierta Cerrada (6–12)

Contramodelo: $U = \{a, b\}, I(P) = \{b\}, I(Q) = \{a\}$.

Bibliografía

1. C. Badesa, I. Jané y R. Jansana *Elementos de lógica formal.* (Ariel, 2000) pp. 259–287.
2. R. Bornat *Using ItL Jape with X* (Department of Computer Science, QMW, 1998)
3. J. Dingel *Propositional and predicate logic: a review.* (2000) pp. 28–33.
4. J.L. Fernández, A. Manjarrés y F.J. Díez *Lógica computacional.* (UNED, 2003) pp. 88–94.
5. M. Huth y M. Ryan *Logic in computer science: modelling and reasoning about systems.* (Cambridge University Press, 2000) pp. 109-127.

Capítulo 8

Formas normales. Cláusulas

Lógica informática (2005–06)

Tema 8: Formas normales. Cláusulas

José A. Alonso Jiménez

Grupo de Lógica Computacional
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

1

Equivalencias

- Equivalencia lógica
  - Prop.: $F \equiv G$ syss $\models F \leftrightarrow G$.
- Propiedades básicas de la equivalencia lógica:
  - Reflexiva: $F \equiv F$
  - Simétrica: Si $F \equiv G$, entonces $G \equiv F$
  - Transitiva: Si $F \equiv G$ y $G \equiv H$, entonces $F \equiv H$
- Principio de sustitución de fórmulas equivalentes:
  - Prop.: Si en la fórmula F_1 se sustituye una de sus subfórmulas G_1 por una fórmula G_2 lógicamente equivalente a G_1, entonces la fórmula obtenida, F_2, es lógicamente equivalente a F_1.
  - Ejemplo: $$\begin{aligned} F_1 &= (\forall x)P(x) \to (\exists x)Q(x) \\ G_1 &= (\forall x)P(x) \\ G_2 &= (\forall y)P(y) \\ F_2 &= (\forall y)P(y) \to (\exists x)Q(x) \end{aligned}$$

2

Fórmula en forma rectificada

- Def.: F está en forma rectificada si ninguna variable aparece libre y ligada y cada cuantificador se refiere a una variable diferente.
- Ejemplos:
  $(\forall x)P(x) \to (\forall y)Q(z,y)$ está en forma rectificada
  $(\forall x)P(x) \to (\forall y)Q(x,y)$ no está en forma rectificada
  $(\forall x)P(x) \to (\forall x)Q(z,x)$ no está en forma rectificada
- Prop.: Para toda fórmula F existe una fórmula equivalente G en forma rectificada.
- Lema del renombramiento: Si y no aparece libre en F, entonces

$$(\forall x)F \equiv (\forall y)F[x/y]$$

$$(\exists x)F \equiv (\exists y)F[x/y].$$

- Ejemplos de rectificación:

$$\begin{aligned}(\forall x)P(x) \to (\forall x)Q(z,x) &\equiv (\forall x)P(x) \to (\forall u)Q(z,u)\\ (\forall x)P(x) \to (\forall y)Q(x,y) &\equiv (\forall z)P(z) \to (\forall y)Q(x,y)\end{aligned}$$

3

Fórmula en forma normal prenexa

- Def.: La fórmula F está en forma normal prenexa (FNP) si es de la forma $(Q_1x_1)\dots(Q_nx_n)G$, donde $Q_i \in \{\forall,\exists\}$, $n \geq 0$ y G no tiene cuantificadores. $(Q_1x_1)\dots(Q_nx_n)$ se llama el prefijo de F y G se llama la matriz de F.
- Ejemplos:

Fórmula	¿FNP?
$\neg(\exists x)[P(x) \to (\forall x)P(x)]$	no
$(\forall x)(\exists y)[P(x) \land \neg P(y)]$	sí
$(\forall x)P(x) \lor (\exists y)Q(y)$	no
$(\forall x)(\exists y)[P(x) \lor Q(y)]$	sí
$(\exists y)(\forall x)[P(x) \lor Q(y)]$	sí
$\neg((\forall x)[P(x) \to Q(x)] \land (\forall x)[Q(x) \to R(x)] \to (\forall x)[P(x) \to R(x)])$	no
$(\exists z)(\forall x)(\forall y)[((\neg P(x) \lor Q(x)) \land (\neg Q(y) \lor R(y))) \land P(z)]$	sí

4

Algoritmo de cálculo de forma normal prenexa

Aplicando a una fórmula los siguientes pasos se obtiene otra fórmula equivalente y que está en forma normal prenexa rectificada:

1. Rectificar la fórmula usando las equivalencias
   $(\forall x)F \equiv (\forall y)F[x/y]$ (1)
   $(\exists x)F \equiv (\exists y)F[x/y]$ (2)
   donde y es una variable que no ocurre libre en F.
2. Eliminar los bicondicionales usando la equivalencia
   $F \leftrightarrow G \equiv (F \rightarrow G) \land (G \rightarrow F)$ (3)
3. Eliminar los condicionales usando la equivalencia
   $F \rightarrow G \equiv \neg F \lor G$ (4)

5

Algoritmo de cálculo de forma normal prenexa

4. Interiorizar las negaciones usando las equivalencias
   $\neg(F \land G) \equiv \neg F \lor \neg G$ (5)
   $\neg(F \lor G) \equiv \neg F \land \neg G$ (6)
   $\neg\neg F \equiv F$ (7)
   $\neg(\forall x)F \equiv (\exists x)\neg F$ (8)
   $\neg(\exists x)F \equiv (\forall x)\neg F$ (9)
5. Exteriorizar los cuantificadores usando las equivalencias
   $(\forall x)F \land G \equiv (\forall x)(F \land G)$ con x no libre en G. (11)
   $(\forall x)F \lor G \equiv (\forall x)(F \lor G)$ con x no libre en G. (12)
   $(\exists x)F \land G \equiv (\exists x)(F \land G)$ con x no libre en G. (13)
   $(\exists x)F \lor G \equiv (\exists x)(F \lor G)$ con x no libre en G. (14)
   $G \land (\forall x)F \equiv (\forall x)(G \land F)$ con x no libre en G. (15)
   $G \lor (\forall x)F \equiv (\forall x)(G \lor F)$ con x no libre en G. (16)
   $G \land (\exists x)F \equiv (\exists x)(G \land F)$ con x no libre en G. (17)
   $G \lor (\exists x)F \equiv (\exists x)(G \lor F)$ con x no libre en G. (18)

6

Ejemplos de cálculo de forma normal prenexa

- Ejemplo 1:

$$
\begin{array}{lll}
 & \neg(\exists x)[P(x) \rightarrow (\forall x)P(x)] & \\
\equiv & \neg(\exists x)[P(x) \rightarrow (\forall y)P(y)] & \text{[por (1)]} \\
\equiv & \neg(\exists x)[\neg P(x) \vee (\forall y)P(y)] & \text{[por (4)]} \\
\equiv & (\forall x)[\neg(\neg P(x) \vee (\forall y)P(y))] & \text{[por (9)]} \\
\equiv & (\forall x)[\neg\neg P(x) \wedge \neg(\forall y)P(y)] & \text{[por (6)]} \\
\equiv & (\forall x)[P(x) \wedge (\exists y)\neg P(y)] & \text{[por (7 y 8)]} \\
\equiv & (\forall x)(\exists y)[P(x) \wedge \neg P(y)] & \text{[por (17)]}
\end{array}
$$

- Ejemplo 2:

$$
\begin{array}{lll}
 & (\forall x)P(x) \vee (\exists y)Q(y) & \\
\equiv & (\forall x)[P(x) \vee (\exists y)Q(y)] & \text{[por (12)]} \\
\equiv & (\forall x)(\exists y)[P(x) \vee Q(y)] & \text{[por (18)]}
\end{array}
$$

- Ejemplo 3:

$$
\begin{array}{lll}
 & (\forall x)P(x) \vee (\exists y)Q(y) & \\
\equiv & (\exists y)[(\forall x)P(x) \vee Q(y)] & \text{[por (18)]} \\
\equiv & (\exists y)(\forall x)[P(x) \vee Q(y)] & \text{[por (12)]}
\end{array}
$$

7

Ejemplos de cálculo de forma normal prenexa

- Ejemplo de cálculo de una forma normal prenexa de

$$
\begin{array}{lll}
 & \neg((\forall x)[P(x) \rightarrow Q(x)] \wedge (\forall x)[Q(x) \rightarrow R(x)] \rightarrow (\forall x)[P(x) \rightarrow R(x)]) & \\
\equiv & \neg((\forall x)[P(x) \rightarrow Q(x)] \wedge (\forall y)[Q(y) \rightarrow R(y)] \rightarrow (\forall z)[P(z) \rightarrow R(z)]) & \text{[por (1)]} \\
\equiv & \neg(\neg((\forall x)[\neg P(x) \vee Q(x)] \wedge (\forall y)[\neg Q(y) \vee R(y)]) \vee (\forall z)[\neg P(z) \vee R(z)]) & \text{[por (4)]} \\
\equiv & \neg\neg((\forall x)[\neg P(x) \vee Q(x)] \wedge (\forall y)[\neg Q(y) \vee R(y)]) \wedge \neg(\forall z)[\neg P(z) \vee R(z)] & \text{[por (6)]} \\
\equiv & ((\forall x)[\neg P(x) \vee Q(x)] \wedge (\forall y)[\neg Q(y) \vee R(y)]) \wedge (\exists z)[\neg(\neg P(z) \vee R(z))] & \text{[por (7, 8)]} \\
\equiv & ((\forall x)[\neg P(x) \vee Q(x)] \wedge (\forall y)[\neg Q(y) \vee R(y)]) \wedge (\exists z)[\neg\neg P(z) \wedge \neg R(z)] & \text{[por (6)]} \\
\equiv & ((\forall x)[\neg P(x) \vee Q(x)] \wedge (\forall y)[\neg Q(y) \vee R(y)]) \wedge (\exists z)[P(z) \wedge \neg R(z)] & \text{[por (7)]} \\
\equiv & (\exists z)[((\forall x)[\neg P(x) \vee Q(x)] \wedge (\forall y)[\neg Q(y) \vee R(y)]) \wedge (P(z) \wedge \neg R(z))] & \text{[por (17)]} \\
\equiv & (\exists z)[(\forall x)[(\neg P(x) \vee Q(x)) \wedge (\forall y)[\neg Q(y) \vee R(y)]] \wedge (P(z) \wedge \neg R(z))] & \text{[por (11)]} \\
\equiv & (\exists z)(\forall x)[((\neg P(x) \vee Q(x)) \wedge (\forall y)[\neg Q(y) \vee R(y)]) \wedge (P(z) \wedge \neg R(z))] & \text{[por (11)]} \\
\equiv & (\exists z)(\forall x)[(\forall y)[(\neg P(x) \vee Q(x)) \wedge (\neg Q(y) \vee R(y))] \wedge (P(z) \wedge \neg R(z))] & \text{[por (15)]} \\
\equiv & (\exists z)(\forall x)(\forall y)[((\neg P(x) \vee Q(x)) \wedge (\neg Q(y) \vee R(y))) \wedge (P(z) \wedge \neg R(z))] & \text{[por (11)]}
\end{array}
$$

8

Fórmula en forma normal prenexa conjuntiva

- Def.: La fórmula F está en forma normal prenexa conjuntiva (FNPC) si es de la forma $(Q_1x_1)\dots(Q_nx_n)G$, donde $Q_i \in \{\forall,\exists\}$, $n \geq 0$, G no tiene cuantificadores y G está en forma normal conjuntiva.
- Algoritmo de cálculo de forma normal prenexa conjuntiva:
  - Algoritmo: Aplicando a una fórmula los siguientes pasos se obtiene otra fórmula equivalente y que está en forma normal prenexa conjuntiva rectificada:
    1. Calcular una forma normal prenexa rectificada usando las equivalencias (1)–(18)
    2. Interiorizar las disyunciones usando las equivalencias
       $A \lor (B \land C) \equiv (A \lor B) \land (A \lor C)$ (19)
       $(A \land B) \lor C \equiv (A \lor C) \land (B \lor C)$ (20)
  - Ejemplo de cálculo de una FNPC de $(\forall x)(\exists y)[P(x) \lor (Q(y) \land \neg R(y))]$:

$$\begin{array}{ll} & (\forall x)(\exists y)[P(x) \lor (Q(y) \land \neg R(y))] \\ \equiv & (\forall x)(\exists y)[(P(x) \lor Q(y)) \land (P(x) \lor \neg R(y))] \quad \text{[por (19)]} \end{array}$$

Fórmula en forma de Skolem

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

- Equisatisfacibilidad:
  - Def.: Las fórmulas F y G son equisatisfacible si:
    F es satisfacible syss G es satisfacible.
    Se representa por $F \equiv_{sat} G$
  - Ejemplos:

$(\exists x)Q(x)$	$\equiv_{sat}$	$Q(a)$
$(\exists x)Q(x)$	$\not\equiv$	$Q(a)$
$(\forall x)(\exists y)P(x,y)$	$\equiv_{sat}$	$(\forall x)P(x,f(x))$
$(\forall x)(\exists y)P(x,y)$	$\not\equiv$	$(\forall x)P(x,f(x))$

Algoritmo de cálculo de forma de Skolem

- Propiedades:
  - Si a es una constante que no ocurre en F, entonces $(\exists x)F \equiv_{sat} F[x/a]$.
  - Si g es un símbolo de función n–aria que no ocurre en F, entonces $(\forall x_1)\dots(\forall x_n)(\exists x)F \equiv_{sat} (\forall x_1)\dots(\forall x_n)F[x/g(x_1,\dots,x_n)]$.
- Algoritmo de cálculo de forma de Skolem:
  - Sea F una fórmula en forma normal prenexa rectificada, la forma de Skolem de F es

$$\text{Sko}(F) = \begin{cases} \text{Sko}(G[x/a]), & \text{si } F \text{ es } (\exists x)G \text{ y} \\ & a \text{ es una nueva constante;} \\ \text{Sko}((\forall x_1)\dots(\forall x_n)G[x/f(x_1,\dots,x_n)]), & \text{si } F \text{ es } (\forall x_1)\dots(\forall x_n)(\exists x)G \text{ y} \\ & f \text{ es un nuevo símbolo de función;} \\ F, & \text{si } F \text{ está en forma de Skolem} \end{cases}$$

  - Propiedad: Si F es una fórmula en forma normal prenexa rectificada, entonces $\text{Sko}(F)$ está en forma de Skolem y $\text{Sko}(F) \equiv_{sat} F$.

11

Ejemplos de cálculo de forma de Skolem

- Ejemplo 1:

$$\begin{aligned} & \text{Sko}((\exists x)(\forall y)(\forall z)(\exists u)(\forall v)(\exists w)P(x,y,z,u,v,w)) \\ & = \text{Sko}((\forall y)(\forall z)(\exists u)(\forall v)(\exists w)P(a,y,z,u,v,w)) \\ & = \text{Sko}((\forall y)(\forall z)(\forall v)(\exists w)P(a,y,z,f(y,z),v,w)) \\ & = \text{Sko}((\forall y)(\forall z)(\forall v)P(a,y,z,f(y,z),v,g(y,z,v))) \\ & = (\forall y)(\forall z)(\forall v)P(a,y,z,f(y,z),v,g(y,z,v)) \end{aligned}$$

- Ejemplo 2:

$$\begin{aligned} & \text{Sko}((\forall x)(\exists y)(\forall z)(\exists w)[\neg P(a,w) \lor Q(f(x),y)]) \\ & = \text{Sko}((\forall x)(\forall z)(\exists w)[\neg P(a,w) \lor Q(f(x),h(x))]) \\ & = \text{Sko}((\forall x)(\forall z)[\neg P(a,g(x,z)) \lor Q(f(x),h(x))]) \\ & = (\forall x)(\forall z)[\neg P(a,g(x,z)) \lor Q(f(x),h(x))] \end{aligned}$$

12

Ejemplos de cálculo de forma de Skolem

- Ejemplo de cálculo de una forma de Skolem de

$$\begin{array}{ll} & \neg(\exists x)[P(x) \rightarrow (\forall x)P(x)] \\ \equiv & (\forall x)(\exists y)[P(x) \wedge \neg P(y)] \quad \text{[por página 7]} \\ \equiv_{sat} & (\forall x)[P(x) \wedge \neg P(f(x))] \end{array}$$

- Ejemplo de cálculo de una forma de Skolem de

$$\begin{array}{ll} & (\forall x)P(x) \vee (\exists y)Q(y) \\ \equiv & (\forall x)(\exists y)[P(x) \vee Q(y)] \quad \text{[por página 7]} \\ \equiv_{sat} & (\forall x)[P(x) \vee Q(f(x))] \end{array}$$

- Ejemplo de cálculo de otra forma de Skolem de

$$\begin{array}{ll} & (\forall x)P(x) \vee (\exists y)Q(y) \\ \equiv & (\exists y)(\forall x)[P(x) \vee Q(y)] \quad \text{[por página 7]} \\ \equiv_{sat} & (\forall x)[P(x) \vee Q(a)] \end{array}$$

- Ejemplo de cálculo de una forma de Skolem de

$$\begin{array}{ll} & \neg((\forall x)[P(x) \rightarrow Q(x)] \wedge (\forall x)[Q(x) \rightarrow R(x)] \rightarrow (\forall x)[P(x) \rightarrow R(x)]) \\ \equiv & (\exists z)(\forall x)(\forall y)[((\neg P(x) \vee Q(x)) \wedge (\neg Q(y) \vee R(y))) \wedge (P(z) \wedge \neg R(z))] \quad \text{[por p. 8]} \\ \equiv_{sat} & (\forall x)(\forall y)[((\neg P(x) \vee Q(x)) \wedge (\neg Q(y) \vee R(y))) \wedge (P(a) \wedge \neg R(a))] \end{array}$$

Sintaxis de la lógica clausal

- Un átomo es una fórnula atómica.
  Variables sobre átomos: $A, B, C, \dots, A_1, A_2, \dots$.
- Un literal es un átomo (A) o la negación de un átomo ($\neg A$).
  Variables sobre literales: $L, L_1, L_2, \dots$.
- Una cláusula es un conjunto finito de literales.
  Variables sobre cláusulas: $C, C_1, C_2, \dots$.
- La cláusula vacía es el conjunto vacío de literales.
  La cláusula vacía se representa por $\square$.
- Conjuntos finitos de cláusulas.
  Variables sobre conjuntos finitos de cláusulas: $S, S_1, S_2, \dots$.

Semántica de la lógica clausal

- Fórmulas correspondientes:
  - Def.: La fórmula correspondiente a la cláusula $\{L_1,\dots,L_n\}$ es
    $(\forall x_1)\dots(\forall x_p)[L_1 \vee \dots \vee L_n]$,
    donde $x_1,\dots,x_p$ son las variables libres de $L_1 \vee \dots \vee L_n$.
  - Def.: La fórmula correspondiente a la cláusula $\square$ es $\bot$.
  - Def.: La fórmula correspondiente al conjunto de cláusulas $\{\{L_1^1,\dots,L_{n_1}^1\},\dots,\{L_1^m,\dots,L_{n_m}^m\}\}$ es
    $(\forall x_1)\dots(\forall x_p)[(L_1^1 \vee \dots \vee L_{n_1}^1) \wedge \dots \wedge (L_1^m \vee \dots \vee L_{n_m}^m)]$,
    donde $x_1,\dots,x_p$ son las variables libres de
    $(L_1^1 \vee \dots \vee L_{n_1}^1) \wedge \dots \wedge (L_1^m \vee \dots \vee L_{n_m}^m)$.
  - Def.: La fórmula correspondiente al conjunto de cláusulas $\emptyset$ es $\top$.
- Semántica:
  - Def.: En cualquier interpretación $\mathscr{I} = (U,I)$, $I(\top) = 1$ e $I(\bot) = 0$.
  - Def.: Los conceptos semánticos relativos a las cláusulas y a los conjuntos de cláusulas son los de sus correspondientes fórmulas.

Forma clausal de una fórmula

- Def.: Una forma clausal de una fórmula F es un conjunto de cláusulas S tal que $F \equiv_{sat} S$.
- Algoritmo: Aplicando a la fórmula F los siguientes pasos se obtiene S que es una forma clausal de F:
  1. Sea $F_1 = (\exists y_1)\dots(\exists y_n)F$, donde $y_1,\dots,y_n$ son las variables libres de F.
  2. Sea F_2 una forma normal prenexa conjuntiva rectificada de F_1 calculada mediante el algoritmo de la página 9.
  3. Sea $F_3 = \text{Sko}(F_2)$, que tiene la forma
     $(\forall x_1)\dots(\forall x_p)[(L_1^1 \vee \dots \vee L_{n_1}^1) \wedge \dots \wedge (L_1^m \vee \dots \vee L_{n_m}^m)]$,
  4. Sea $S = \{\{L_1^1,\dots,L_{n_1}^1\},\dots,\{L_1^m,\dots,L_{n_m}^m\}\}$.
- Prop.: $F \equiv_{sat} F_1 \equiv F_2 \equiv_{sat} F_3 \equiv S$.

Ejemplos de cálculo de forma clausal de una fórmula

- Ejemplo de cálculo de una forma clausal de

$$\begin{array}{ll} & \neg(\exists x)[P(x) \rightarrow (\forall x)P(x)] \\ \equiv_{sat} & (\forall x)[P(x) \land \neg P(f(x))] \qquad \text{[pág. 13]} \\ \equiv & \{\{P(x)\},\{\neg P(f(x))\}\} \end{array}$$

- Ejemplo de cálculo de una forma clausal de

$$\begin{array}{ll} & (\forall x)P(x) \lor (\exists y)Q(y) \\ \equiv_{sat} & (\forall x)[P(x) \lor Q(f(x))] \quad \text{[pág. 13]} \\ \equiv & \{\{P(x),Q(f(x))\}\} \end{array}$$

- Ejemplo de cálculo de otra forma clausal de

$$\begin{array}{ll} & (\forall x)P(x) \lor (\exists y)Q(y) \\ \equiv_{sat} & (\forall x)[P(x) \lor Q(a)] \qquad \text{[pág. 13]} \\ \equiv & \{\{P(x),Q(a)\}\} \end{array}$$

- Ejemplo de cálculo de una forma clausal de

$$\begin{array}{ll} & \neg((\forall x)[P(x) \rightarrow Q(x)] \land (\forall x)[Q(x) \rightarrow R(x)] \rightarrow (\forall x)[P(x) \rightarrow R(x)]) \\ \equiv_{sat} & (\forall x)(\forall y)[((\neg P(x) \lor Q(x)) \land (\neg Q(y) \lor R(y))) \land (P(a) \land \neg R(a))] \qquad \text{[pág. 13]} \\ \equiv & \{\{\neg P(x),Q(x)\},\{\neg Q(y),R(y)\},\{P(a)\},\{\neg R(a)\}\} \end{array}$$

17

Ejemplos de cálculo de forma clausal de una fórmula

$$\begin{array}{lll} & \neg((\forall x)[P(x) \rightarrow Q(x)] \land (\exists x)P(x) \rightarrow (\exists x)Q(x)) & \\ \equiv & \neg((\forall x)[P(x) \rightarrow Q(x)] \land (\exists y)P(y) \rightarrow (\exists z)Q(z)) & [(2)] \\ \equiv & \neg(\neg((\forall x)[P(x) \rightarrow Q(x)] \land (\exists y)P(y)) \lor (\exists z)Q(z)) & [(4)] \\ \equiv & \neg(\neg((\forall x)[\neg P(x) \lor Q(x)] \land (\exists y)P(y)) \lor (\exists z)Q(z)) & [(4)] \\ \equiv & \neg\neg((\forall x)[\neg P(x) \lor Q(x)] \land (\exists y)P(y)) \land \neg(\exists z)Q(z) & [(6)] \\ \equiv & ((\forall x)[\neg P(x) \lor Q(x)] \land (\exists y)P(y)) \land \neg(\exists z)Q(z) & [(7)] \\ \equiv & ((\forall x)[\neg P(x) \lor Q(x)] \land (\exists y)P(y)) \land (\forall z)\neg Q(z) & [(9)] \\ \equiv & (\exists y)[(\forall x)[\neg P(x) \lor Q(x)] \land P(y)] \land (\forall z)\neg Q(z) & [(17)] \\ \equiv & (\exists y)[((\forall x)[\neg P(x) \lor Q(x)] \land P(y)) \land (\forall z)\neg Q(z)] & [(13)] \\ \equiv & (\exists y)[(\forall x)[(\neg P(x) \lor Q(x)) \land P(y))] \land (\forall z)\neg Q(z) & [(11)] \\ \equiv & (\exists y)(\forall x)[((\neg P(x) \lor Q(x)) \land P(y)) \land (\forall z)\neg Q(z)] & [(11)] \\ \equiv & (\exists y)(\forall x)(\forall z)[(\neg P(x) \lor Q(x)) \land P(y) \land \neg Q(z)] & [(15)] \\ \equiv_{sat} & (\forall x)(\forall z)[(\neg P(x) \lor Q(x) \land P(a)) \land \neg Q(z)] & [(15)] \\ \equiv & \{\{\neg P(x),Q(x)\},\{P(a)\},\{\neg Q(z)\}\} & \end{array}$$

18

Forma clausal de un conjunto de fórmulas

- Equisatisfacibilidad de conjuntos de fórmulas:
  - Def.: Los conjuntos de fórmulas S_1 y S_2 son equisatisfacible si:
    S_1 es satisfacible syss S_2 es satisfacible.
    Se representa por $S_1 \equiv_{sat} S_2$
- Forma clausal de un conjunto de fórmulas:
  - Def.: Una forma clausal de un conjunto de fórmulas S es un conjunto de cláusulas equisatisfacible con S.
  - Prop.: Si $S_1,\dots,S_n$ son formas clausales de $F_1,\dots,F_n$, entonces $S_1 \cup \dots \cup S_n$ es una forma clausal de $\{F_1,\dots,F_n\}$.
  - Ejemplo: Una forma clausal de
    $\{(\forall x)[P(x) \rightarrow Q(x)], (\exists x)P(x), \neg(\exists x)Q(x)\}$
    es
    $\{\{\neg P(x), Q(x)\}, \{P(a)\}, \{\neg Q(z)\}\}$.

Consecuencia e inconsistencia de cláusulas

- Prop: Sean $S_1,\dots,S_n$ formas clausales de las fórmulas $F_1,\dots,F_n$ y S una forma clausal de $\neg G$. Son equivalentes:
  1. $\{F_1,\dots,F_n\} \models G$.
  2. $\{F_1,\dots,F_n,\neg G\}$ es inconsistente.
  3. $S_1 \cup \dots \cup S_n \cup S$ es inconsistente.
- Ejemplos:
  - Ejemplo 1:
    $\{(\forall x)[P(x) \rightarrow Q(x)], (\exists x)P(x)\} \models (\exists x)Q(x)$
    syss $\{\{\neg P(x), Q(x)\}, \{P(a)\}, \{\neg Q(z)\}\}$ es inconsistente.
  - Ejemplo 2:
    $\{(\forall x)[P(x) \rightarrow Q(x)], (\forall x)[Q(x) \rightarrow R(x)]\} \models (\forall x)[P(x) \rightarrow R(x)]$
    syss $\{\{\neg P(x), Q(x)\}, \{\neg Q(y), R(y)\}, \{P(a)\}, \{\neg R(a)\}\}$ es inconsistente.

Bibliografía

1. M.L. Bonet *Apuntes de LPO.* (Univ. Politécnica de Cataluña, 2003) pp. 26–31.
2. C.L. Chang y R.C.T. Lee *Symbolic logic and mechanical theorem proving* (Academic Press, 1973) pp. 35–39 y 46–51.
3. J.L. Fernández, A. Manjarrés y F.J. Díez *Lógica computacional.* (UNED, 2003) pp. 101–106.
4. S. Hölldobler *Computational logic.* (U. de Dresden, 2004) pp. 62–67.
5. R. Nieuwenhuis *Lógica de primer orden.* (U. Politénica de Cataluña, 2003) pp. 16–17
6. M. Ojeda e I. Pérez *Lógica para la computación (Vol. 2: Lógica de Primer Orden)* (Ágora, 1997) pp. 37–49
7. E. Paniagua, J.L. Sánchez y F. Martín *Lógica computacional* (Thomson, 2003) pp. 153–160.
8. L. Paulson *Logic and proof* (U. Cambridge, 2002) pp. 43–47.
9. U. Schöning *Logic for computer scientists* (Birkäuser, 1989) pp. 51–61.

Capítulo 9

Modelos de Herbrand

Lógica informática (2005–06)
Tema 9: Modelos de Herbrand

José A. Alonso Jiménez

Grupo de Lógica Computacional
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

1

Reducción de la LPO básica a proposicional

- Observación:
  - En este tema sólo se consideran lenguajes de primer orden sin igualdad.
- Reducción de la LPO básica a proposicional
  - Def.: Una fórmula básica es una fórmula sin variables ni cuantificadores.
  - Prop.: Sea S un conjunto de fórmulas básicas. Son equivalentes:
    1. S es consistente en el sentido de la lógica de primer orden.
    2. S es consistente en el sentido de la lógica proposicional.

2

Ejemplos de reducción de la LPO básica a proposicional

- $\{P(a) \lor P(b), \neg P(b) \lor P(c), P(a) \to P(c)\}$
  es consistente en el sentido de la lógica de primer orden
  (con modelos $\mathscr{I}_4, \mathscr{I}_6, \mathscr{I}_8$).
- $\{P(a) \lor P(b), \neg P(b) \lor P(c), P(a) \to P(c), \neg P(c)\}$
  es inconsistente en el sentido de la lógica de primer orden.

	P^I	$P(a) \lor P(b)$	$\neg P(b) \lor P(c)$	$P(a) \to P(c)$	$\neg P(c)$
$\mathscr{I}_1$	$\emptyset$	0	1	1	1
$\mathscr{I}_2$	$\{c^I\}$	0	1	1	0
$\mathscr{I}_3$	$\{b^I\}$	1	0	1	1
$\mathscr{I}_4$	$\{b^I, c^I\}$	1	1	1	0
$\mathscr{I}_5$	$\{a^I\}$	1	1	0	1
$\mathscr{I}_6$	$\{a^I, c^I\}$	1	1	1	0
$\mathscr{I}_7$	$\{a^I, b^I\}$	1	0	0	1
$\mathscr{I}_8$	$\{a^I, b^I, c^I\}$	1	1	1	0

3

Ejemplos de reducción de la LPO básica a proposicional

- $\{P(a) \lor P(b), \neg P(b) \lor P(c), P(a) \to P(c)\}$
  es consistente en el sentido proposicional (con modelos v_4, v_6, v_8).
- $\{P(a) \lor P(b), \neg P(b) \lor P(c), P(a) \to P(c), \neg P(c)\}$
  es inconsistente en el sentido proposicional.
  Se consideran los cambios $P(a)/p, P(b)/q, P(c)/r$

	p	q	r	$p \lor q$	$\neg q \lor r$	$p \to r$	$\neg r$
v_1	0	0	0	0	1	1	1
v_2	0	0	1	0	1	1	0
v_3	0	1	0	1	0	1	1
v_4	0	1	1	1	1	1	0
v_5	1	0	0	1	1	0	1
v_6	1	0	1	1	1	1	0
v_7	1	1	0	1	0	0	1
v_8	1	1	1	1	1	1	0

4

Notación

- L representa un lenguaje de primer orden sin igualdad.
- $\mathscr{C}$ es el conjunto de constantes de L.
- $\mathscr{F}$ es el conjunto de símbolos de función de L.
- $\mathscr{R}$ es el conjunto de símbolos de relación de L.
- $\mathscr{F}_n$ es el conjunto de símbolos de función n–aria de L.
- $\mathscr{R}_n$ es el conjunto de símbolos de relación n–aria de L.
- f/n indica que f es un símbolo de función n–aria de L.
- P/n indica que f es un símbolo de relación n–aria de L.

5

Universo de Herbrand

- Def.: El universo de Herbrand de L es el conjunto de los términos básicos de L. Se representa por $\text{UH}(L)$.
- Prop.: $\text{UH}(L) = \bigcup_{i \geq 0} H_i(L)$, donde $H_i(L)$ es el nivel i del $\text{UH}(L)$ definido por

$$\begin{array}{ll} H_0(L) & = \begin{cases} \mathscr{C}, & \text{si } \mathscr{C} \neq \emptyset; \\ \{a\}, & \text{en caso contrario. (} a \text{ es una nueva constante).} \end{cases} \\ H_{i+1}(L) & = H_i(L) \cup \{ f(t_1, \dots, t_n) : f \in \mathscr{F}_n \text{ y } t_1, \dots, t_n \in H_i(L) \} \end{array}$$

- Prop.: $\text{UH}(L)$ es finito syss L no tiene símbolos de función.

6

Ejemplos de universo de Herbrand

- Si $\mathscr{C} = \{a,b,c\}$ y $\mathscr{F} = \emptyset$, entonces

$$\begin{aligned} H_0(L) &= \{a,b,c\} \\ H_1(L) &= \{a,b,c\} \\ &\vdots \\ \mathrm{UH}(L) &= \{a,b,c\} \end{aligned}$$

- Si $\mathscr{C} = \emptyset$ y $\mathscr{F} = \{f/1\}$, entonces

$$\begin{aligned} H_0(L) &= \{a\} \\ H_1(L) &= \{a, f(a)\} \\ H_2(L) &= \{a, f(a), f(f(a))\} \\ &\vdots \\ \mathrm{UH}(L) &= \{a, f(a), f(f(a)), \dots\} = \{f^i(a) : i \in \mathbb{N}\} \end{aligned}$$

Ejemplos de universo de Herbrand

- Si $\mathscr{C} = \{a,b\}$ y $\mathscr{F} = \{f/1, g/1\}$, entonces

$$\begin{aligned} H_0(L) &= \{a,b\} \\ H_1(L) &= \{a,b, f(a), f(b), g(a), g(b)\} \\ H_2(L) &= \{a,b, f(a), f(b), g(a), g(b), \quad f(f(a)), f(f(b)), f(g(a)), f(g(b)), \\ &\qquad g(f(a)), g(f(b)), g(g(a)), g(g(b))\ \} \\ &\vdots \end{aligned}$$

- Si $\mathscr{C} = \{a,b\}$ y $\mathscr{F} = \{f/2\}$, entonces

$$\begin{aligned} H_0(L) &= \{a,b\} \\ H_1(L) &= \{a,b, f(a,a), f(a,b), f(b,a), f(b,b)\} \\ H_2(L) &= \{a,b, f(a,a), f(a,b), f(b,a), f(b,b), f(a, f(a,a)), f(a, f(a,b)), \dots\} \\ &\vdots \end{aligned}$$

Base de Herbrand

- Def.: La base de Herbrand de L es el conjunto de los átomos básicos de L. Se representa por $\text{BH}(L)$.
- Prop.: $\text{BH}(L) = \{P(t_1,\dots,t_n) : P \in \mathscr{R}_n \text{ y } t_1,\dots,t_n \in \text{UH}(L)\}$.
- Prop.: $\text{BH}(L)$ es finita syss L no tiene símbolos de función.
- Ejemplos:
  - Si $\mathscr{C} = \{a,b,c\}$, $\mathscr{F} = \emptyset$ y $\{\mathscr{R} = P/1\}$, entonces

$$\begin{aligned} \text{UH}(L) &= \{a,b,c\} \\ \text{BH}(L) &= \{P(a),P(b),P(c)\} \end{aligned}$$

  - Si $\mathscr{C} = \{a\}$, $\mathscr{F} = \{f/1\}$ y $\mathscr{R} = \{P/1,Q/1,R/1\}$, entonces

$$\begin{aligned} \text{UH}(L) &= \{a, f(a), f(f(a)),\dots\} = \{f^i(a) : i \in \mathbb{N}\} \\ \text{BH}(L) &= \{P(a),Q(a),R(a),P(f(a)),Q(f(a)),R(f(a)),\dots\} \end{aligned}$$

9

Interpretaciones de Herbrand

- Def.: Una interpretación de Herbrand es una interpretación $\mathscr{I} = (U,I)$ tal que
  - U es el universo de Herbrand de L;
  - $I(c) = c$, para cada constante c de L;
  - $I(f) = f$, para cada símbolo de función f de L.
- Prop.: Sea $\mathscr{I}$ una interpretación de Herbrand de L. Si t es un término básico de L, entonces $\mathscr{I}(t) = t$.
- Prop.: Una interpretación de Herbrand queda determinada por un subconjunto de la base de Herbrand, el conjunto de átomos básicos verdaderos en esa interpretación.

10

Modelos de Herbrand

- Nota: Las definiciones de universo de Herbrand, base de Herbrand e interpretación de Herbrand definidas para un lenguaje se extienden a fórmulas y conjuntos de fórmulas considerando el lenguaje formado por los símbolos no lógicos que aparecen.
- Def.: Un modelo de Herbrand de una fórmula F es una interpretación de Herbrand de F que es modelo de F.
- Def.: Un modelo de Herbrand de un conjunto de fórmulas S es una interpretación de Herbrand de S que es modelo de S.
- Ejemplo: Los modelos de Herbrand de $\{P(a) \lor P(b), \neg P(b) \lor P(c), P(a) \to P(c)\}$ son $\{P(b), P(c)\}, \{P(a), P(c)\}$ y $\{P(a), P(b), P(c)\}$ (ver página 3).
- Ejemplo: Sea $S = \{(\forall x)(\forall y)[Q(b,x) \to P(a) \lor R(y)], P(b) \to \neg(\exists z)(\exists u)Q(z,u)\}$. Entonces,
  $$\begin{aligned} \text{UH}(S) &= \{a,b\} \\ \text{BH}(S) &= \{P(a), P(b), Q(a,a), Q(a,b), Q(b.a), Q(b,b), R(a), R(b)\} \end{aligned}$$
  Un modelo de Herbrand de S es $\{P(a)\}$.

11

Interpretación de Herbrand correspondiente

- Sea $S = \{\{\neg Q(b,x), P(a), R(y)\}, \{\neg P(b), \neg Q(z,u)\}\}$ e $\mathscr{I} = (\{1,2\}, I)$ con $a^I = 1, b^I = 2, P^I = \{1\}, Q^I = \{(1,1),(2,2)\}, R^I = \{2\}$. Entonces, $\mathscr{I} \models S$.
  Cálculo de la interpretación de Herbrand $\mathscr{I}^*$ correspondiente a $\mathscr{I}$:
  $$\begin{aligned} \mathscr{I}^* &= (\text{UH}(S), I^*) \\ \text{UH}(S) &= \{a,b\} \\ \text{BH}(S) &= \{P(a), P(b), Q(a,a), Q(a,b), Q(b.a), Q(b,b), R(a), R(b)\} \\ I^*(P(a)) &= P^I(a^I) = P^I(1) = \text{V} \\ I^*(P(b)) &= P^I(b^I) = P^I(2) = \text{F} \\ I^*(Q(a,a)) &= Q^I(a^I, a^I) = Q^I(1,1) = \text{V} \\ I^*(Q(a,b)) &= Q^I(a^I, b^I) = Q^I(1,2) = \text{F} \\ I^*(Q(b,a)) &= Q^I(b^I, a^I) = Q^I(2,1) = \text{F} \\ I^*(Q(b,b)) &= Q^I(b^I, b^I) = Q^I(2,2) = \text{V} \\ I^*(R(a))) &= R^I(a^I) = R^I(1) = \text{F} \\ I^*(R(b)) &= R^I(b^I) = R^I(2) = \text{V} \end{aligned}$$
  $I^* = \{P(a), Q(a,a), Q(b,b), R(b)\}$ y $\mathscr{I}^* \models S$.

12

Interpretación de Herbrand correspondiente

- Sea S el conjunto de cláusulas $\{\{P(a)\},\{Q(y,f(a))\}\}$ e $\mathscr{I} = (\{1,2\},I)$ con $a^I = 1, f^I = \{(1,2),(2,1)\}, P^I = \{1\}, Q^I = \{(1,2),(2,2)\}$. Entonces, $\mathscr{I} \models S$.
  Cálculo de la interpretación de Herbrand $\mathscr{I}^*$ correspondiente a $\mathscr{I}$:

$$\begin{array}{ll}
\mathscr{I}^* & = (\mathsf{UH}(S), I^*) \\
\mathsf{UH}(S) & = \{a, f(a), f(f(a)), \dots\} = \{f^i(a) : i \in \mathbb{N}\} \\
\mathsf{BH}(S) & = \{P(f^n(a)) : n \in \mathbb{N}\} \cup \{Q(f^n(a), f^m(a)) : n,m \in \mathbb{N}\} \\
I^*(P(a)) & = P^I(a^I) = P^I(1) = \mathsf{V} \\
I^*(P(f(a))) & = P^I(f^I(a^I)) = P^I(f^I(1)) = P^I(2) = \mathsf{F} \\
I^*(P(f(f(a)))) & = P^I(f^I(f^(a^I))) = P^I(1) = \mathsf{V} \\
I^*(P(f^n(a))) & = \begin{cases} \mathsf{V}, & \text{si } n \text{ es par;} \\ \mathsf{F}, & \text{en caso contrario.} \end{cases} \\
I^*(Q(f^n(a), f^m(a))) & = \begin{cases} \mathsf{V}, & \text{si } m \text{ es impar;} \\ \mathsf{F}, & \text{en caso contrario.} \end{cases}
\end{array}$$

  $I^* = \{P(f^{2n}(a)) : n \in \mathbb{N}\} \cup \{Q(f^n(a), f^{2m+1}(a)) : n,m \in \mathbb{N}\}$
  $\mathscr{I}^* \models S$.

13

Consistencia mediante modelos de Herbrand

- Prop.: Sea S un conjunto de fórmulas básicas. Son equivalentes:
  1. S es consistente.
  2. S tiene un modelo de Herbrand.
- Prop.: Sea S un conjunto de cláusulas. Si $\mathscr{I}^*$ es una interpretación de Herbrand correspondiente a un modelo $\mathscr{I}$ de S, entonces $\mathscr{I}^*$ es un modelo de S.
- Prop.: Sea S un conjunto de cláusulas. Son equivalentes:
  1. S es consistente.
  2. S tiene un modelo de Herbrand.
- Prop.: Sea S un conjunto de cláusulas. Son equivalentes:
  1. S es inconsistente.
  2. S no tiene ningún modelo de Herbrand.
- Prop.: Existen conjuntos de fórmulas consistentes sin modelos de Herbrand.

14

Ejemplo de consistente sin modelos de Herbrand

- Sea $S = \{(\exists x)P(x), \neg P(a)\}$. Entonces,
  - S es consistente.
    $\mathscr{I} \models S$ con $\mathscr{I} = (\{1,2\}, I)$, $a^I = 1$ y $P^I = \{2\}$.
  - S no tiene modelos de Herbrand
    $\text{UH}(S) = \{a\}$
    $\text{BH}(S) = \{P(a)\}$
    Las interpretaciones de Herbrand de S son $\emptyset$ y $\{P(a)\}$.
    $\emptyset \not\models S$
    $\{P(a)\} \not\models S$

Instancias básicas de una cláusula

- Def.: Una sustitución σ (de L) es una aplicación σ : Var $\rightarrow$ Térm(L).
- Def.: Sea $C = \{L_1, \dots, L_n\}$ una cláusula de L y σ una sustitución de L. Entonces, $C\sigma = \{L_1\sigma, \dots, L_n\sigma\}$ es una instancia de C.
- Ejemplo: Sea $C = \{P(x,a), \neg P(x, f(y))\}$.
  $C[x/a, y/f(a)] = \{P(f(a),a), \neg P(f(a), f(f(a)))\}$
- Def.: $C\sigma$ es una instancia básica de C si todos los literales de $C\sigma$ son básicos.
- Ejemplo: Sea $C = \{P(x,a), \neg P(x, f(y))\}$.

$\{P(f(a),a), \neg P(f(a), f(f(a)))\}$	es una instancia básica de C.
$\{P(f(a),a), \neg P(f(f(a)), f(a))\}$	no es una instancia básica de C.
$\{P(x,a), \neg P(f(f(a)), f(a))\}$	no es una instancia básica de C.

Extensiones de Herbrand

- Def.: La extensión de Herbrand de un conjunto de cláusulas S es el conjunto de fórmulas
  $\text{EH}(S) = \{C\sigma : C \in S$ y, para toda variable x en $C, \sigma(x) \in \text{UH}(S)\}$.
- Prop.: $\text{EH}(L) = \bigcup_{i \geq 0} \text{EH}_i(L)$, donde $\text{EH}_i(L)$ es el nivel i de la $\text{EH}(L)$ definido por $\text{EH}_i(S) = \{C\sigma : C \in S$ y, para toda variable x en $C, \sigma(x) \in \text{UH}_i(S)\}$.
- Ejemplos:
  - Sea $S = \{\{P(x)\}, \{\neg P(f(x))\}\}$ (p. 8.17). Entonces,
    $$\begin{aligned} \text{EH}_0(S) &= \{\{P(a)\}, \{\neg P(f(a))\}\} \\ \text{EH}_1(S) &= \text{EH}_0(S) \cup \{\{P(f(a))\}, \{\neg P(f(f(a)))\}\} \\ \text{EH}_2(S) &= \text{EH}_1(S) \cup \{\{P(f(f(a)))\}, \{\neg P(f(f(f(a))))\}\} \end{aligned}$$
  - Sea $S = \{\{\neg P(x), Q(x)\}, \{P(a)\}, \{\neg Q(z)\}\}$ (p. 8.21).
    Entonces, $\text{EH}(S) = \{\{\neg P(a), Q(a)\}, \{P(a)\}, \{\neg Q(a)\}\}$.
  - Sea $S = \{\{\neg P(x), Q(x)\}, \{\neg Q(y), R(y)\}, \{P(a)\}, \{\neg R(a)\}\}$ (p. 8.21).
    Entonces, $\text{EH}(S) = \{\{\neg P(a), Q(a)\}, \{\neg Q(a), R(a)\}, \{P(a)\}, \{\neg R(a)\}\}$.

17

Teorema de Herbrand

- Teorema de Herbrand: Sea S un conjunto de cláusulas. Son equivalentes:
  1. S es consistente.
  2. $\text{EH}(S)$ es consistente (en el sentido proposicional).
- Prop.: Sea S un conjunto de cláusulas. Entonces, son equivalentes
  1. S es inconsistente.
  2. $\text{EH}(S)$ tiene un subconjunto finito inconsistente (en el sentido proposicional).
  3. Para algún i, $\text{EH}_i(S)$ es inconsistente (en el sentido proposicional).

18

Semidecisón mediante el teorema de Herbrand

- Entrada: Un conjunto de cláusulas S.
- Procedimiento:
  1. Hacer $i := 0$.
  2. Calcular $\text{EH}_i(S)$.
  3. Si $\text{EH}_i(S)$ es inconsistente (en el sentido proposicional), parar e indicar que S es inconsistente.
  4. Si $\text{EH}_i(S)$ es consistente (en el sentido proposicional), hacer $i := i+1$ y volver al paso 2.

19

Ejemplos de decisión mediante el teorema de Herbrand

- $S = \{\{\neg P(x), Q(x)\}, \{P(a)\}, \{\neg Q(z)\}\}$ (p. 17) es inconsistente.
  $\text{EH}_0(S) = \{\{\neg P(a), Q(a)\}, \{P(a)\}, \{\neg Q(a)\}\}$ es inconsistente.

1	$\{\neg P(a), Q(a)\}$	
2	$\{P(a)\}$	
3	$\{\neg Q(a)\}$	
4	$\{Q(a)\}$	Res $1,2$
5	$\square$	Res $3,4$

- $S = \{\{\neg P(x), Q(x)\}, \{\neg Q(y), R(y)\}, \{P(a)\}, \{\neg R(a)\}\}$ es inconsistente.
  $\text{EH}_0(S) = \{\{\neg P(a), Q(a)\}, \{\neg Q(a), R(a)\}, \{P(a)\}, \{\neg R(a)\}\}$.

1	$\{\neg P(a), Q(a)\}$	
2	$\{\neg Q(a), R(a)\}$	
3	$\{P(a)\}$	
4	$\{\neg R(a)\}$	
5	$\{Q(a)\}$	Res $1,3$
6	$\{R(a)\}$	Res $5,2$
7	$\square$	Res $6,4$

20

Ejemplos de decisión mediante el teorema de Herbrand

- $S = \{\{P(x)\}, \{\neg P(f(x))\}\}$ es inconsistente (p. 17).
  - $\text{EH}_0(S) = \{\{P(a)\}, \{\neg P(f(a))\}\}$ es consistente
    $\mathscr{I} = \{P(a)\} \models \text{EH}_0(S)$
  - $\text{EH}_1(S) = \text{EH}_0(S) \cup \{\{P(f(a))\}, \{\neg P(f(f(a)))\}\}$ es inconsistente.

    1 $\{P(f(a))\}$
    2 $\{\neg P(f(a))\}$
    3 $\square$ Res $1,2$

- $S = \{\{\neg P(x), Q(f(x),x)\}, \{P(g(b))\}, \{\neg Q(y,z)\}\}$ es inconsistente. Dem.:
  $S' = \{\{\neg P(g(b)), Q(f(g(b)),g(b))\}, \{P(g(b))\}, \{\neg Q(f(g(b)),g(b))\}\} \subset \text{EH}(S)$ es inconsistente.

    1 $\{\neg P(g(b)), Q(f(g(b)),g(b))\}$
    2 $\{P(g(b))\}$
    3 $\{\neg Q(f(g(b)),g(b))\}$
    4 $\{Q(f(g(b)),g(b))\}$ Res $1,2$
    5 $\square$ Res $3,3$

Bibliografía

1. M.L. Bonet *Apuntes de LPO.* (Univ. Politécnica de Cataluña, 2003) pp. 31–34.
2. C.L. Chang y R.C.T. Lee *Symbolic logic and mechanical theorem proving* (Academic Press, 1973) pp. 51–62.
3. M. Ojeda e I. Pérez *Lógica para la computación (Vol. 2: Lógica de Primer Orden)* (Ágora, 1997) pp. 59–74.
4. E. Paniagua, J.L. Sánchez y F. Martín *Lógica computacional* (Thomson, 2003) pp. 160–169.
5. L. Paulson *Logic and proof* (U. Cambridge, 2002) pp. 47–50.
6. U. Schöning *Logic for computer scientists* (Birkäuser, 1989) pp. 70–78.

Capítulo 10

Resolución en lógica de primer orden

Lógica informática (2005–06)
Tema 10: Resolución en lógica de primer orden

José A. Alonso Jiménez

Grupo de Lógica Computacional
Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

Ejemplos de consecuencia mediante resolución

- Ejemplo 1: $\{(\forall x)[P(x) \rightarrow Q(x)], (\exists x)P(x)\} \models (\exists x)Q(x)$
  syss $\{\{\neg P(x), Q(x)\}, \{P(a)\}, \{\neg Q(z)\}\}$ es inconsistente.

1	$\{\neg P(x), Q(x)\}$	Hipótesis
2	$\{P(a)\}$	Hipótesis
3	$\{\neg Q(z)\}$	Hipótesis
4	$\{Q(a)\}$	Resolvente de 1 y 2 con $\sigma = [x/a]$
5	$\square$	Resolvente de 3 y 4 con $\sigma = [z/a]$

- Ejemplo 2: $\{(\forall x)[P(x) \rightarrow Q(x)], (\forall x)[Q(x) \rightarrow R(x)]\} \models (\forall x)[P(x) \rightarrow R(x)]$
  syss $\{\{\neg P(x), Q(x)\}, \{\neg Q(y), R(y)\}, \{P(a)\}, \{\neg R(a)\}\}$ es inconsistente.

1	$\{\neg P(x), Q(x)\}$	Hipótesis
2	$\{\neg Q(y), R(y)\}$	Hipótesis
3	$\{P(a)\}$	Hipótesis
4	$\{\neg R(a)\}$	Hipótesis
5	$\{Q(a)\}$	Resolvente de 1 y 2 con $\sigma = [x/a]$
6	$\{R(a)\}$	Resolvente de 2 y 5 con $\sigma = [y/a]$
5	$\square$	Resolvente de 3 y 4 con $\sigma = \varepsilon$

Unificadores

- Def.: La sustitución σ es un unificador de los términos t_1 y t_2 si $t_1\sigma = t_2\sigma$.
- Def.: Los términos t_1 y t_2 son unificables si tienen algún unificador.
- Def.: t es una instancia común de t_1 y t_2 si existe una sustitución σ tal que $t = t_1\sigma = t_2\sigma$.
- Ejemplos:

t_1	t_2	Unificador	Instancia común
$f(x,g(z))$	$f(g(y),x)$	$[x/g(z),y/z]$	$f(g(z),g(z))$
$f(x,g(z))$	$f(g(y),x)$	$[x/g(y),z/y]$	$f(g(y),g(y))$
$f(x,g(z))$	$f(g(y),x)$	$[x/g(a),y/a]$	$f(g(a),g(a))$
$f(x,y)$	$f(y,x)$	$[x/a,y/a]$	$f(a,a)$
$f(x,y)$	$f(y,x)$	$[y/x]$	$f(x,x)$
$f(x,y)$	$g(a,b)$	No tiene	No tiene
$f(x,x)$	$f(a,b)$	No tiene	No tiene
$f(x)$	$f(g(x))$	No tiene	No tiene

- Nota: Las anteriores definiciones se extienden a conjuntos de términos y de literales. 3

Composición de sustituciones e identidad

- Composición de sustituciones:
  - Def.: La composición de las sustituciones σ_1 y σ_2 es la sustitución $\sigma_1\sigma_2$ definida por $x(\sigma_1\sigma_2) = (x\sigma_1)\sigma_2$, para toda variable x.
  - Ejemplo: Si $\sigma_1 = [x/f(z,a),y/w]$ y $\sigma_2 = [x/b,z/g(w)]$, entonces
    - $x\sigma_1\sigma_2 = (x\sigma_1)\sigma_2 = f(z,a)\sigma_2 = f(z\sigma_2,a\sigma_2) = f(g(w),a)$
    - $y\sigma_1\sigma_2 = (y\sigma_1)\sigma_2 = w\sigma_2 = w$
    - $z\sigma_1\sigma_2 = (z\sigma_1)\sigma_2 = z\sigma_2 = g(w)$
    - $w\sigma_1\sigma_2 = (w\sigma_1)\sigma_2 = w\sigma_2 = w$

    Por tanto, $\sigma_1\sigma_2 = [x/f(g(w),a),y/w,z/g(w)]$.
- Def.: La substitución identidad es la sustitución ε tal que, para todo x, $x\varepsilon = x$.
- Propiedades:
  1. Asociativa: $\sigma_1(\sigma_2\sigma_3) = (\sigma_1\sigma_2)\sigma_3$
  2. Neutro: $\sigma\varepsilon = \varepsilon\sigma = \sigma$.

4

Comparación de sustituciones

- Def.: La sustitución σ_1 es más general que la σ_2 si existe una sustitución σ_3 tal que $\sigma_2 = \sigma_1\sigma_3$, Se representa por $\sigma_2 \leq \sigma_1$.
- Def.: Las sustituciones σ_1 y σ_2 son equivalentes si $\sigma_1 \leq \sigma_2$ y $\sigma_2 \leq \sigma_1$. Se representa por $\sigma_1 \equiv \sigma_2$.
- Ejemplos: Sean $\sigma_1 = [x/g(z), y/z], \sigma_2 = [x/g(y), z/y]$ y $\sigma_3 = [x/g(a), y/a]$. Entonces,
  1. $\sigma_1 = \sigma_2[y/z]$
  2. $\sigma_2 = \sigma_1[z/y]$
  3. $\sigma_3 = \sigma_1[z/a]$
  4. $\sigma_1 \equiv \sigma_2$
  5. $\sigma_3 \leq \sigma_1$
- Ejemplo: $[x/a, y/a] \leq [y/x]$, ya que $[x/a, y/a] = [y/x][x/a, y/a]$.

5

Unificador de máxima generalidad

- Unificador de máxima generalidad:
- Def.: La sustitución σ es un unificador de máxima generalidad (UMG) de los términos t_1 y t_2 si
  - σ es un unificador de t_1 y t_2.
  - σ es más general que cualquier unificador de t_1 y t_2.
- Ejemplos:
  1. $[x/g(z), y/z]$ es un UMG de $f(x, g(z))$ y $f(g(y), x)$.
  2. $[x/g(y), z/y]$ es un UMG de $f(x, g(z))$ y $f(g(y), x)$.
  3. $[x/g(a), y/a]$ no es un UMG de $f(x, g(z))$ y $f(g(y), x)$.
- Nota: Las anterior definición se extienden a conjuntos de términos y de literales.

6

Unificación de listas de términos

- Notación de lista:
  - $(a_1,\dots,a_n)$ representa una lista cuyos elementos son $a_1,\dots,a_n$.
  - $(a|R)$ representa una lista cuyo primer elemento es a y resto es R.
  - $()$ representa la lista vacía.
- Unificadores de listas de términos:
  - Def.: σ es un unificador de $(s_1\dots,s_n)$ y $(t_1\dots,t_n)$ si $s_1\sigma = t_1\sigma,\dots,s_n\sigma = t_n\sigma$.
  - Def.: $(s_1\dots,s_n)$ y $(t_1\dots,t_n)$ son unificables si tienen algún unificador.
  - Def.: σ es un unificador de máxima generalidad (UMG) de $(s_1\dots,s_n)$ y $(t_1\dots,t_n)$ si σ es un unificador de $(s_1\dots,s_n)$ y $(t_1\dots,t_n)$ más general que cualquier otro.
- Aplicación de una sustitución a una lista de ecuaciones:
  - $(s_1 = t_1,\dots,s_n = t_n)\sigma = (s_1\sigma = t_1\sigma,\dots,s_n\sigma = t_n\sigma)$.
- Algoritmo de unificación de listas de términos:
  - Entrada: Lista de ecuaciones $L = (s_1 = t_1,\dots,s_n = t_n)$ y sustitución σ.
  - Salida: Un UMG de las listas $(s_1\dots,s_n)$ y $(t_1\dots,t_n)$, si son unificables;
    “No unificables”, en caso contrario.

Algoritmo de unificación

- Procedimiento $\mathrm{unif}(L,\sigma)$:

1. Si $L = ()$, entonces $\mathrm{unif}(L,\sigma) = \sigma$.
2. Si $L = (t = t|L')$, entonces $\mathrm{unif}(L,\sigma) = \mathrm{unif}(L',\sigma)$.
3. Si $L = (f(t_1,\dots,t_m) = f(t'_1\dots,t'_m)|L')$, entonces
   $\mathrm{unif}(L,\sigma) = \mathrm{unif}((t_1 = t'_1,\dots,t_m = t'_m|L'),\sigma)$.
4. Si $L = (x = t|L')$ (ó $L = (t = x|L')$) y x no aparece en t, entonces
   $\mathrm{unif}(L,\sigma) = \mathrm{unif}(L'[x/t],\sigma[x/t])$.
5. Si $L = (x = t|L')$ (ó $L = (t = x|L')$) y x aparece en t, entonces
   $\mathrm{unif}(L,\sigma) =$ “No unificables”.
6. Si $L = (f(t_1,\dots,t_m) = g(t'_1\dots,t'_m)|L')$, entonces
   $\mathrm{unif}(L,\sigma) =$ “No unificables”.
7. Si $L = (f(t_1,\dots,t_m) = f(t'_1\dots,t'_p)|L')$ y $m \neq p$, entonces
   $\mathrm{unif}(L,\sigma) =$ “No unificables”.

Algoritmo de unificación de dos términos

- Entrada: Dos términos t_1 y t_2.
- Salida: Un UMG de t_1 y t_2, si son unificables;
  "No unificables", en caso contrario.
- Procedimiento: $\mathsf{unif}((t_1 = t_2), \varepsilon)$.
- Ejemplo 1: Unificar $f(x, g(z))$ y $f(g(y), x)$:

$$
\begin{array}{lll}
 & \mathsf{unif}((f(x,g(z)) = f(g(y),x)),\varepsilon) & \\
= & \mathsf{unif}((x = g(y), g(z) = x), \varepsilon) & \text{por 3} \\
= & \mathsf{unif}((g(z) = x)[x/g(y)], \varepsilon[x/g(y)]) & \text{por 4} \\
= & \mathsf{unif}((g(z) = g(y)), [x/g(y)]) & \\
= & \mathsf{unif}((z = y), [x/g(y)]) & \text{por 3} \\
= & \mathsf{unif}((), [x/g(y)][z/y]) & \text{por 4} \\
= & \mathsf{unif}((), [x/g(y), z/y]) & \\
= & [x/g(y), z/y] & \text{por 1}
\end{array}
$$

9

Ejemplos de unificación

- Ejemplo 2: Unificar $f(x, b)$ y $f(a, y)$:

$$
\begin{array}{lll}
 & \mathsf{unif}((f(x,b) = f(a,y),\varepsilon) & \\
= & \mathsf{unif}((x = a, b = y), \varepsilon) & \text{por 3} \\
= & \mathsf{unif}((b = y)[x/a], \varepsilon[x/a]) & \text{por 4} \\
= & \mathsf{unif}((b = y), [x/a]) & \\
= & \mathsf{unif}((), [x/a][y/b]) & \text{por 4} \\
= & [x/a, y/b]) & \text{por 1}
\end{array}
$$

- Ejemplo 3: Unificar $f(x, x)$ y $f(a, b)$:

$$
\begin{array}{lll}
 & \mathsf{unif}((f(x,x) = f(a,b)),\varepsilon) & \\
= & \mathsf{unif}((x = a, x = b), \varepsilon) & \text{por 3} \\
= & \mathsf{unif}((x = b)[x/a], \varepsilon[x/a]) & \text{por 4} \\
= & \mathsf{unif}((a = b), [x/a]) & \\
= & \text{“No unificable”} & \text{por 6}
\end{array}
$$

10

Ejemplos de unificación

- Ejemplo 4: Unificar $f(x,g(y))$ y $f(y,x)$:

$$\begin{array}{lll} & \mathsf{unif}((f(x,g(y)) = f(y,x)),\varepsilon) & \\ = & \mathsf{unif}((x = y, g(y) = x),\varepsilon) & \text{por 3} \\ = & \mathsf{unif}((g(y) = x)[x/y],\varepsilon[x/y]) & \text{por 4} \\ = & \mathsf{unif}((g(y) = y),[x/y]) & \\ = & \text{“No unificable”} & \text{por 5} \end{array}$$

- Ejemplo 5: Unificar $j(w,a,h(w))$ y $j(f(x,y),x,z)$

$$\begin{array}{lll} & \mathsf{unif}((j(w,a,h(w)) = j(f(x,y),x,z))\varepsilon) & \\ = & \mathsf{unif}((w = f(x,y), a = x, h(w) = z),\varepsilon) & \text{por 3} \\ = & \mathsf{unif}((a = x, h(w) = z)[w/f(x,y)],\varepsilon[w/f(x,y)]) & \text{por 4} \\ = & \mathsf{unif}((a = x, h(f(x,y)) = z),[w/f(x,y)]) & \\ = & \mathsf{unif}((h(f(x,y)) = z)[x/a],[w/f(x,y)][x/a]) & \text{por 4} \\ = & \mathsf{unif}((h(f(a,y)) = z),[w/f(a,y),x/a]) & \\ = & \mathsf{unif}((),[w/f(a,y),x/a][z/h(f(a,y))]) & \text{por 4} \\ = & [w/f(a,y),x/a,z/h(f(a,y)))] & \text{por 1} \end{array}$$

11

Ejemplos de unificación

- Ejemplo 6: Unificar $j(w,a,h(w))$ y $j(f(x,y),x,y)$

$$\begin{array}{lll} & \mathsf{unif}((j(w,a,h(w)) = j(f(x,y),x,y))\varepsilon) & \\ = & \mathsf{unif}((w = f(x,y), a = x, h(w) = y),\varepsilon) & \text{por 3} \\ = & \mathsf{unif}((a = x, h(w) = y)[w/f(x,y)],\varepsilon[w/f(x,y)]) & \text{por 4} \\ = & \mathsf{unif}((a = x, h(f(x,y)) = y),[w/f(x,y)]) & \\ = & \mathsf{unif}((h(f(x,y)) = y)[x/a],[w/f(x,y)][x/a]) & \text{por 4} \\ = & \mathsf{unif}((h(f(a,y)) = y),[w/f(a,y),x/a]) & \\ = & \text{“No unificable”} & \text{por 5} \end{array}$$

- Ejemplo 7: Unificar $f(a,y)$ y $f(a,b)$:

$$\begin{array}{lll} & \mathsf{unif}((f(a,y) = f(a,b),\varepsilon)) & \\ = & \mathsf{unif}((a = a, y = b),\varepsilon) & \text{por 3} \\ = & \mathsf{unif}((y = b),\varepsilon) & \text{por 2} \\ = & \mathsf{unif}((),[y/b]) & \text{por 4} \\ = & [y/b] & \text{por 1} \end{array}$$

12

Separación de variables

- Def.: La sustitución $[x_1/t_1,\dots,x_n/t_n]$ es un renombramiento si todos los t_i son variables.
- Prop.: Si θ es un renombramiento, entonces $C \equiv C\theta$.
- Def.: Las cláusulas C_1 y C_2 están separadas sin no tienen ninguna variable común.
- Def.: Una separación de las variables de C_1 y C_2 es un par de renombramientos (θ_1,θ_2) tales que $C_1\theta_1$ y $C_2\theta_2$ están separadas.
- Ejemplo: Una separación de variables de $C_1 = \{P(x), Q(x,y)\}$ y $C_2 = \{R(f(x,y))\}$ es $(\theta_1 = [x/x_1, y/y_1], \theta_2 = [x/x_2, y/y_2])$.

Resolvente binaria

- Def.: La cláusula C es una resolvente binaria de las cláusulas C_1 y C_2 si existen una separación de variables (θ_1,θ_2) de C_1 y C_2, un literal $L_1 \in C_1$, un literal $L_2 \in C_2$ y un UMG σ de $L_1\theta_1$ y $L_2^c\theta_2$ tales que
  $$C = (C_1\theta_1\sigma \setminus \{L_1\theta_1\sigma_1\}) \cup (C_2\theta_2\sigma \setminus \{L_2\theta_2\sigma\}).$$
- Ejemplo: Sean
  $$\begin{aligned}
  C_1 &= \{\neg P(x), Q(f(x))\},\\
  C_2 &= \{\neg Q(x), R(g(x))\},\\
  L_1 &= Q(f(x)),\\
  L_2 &= \neg Q(x),\\
  \theta_1 &= [x/x_1],\\
  \theta_2 &= [x/x_2],\\
  L_1\theta_1 &= Q(f(x_1)),\\
  L_2^c\theta_2 &= Q(x_2),\\
  \sigma &= [x_2/f(x_1)]
  \end{aligned}$$
  Entonces, $C = \{\neg P(x_1), R(g(f(x_1)))\}$ es una resolvente binaria de C_1 y C_2.

Factorización

- Def.: La cláusula C es un factor de la cláusula D si existen dos literales L_1 y L_2 en D que son unificables y $C = D\sigma \setminus \{L_2\sigma\}$ donde σ es un UMG de L_1 y L_2.
- Ejemplo: Sean

$$\begin{aligned} D &= \{P(x,y), P(y,x), Q(a)\} \\ L_1 &= P(x,y) \\ L_2 &= P(y,x) \\ \sigma &= [y/x] \end{aligned}$$

Entonces,
$C = \{P(x,x), Q(a)\}$ es un factor de D.

Ejemplos de refutación por resolución

- Refutación de $S = \{\{\neg P(x, f(x,y))\}, \{P(a,z), \neg Q(z,v)\}, \{Q(u,a)\}\}$

1	$\{\neg P(x, f(x,y))\}$	Hipótesis
2	$\{P(a,z), \neg Q(z,v)\}$	Hipótesis
3	$\{Q(u,a)\}$	Hipótesis
4	$\{\neg Q(f(a,y),v)\}$	Resolvente de 1 y 2 con $\sigma = [x/a, z/f(a,y)]$
5	$\square$	Resolvente de 3 y 4 con $\sigma = [u/f(a,y), v/a]$

- Refutación de $S = \{\{P(x)\}, \{\neg P(f(x))\}\}$

1	$\{P(x)\}$	Hipótesis
2	$\{\neg P(f(x))\}$	Hipótesis
3	$\square$	Resolvente de 1 y 2 con $\theta_1 = \varepsilon, \theta_2 = [x/x'], \sigma = [x/f(x')]$

- Refutación de $S = \{\{P(x,y), P(y,x)\}, \{\neg P(u,v), \neg P(v,u)\}\}$

1	$\{P(x,y), P(y,x)\}$	Hipótesis
2	$\{\neg P(u,v), \neg P(v,u)\}$	Hipótesis
3	$\{P(x,x)\}$	Factor de 1 con $[y/x]$
4	$\{\neg P(u,u)\}$	Factor de 2 con $[v/u]$
5	$\square$	Resolvente de 3 y 4 con $[x/u]$

Resolución

- Sea S un conjunto de cláusulas.
- La sucesión $(C_1,\dots,C_n)$ es una demostración por resolución de la cláusula C a partir de S si $C = C_n$ y para todo $i \in \{1,...,n\}$ se verifica una de las siguientes condiciones:
  - $C_i \in S$;
  - existen $j,k < i$ tales que C_i es una resolvente de C_j y C_k
  - existe $j < i$ tal que C_i es un factor de C_j
- La cláusula C es demostrable por resolución a partir de S si existe una demostración por resolución de C a partir de S.
- Una refutación por resolución de S es una demostración por resolución de la cláusula vacía a partir de S.
- Se dice que S es refutable por resolución si existe una refutación por resolución a partir de S.

Demostraciones por resolución

- Def.: Sean $S_1,\dots,S_n$ formas clausales de las fórmulas $F_1,\dots,F_n$ y S una forma clausal de $\neg F$. Una demostración por resolución de F a partir de $\{F_1,\dots,F_n\}$ es una refutación por resolución de $S_1 \cup \dots \cup S_n \cup S$.
- Def.: La fórmula F es demostrable por resolución a partir de $\{F_1,\dots,F_n\}$ si existe una demostración por resolución de F a partir de $\{F_1,\dots,F_n\}$.
  Se representa por $\{F_1,\dots,F_n\} \vdash_{Res} F$.
- Ejemplo: (tema 8 p. 21) $\{(\forall x)[P(x) \rightarrow Q(x)], (\exists x)P(x)\} \vdash_{Res} (\exists x)Q(x)$

1	$\{\neg P(x), Q(x)\}$	Hipótesis
2	$\{P(a)\}$	Hipótesis
3	$\{\neg Q(z)\}$	Hipótesis
4	$\{Q(a)\}$	Resolvente de 1 y 2 con $[x/a]$
5	$\square$	Resolvente de 3 y 4 con $[z/a]$

Ejemplos de demostraciones por resolución

- Ejemplo: (tema 8 p. 21)
  $\{(\forall x)[P(x) \to Q(x)], (\forall x)[Q(x) \to R(x)] \vdash_{Res} (\forall x)[P(x) \to R(x)]\}$

1	$\{\neg P(x), Q(x)\}$	Hipótesis
2	$\{\neg Q(y), R(y)\}$	Hipótesis
3	$\{P(a)\}$	Hipótesis
4	$\{\neg R(a)\}$	Hipótesis
5	$\{Q(a)\}$	Resolvente de 1 y 2 con $[x/a]$
6	$\{R(a)\}$	Resolvente de 2 y 5 con $[y/a]$
5	$\square$	Resolvente de 6 y 4 con

- Ejemplo: (tema 6 p. 55) $\vdash_{Res} (\exists x)[P(x) \to (\forall y)P(y)]$

1	$\{P(x)\}$	Hipótesis
2	$\{\neg P(f(x))\}$	Hipótesis
3	$\square$	Resolvente de 1 y 2 con $\theta_2 = [x/x'], \sigma = [x/f(x')]$

Ejemplos de demostraciones por resolución

- Ejemplo: $\vdash_{Res} (\forall x)(\exists y)\neg(P(y,x) \leftrightarrow \neg P(y,y))$
  - Forma clausal:

$$
\begin{array}{ll}
 & \neg(\forall x)(\exists y)\neg(P(y,x) \leftrightarrow \neg P(y,y)) \\
\equiv & \neg(\forall x)(\exists y)\neg((P(y,x) \to \neg P(y,y)) \wedge (\neg P(y,y) \to P(y,x))) \\
\equiv & \neg(\forall x)(\exists y)\neg((\neg P(y,x) \vee \neg P(y,y)) \wedge (\neg\neg P(y,y) \vee P(y,x))) \\
\equiv & \neg(\forall x)(\exists y)\neg((\neg P(y,x) \vee \neg P(y,y)) \wedge (P(y,y) \vee P(y,x))) \\
\equiv & (\exists x)(\forall y)\neg\neg((\neg P(y,x) \vee \neg P(y,y)) \wedge (P(y,y) \vee P(y,x))) \\
\equiv & (\exists x)(\forall y)((\neg P(y,x) \vee \neg P(y,y)) \wedge (P(y,y) \vee P(y,x))) \\
\equiv_{sat} & (\forall y)((\neg P(y,a) \vee \neg P(y,y)) \wedge (P(y,y) \vee P(y,a))) \\
\equiv & \{\{\neg P(y,a), \neg P(y,y)\}, \{P(y,y), P(y,a)\}\}
\end{array}
$$

  - Refutación:

1	$\{\neg P(y,a), \neg P(y,y)\}$	Hipótesis
2	$\{P(y,y), P(y,a)\}$	Hipótesis
3	$\{\neg P(a,a)\}$	Factor de 1 con $[y/a]$
4	$\{\neg P(a,a)\}$	Factor de 2 con $[y/a]$
5	$\square$	Resolvente de 3 y 4

Paradoja del barbero de Russell

Ejemplo (Paradoja del barbero de Russell): En una isla pequeña hay sólo un barbero. El gobernador de la isla ha publicado la siguiente norma: *"El barbero afeita a todas las personas que no se afeitan a sí misma y sólo a dichas personas"*. Demostrar que la norma es inconsistente.

– Representación: $(\forall x)[\text{afeita}(b,x) \leftrightarrow \neg\text{afeita}(x,x)]$

– Forma clausal:

$$
\begin{array}{ll}
 & (\forall x)[\text{afeita}(b,x) \leftrightarrow \neg\text{afeita}(x,x)] \\
\equiv & (\forall x)[(\text{afeita}(b,x) \rightarrow \neg\text{afeita}(x,x)) \wedge (\neg\text{afeita}(x,x) \rightarrow \text{afeita}(b,x))] \\
\equiv & (\forall x)[(\neg\text{afeita}(b,x) \vee \neg\text{afeita}(x,x)) \wedge (\neg\neg\text{afeita}(x,x) \vee \text{afeita}(b,x))] \\
\equiv & (\forall x)[(\neg\text{afeita}(b,x) \vee \neg\text{afeita}(x,x)) \wedge (\text{afeita}(x,x) \vee \text{afeita}(b,x))] \\
\equiv & \{\{\neg\text{afeita}(b,x), \neg\text{afeita}(x,x)\}, \{\text{afeita}(x,x), \text{afeita}(b,x)\}\}
\end{array}
$$

– Refutación:

1	$\{\neg\text{afeita}(b,x), \neg\text{afeita}(x,x)\}$	Hipótesis
2	$\{\text{afeita}(x,x), \text{afeita}(b,x)\}$	Hipótesis
3	$\{\neg\text{afeita}(b,b)\}$	Factor de 1 con $[x/b]$
4	$\{\text{afeita}(b,b)\}$	Factor de 2 con $[x/b]$
5	$\square$	Resolvente de 3 y 4

21

Adecuación y completitud de la resolución

- Propiedades:
  - Si C es una resolvente de C_1 y C_2, entonces $\{C_1, C_2\} \models C$.
  - Si D es un factor de C entonces $C \models D$.
  - Si $\square \in S$, entonces S es inconsistente.
  - Si el conjunto de cláusulas S es refutable por resolución, entonces S es inconsistente.
- Teor.: El cálculo de resolución (para la lógica de primer orden sin igualdad) es adecuado y completo; es decir,

  Adecuado: $S \vdash_{Res} F \implies S \models F$

  Completo: $S \models F \implies S \vdash_{Res} F$

22

Determinación de no–consecuencia por resolución

- Enunciado: Comprobar, por resolución, que
  $(\forall x)[P(x) \lor Q(x)] \not\models (\forall x)P(x) \lor (\forall x)Q(x)$.
- Reducción 1: Comprobar que es consistente
  $\{(\forall x)[P(x) \lor Q(x)], \neg((\forall x)P(x) \lor (\forall x)Q(x))\}$
- Reducción 2: Comprobar que es consistente
  $\{\{P(x), Q(x)\}, \{\neg P(a)\}, \{\neg Q(b)\}\}$
- Resolución:

1	$\{P(x), Q(x)\}$	Hipótesis
2	$\{\neg P(a)\}$	Hipótesis
3	$\{\neg Q(b)\}$	Hipótesis
4	$\{Q(a)\}$	Resolvente de 1 y 2
5	$\{P(b)\}$	Resolvente de 1 y 3

- Modelo: $U = \{a, b\}, I(P) = \{b\}, I(Q) = \{a\}$.

Bibliografía

1. Fitting, M. *First-Order Logic and Automated Theorem Proving (2nd ed.)* (Springer, 1996) pp. 137–141.
2. M.L. Bonet *Apuntes de LPO.* (Univ. Politécnica de Cataluña, 2003) pp. 34–40.
3. C.L. Chang y R.C.T. Lee *Symbolic logic and mechanical theorem proving* (Academic Press, 1973) pp. 70–99.
4. M. Genesereth *Computational Logic (Chapter 9: Relational Resolution)* (Stanford University, 2003)
5. S. Hölldobler *Computational logic.* (U. de Dresden, 2004) pp. 71–74.
6. M. Ojeda e I. Pérez *Lógica para la computación (Vol. 2: Lógica de Primer Orden)* (Ágora, 1997) pp. 138–164.
7. L. Paulson *Logic and proof* (U. Cambridge, 2002) pp. 50–61.
8. U. Schöning *Logic for computer scientists* (Birkäuser, 1989) pp. 79–96.