

# Lógica informática (2012–13)

## Tema 1: Sintaxis y semántica de la lógica proposicional

José A. Alonso Jiménez  
Andrés Cordon Franco  
María J. Hidalgo Doblado

Grupo de Lógica Computacional  
Departamento de Ciencias de la Computación e I.A.  
Universidad de Sevilla

# Tema 1: Sintaxis y semántica de la lógica proposicional

1. Introducción
2. Sintaxis de la lógica proposicional
3. Semántica proposicional

# Tema 1: Sintaxis y semántica de la lógica proposicional

## 1. Introducción

Panorama de la lógica

Ejemplos de argumentos y formalizaciones

## 2. Sintaxis de la lógica proposicional

## 3. Semántica proposicional

# Lógica

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

## 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,  $q$ .
- ▶  $p \wedge \neg q \rightarrow r, \neg r, p \models q$ .

# Tema 1: Sintaxis y semántica de la lógica proposicional

## 1. Introducción

## 2. Sintaxis de la lógica proposicional

El lenguaje de la lógica proposicional

Recursión e inducción sobre fórmulas

Árboles de análisis (o de formación)

Eliminación de paréntesis

Subfórmulas

## 3. Semántica proposicional

## El lenguaje de la lógica proposicional

- ▶ 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 (fórmulas atómicas).
    - ▶ 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 las variables proposicionales.
  - ▶  $Prop$  representa el conjunto de las fórmulas.
  - ▶  $*$  representa una conectiva binaria.
- ▶ Forma de Backus Naur (BNF) de las fórmulas proposicionales:
  - ▶  $F ::= p \mid \neg G \mid (F \wedge G) \mid (F \vee G) \mid (F \rightarrow G) \mid (F \leftrightarrow G).$

## Definiciones por recursión sobre fórmulas

- ▶ Número de paréntesis de una fórmula:
  - ▶ Def: El número de paréntesis de una fórmula  $F$  se define recursivamente por:

$$\text{np}(F) = \begin{cases} 0, & \text{si } F \text{ es atómica;} \\ \text{np}(G), & \text{si } F \text{ es } \neg G; \\ 2 + \text{np}(G) + \text{np}(H), & \text{si } F \text{ es } (G * H) \end{cases}$$

- ▶ Ejemplos:
  - ▶  $\text{np}(p) = 0$
  - ▶  $\text{np}(q) = 0$
  - ▶  $\text{np}(\neg q) = 0$
  - ▶  $\text{np}((\neg q \vee p)) = 2$
  - ▶  $\text{np}((p \rightarrow (\neg q \vee p))) = 4$

## Demostración por inducción sobre fórmulas

- ▶ **Principio de inducción sobre fórmulas:** Sea  $\mathcal{P}$  una propiedad sobre las fórmulas que verifica las siguientes condiciones:
  - ▶ Todas las fórmulas atómicas tienen la propiedad  $\mathcal{P}$ .
  - ▶ Si  $F$  y  $G$  tienen la propiedad  $\mathcal{P}$ , entonces  $\neg F$ ,  $(F \wedge G)$ ,  $(F \vee G)$ ,  $(F \rightarrow G)$  y  $(F \leftrightarrow G)$ , tienen la propiedad  $\mathcal{P}$ .

Entonces todas las fórmulas proposicionales tienen la propiedad  $\mathcal{P}$ .

- ▶ **Propiedad:** Todas las fórmulas proposicionales tienen un número par de paréntesis.
  - ▶ **Demostración por inducción sobre las fórmulas.**
    - ▶ **Base:**  $F$  atómica  $\implies np(F) = 0$  es par.
    - ▶ **Paso:** Supongamos que  $np(F)$  y  $np(G)$  es par (**hipótesis de inducción**). Entonces,
      - $np(\neg F) = np(F)$  es par y
      - $np((F * G)) = 2 + np(F) + np(G)$  es par,
 para cualquier conectiva binaria  $*$ .



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

## Subfórmulas

- ▶ Def: El conjunto  $\text{Subf}(F)$  de las **subfórmulas** de una fórmula  $F$  se define recursivamente por:

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

- ▶ Ejemplos:

- ▶  $\text{Subf}(p) = \{p\}$
- ▶  $\text{Subf}(q) = \{q\}$
- ▶  $\text{Subf}(\neg q) = \{\neg q, q\}$
- ▶  $\text{Subf}(\neg q \vee p) = \{\neg q \vee p, \neg q, q, p\}$
- ▶  $\text{Subf}(p \rightarrow \neg q \vee p) = \{p \rightarrow \neg q \vee p, p, \neg q \vee p, \neg q, q\}$

# Tema 1: Sintaxis y semántica de la lógica proposicional

## 1. Introducción

## 2. Sintaxis de la lógica proposicional

## 3. Semántica proposicional

Valores y funciones de verdad

Interpretaciones

Modelos, satisfacibilidad y validez

Algoritmos para satisfacibilidad y validez

Selección de tautologías

Equivalencia lógica

Modelos de conjuntos de fórmulas

Consistencia y consecuencia lógica

Argumentaciones y problemas lógicos

## Valores y funciones de verdad

- ▶ Valores de verdad ( $\mathbb{B}$ ): **1**: verdadero y **0**: falso.
- ▶ Funciones de verdad:

- ▶  $H_{\neg} : \{0, 1\} \rightarrow \{0, 1\}$  t.q.  $H_{\neg}(i) = \begin{cases} 1, & \text{si } i = 0; \\ 0, & \text{si } i = 1. \end{cases}$

- ▶  $H_{\wedge} : \{0, 1\}^2 \rightarrow \{0, 1\}$  t.q.  $H_{\wedge}(i, j) = \begin{cases} 1, & \text{si } i = j = 1; \\ 0, & \text{en otro caso.} \end{cases}$

- ▶  $H_{\vee} : \{0, 1\}^2 \rightarrow \{0, 1\}$  t.q.  $H_{\vee}(i, j) = \begin{cases} 0, & \text{si } i = j = 0; \\ 1, & \text{en otro caso.} \end{cases}$

- ▶  $H_{\rightarrow} : \{0, 1\}^2 \rightarrow \{0, 1\}$  t.q.  $H_{\rightarrow}(i, j) = \begin{cases} 0, & \text{si } i = 1, j = 0; \\ 1, & \text{en otro caso.} \end{cases}$

- ▶  $H_{\leftrightarrow} : \{0, 1\}^2 \rightarrow \{0, 1\}$  t.q.  $H_{\leftrightarrow}(i, j) = \begin{cases} 1, & \text{si } i = j; \\ 0, & \text{en otro caso.} \end{cases}$

## Interpretaciones de fórmulas

- Funciones de verdad mediante **tablas de verdad**:

$i$	$\neg i$
1	0
0	1

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

- Interpretación:

- Def.: Una **interpretación** es una aplicación  $I : VP \rightarrow \mathbb{B}$ .
- Prop: Para cada interpretación  $I$  existe una única aplicación  $I' : Prop \rightarrow \mathbb{B}$  tal que:

$$I'(F) = \begin{cases} I(F), & \text{si } F \text{ es atómica;} \\ \neg(I'(G)), & \text{si } F = \neg G; \\ H_*(I'(G), I'(H)), & \text{si } F = G * H \end{cases}$$

Se dice que  $I'(F)$  es el **valor de verdad de  $F$  respecto de  $I$** .

## Interpretaciones de fórmulas

- ▶ Ejemplo: Sea  $F = (p \vee q) \wedge (\neg q \vee r)$ 
  - ▶ valor de  $F$  en una interpretación  $I_1$  tal que

$$I_1(p) = I_1(r) = 1, I_1(q) = 0$$

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

- ▶ valor de  $F$  en una interpretación  $I_2$  tal que
  - $I_2(r) = 1, I_2(p) = I_2(q) = 0$

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

- ▶ Prop.: Sea  $F$  una fórmula y  $I_1, I_2$  dos interpretaciones. Si  $I_1(p) = I_2(p)$  para todas las variables proposicionales de  $F$ , entonces  $I_1'(F) = I_2'(F)$ .
- ▶ Notación: Se escribe  $I(F)$  en lugar de  $I'(F)$ .

## Modelos y satisfacibilidad

### ► Modelo de una fórmula

► Def.:  $I$  es modelo de  $F$  si  $I(F) = 1$ .

► Notación:  $I \models F$ .

► Ejemplo (continuación del anterior):

– si  $I_1(p) = I_1(r) = 1, I_1(q) = 0$ , entonces  $I_1 \models (p \vee q) \wedge (\neg q \vee r)$

– si  $I_2(r) = 1, I_2(p) = I_2(q) = 0$ , entonces  $I_2 \not\models (p \vee q) \wedge (\neg q \vee r)$ .

### ► Fórmulas satisfacibles e insatisfacibles

► Def.:  $F$  es satisfacible si  $F$  tiene algún modelo.

► Ejemplo:  $(p \rightarrow q) \wedge (q \rightarrow r)$  es satisfacible

$$I(p) = I(q) = I(r) = 0.$$

► Def.:  $F$  es insatisfacible si  $F$  no tiene ningún modelo.

► Ejemplo:  $p \wedge \neg p$  es insatisfacible

$p$	$\neg p$	$p \wedge \neg p$
1	0	0
0	1	0

## Tautologías y contradicciones

- ▶ Def.:  $F$  es una tautología (o válida) si toda interpretación es modelo de  $F$ . Se representa por  $\models F$ .
- ▶ Def.:  $F$  es una contradicción si ninguna interpretación es modelo de  $F$ .
- ▶ Def.:  $F$  es contingente si no es tautología ni contradicción.
- ▶ Ejemplos:
  1.  $(p \rightarrow q) \vee (q \rightarrow p)$  es una tautología.
  2.  $(p \rightarrow q) \wedge \neg(p \rightarrow q)$  es una contradicción.
  3.  $p \rightarrow q$  es contingente.

$p$	$q$	$p \rightarrow q$	$q \rightarrow p$	$(p \rightarrow q) \vee (q \rightarrow p)$	$\neg(p \rightarrow q)$	$(p \rightarrow q) \wedge \neg(p \rightarrow q)$
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

Todas las fórmulas		
Tautologías	Contingentes	Contradicciones
Verdadera en todas las interpretaciones  (ej. $p \vee \neg p$ )	Verdadera en algunas interpretaciones y falsa en otras  (ej. $p \rightarrow q$ )	Falsa en todas las interpretaciones  (ej. $p \wedge \neg p$ )
Satisfacibles		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  $\not\implies \neg F$  es insatisfacible.

$p \rightarrow q$  es satisfacible.

$$I(p) = I(q) = 1$$

$\neg(p \rightarrow q)$  es satisfacible.

$$I(p) = 1, I(q) = 0.$$

## Algoritmos para SAT y TAUT

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

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

- Tabla de verdad simplificada para  $\models (p \rightarrow q) \vee (q \rightarrow p)$ :

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

- ▶ 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  $\not\models (p \leftrightarrow q) \vee (q \leftrightarrow p)$

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

## Selección de tautologías

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

## Fórmulas equivalentes

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

## Propiedades de la equivalencia lógica

- ▶ Relación entre equivalencia y bicondicional:
  - ▶  $F \equiv G \text{ syss } \models F \leftrightarrow G$ .
- ▶ 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  $F$  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:
 
$$\begin{aligned} F &= \neg(p \wedge q) \rightarrow \neg(p \wedge \neg\neg r) \\ G &= \neg(p \wedge q) \\ G' &= \neg p \vee \neg q \\ F' &= (\neg p \vee \neg q) \rightarrow \neg(p \wedge \neg\neg r) \end{aligned}$$

## Modelo de conjuntos de fórmulas

► Notación:

- $S, S_1, S_2, \dots$  representarán conjuntos de fórmulas.

► Modelo de un conjunto de fórmulas:

- Def.:  $I$  es modelo de  $S$  si para toda  $F \in S$  se tiene que  $I \models F$ .

- Representación:  $I \models S$ .

- Ejemplo: Sea  $S = \{(p \vee q) \wedge (\neg q \vee r), q \rightarrow r\}$

La interpretación  $I_1$  tal que  $I_1(p) = 1, I_1(q) = 0, I_1(r) = 1$  es modelo de  $S$  ( $I_1 \models S$ ).

$$\begin{array}{ccccccccccc} \{(p & \vee & q) & \wedge & (\neg & q & \vee & r), & q & \rightarrow & r\} \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}$$

La interpretación  $I_2$  tal que  $I_2(p) = 0, I_2(q) = 1, I_2(r) = 0$  no es modelo de  $S$  ( $I_2 \not\models S$ ).

$$\begin{array}{ccccccccccc} \{(p & \vee & q) & \wedge & (\neg & q & \vee & r), & q & \rightarrow & r\} \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{array}$$

## Conjunto consistente de fórmulas

- ▶ Def.:  $S$  es consistente si  $S$  tiene algún modelo.
- ▶ Def.:  $S$  es inconsistente si  $S$  no tiene ningún modelo.
- ▶ Ejemplos:
  - ▶  $\{(p \vee q) \wedge (\neg q \vee r), p \rightarrow r\}$  es consistente (con modelos  $l_4, l_6, l_8$ )
  - ▶  $\{(p \vee q) \wedge (\neg q \vee r), p \rightarrow r, \neg r\}$  es inconsistente

	$p$	$q$	$r$	$(p \vee q)$	$(\neg q \vee r)$	$(p \vee q) \wedge (\neg q \vee r)$	$p \rightarrow r$	$\neg r$
$l_1$	0	0	0	0	1	0	1	1
$l_2$	0	0	1	0	1	0	1	0
$l_3$	0	1	0	1	0	0	1	1
$l_4$	0	1	1	1	1	1	1	0
$l_5$	1	0	0	1	1	1	0	1
$l_6$	1	0	1	1	1	1	1	0
$l_7$	1	1	0	1	0	0	0	1
$l_8$	1	1	1	1	1	1	1	0

## Consecuencia lógica

- ▶ Def.:  $F$  es consecuencia de  $S$  si todos los modelos de  $S$  son modelos de  $F$ .
- ▶ Representación:  $S \models F$ .
- ▶ Ejemplos:  $\{p \rightarrow q, q \rightarrow r\} \models p \rightarrow r$  y  $\{p\} \not\models p \wedge q$

	$p$	$q$	$r$	$p \rightarrow q$	$q \rightarrow r$	$p \rightarrow r$	$p$	$q$	$p \wedge q$
$I_1$	0	0	0	1	1	1	1	1	1
$I_2$	0	0	1	1	1	1	1	0	0
$I_3$	0	1	0	1	0	1	0	1	0
$I_4$	0	1	1	1	1	1	0	0	0
$I_5$	1	0	0	0	1	0			
$I_6$	1	0	1	0	1	1			
$I_7$	1	1	0	1	0	0			
$I_8$	1	1	1	1	1	1			

## 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

## Ejemplo de argumentación

- ▶ Problema de los animales: Se sabe que
  1. Los animales con pelo o 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:

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

$$\models \text{es\_cebra}$$

## 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:

$$F_1 = a \leftrightarrow (b \wedge c \leftrightarrow c), F_2 = b \leftrightarrow (a \wedge c \rightarrow b \wedge c \wedge \neg a) \text{ y}$$

$$F_3 = c \leftrightarrow (\neg b \leftrightarrow a \vee b).$$

- ▶ Modelos de  $\{F_1, F_2, F_3\}$ :

Si  $I$  es modelo de  $\{F_1, F_2, F_3\}$ , entonces  $I(a) = 1, I(b) = 1, I(c) = 0$ .

- ▶ 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. J.A. Díez *Iniciación a la Lógica*, (Ariel, 2002)  
Cap. 2 (El lenguaje de la lógica proposicional) y 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).