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.
  - $p \land \neg q \rightarrow r,\ \neg r,\ p \models q$.

Sintaxis proposicional: Fórmulas proposicionales

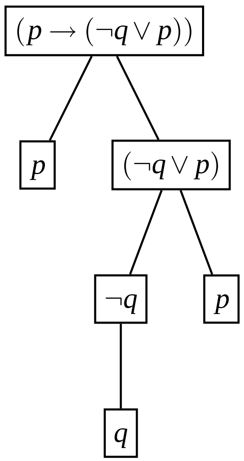
- Alfabeto proposicional:
  - variables proposicionales: $p_0, p_1, \dots; p, q, r$.
  - conectivas lógicas:
    - monaria: $\neg$ (negación),
    - binarias: $\land$ (conjunción), $\lor$ (disyunción), $\rightarrow$ (condicional), $\leftrightarrow$ (bicondicional).
  - símbolos auxiliares: "(" y ")".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

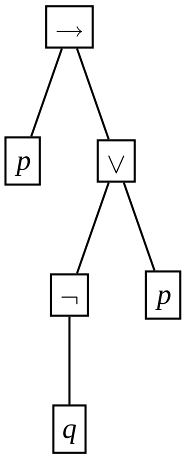
Fórmulas proposicionales (BNF)

- 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:
  - $F ::= p \mid \neg G \mid (F \wedge G) \mid (F \vee G) \mid (F \to G) \mid (F \leftrightarrow G)$.

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: $\neg, \wedge, \vee, \to, \leftrightarrow$.

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

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

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$(p \to \neg q \vee p) = \{p \to \neg q \vee p, p, \neg q \vee p, \neg q, q\}$

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. $H_\leftrightarrow(i,j) = \begin{cases} 1, & \text{si } i = j; \\ 0, & \text{en otro caso.} \end{cases}$

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 $v : \text{VP} \to \mathbb{B}$.
  - Prop: Para cada valoración de verdad v existe una única aplicación $\hat{v} : \text{Prop} \to \mathbb{B}$ 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.

Valoración de fórmulas

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

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

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

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

- 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 $\hat{v}(F)$.

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 $v_1 \models (p \vee q) \wedge (\neg q \vee r)$
    – si $v_2(r) = 1, v_2(p) = v_2(q) = 0$, entonces $v_2 \not\models (p \vee q) \wedge (\neg q \vee r)$.
- Fórmulas satisfacibles e insatisfacibles
  - Def.: F es satisfacible si F tiene algún modelo.
  - Ejemplo: $(p \to q) \wedge (q \to r)$ es satisfacible
    $v(p) = v(q) = v(r) = 0$.
  - 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

- 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.

$p \to q$ es satisfacible.

$$v(p) = v(q) = 1$$

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

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

Algoritmos para SAT y TAUT

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

p	q	$(p \to q)$	$(q \to p)$	$(p \to q) \lor (q \to p)$
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	$(p$	$\to$	$q)$	$\lor$	$(q$	$\to$	$p)$
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 $\models (p \to q) \lor (q \to p)$

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

  - Método de Quine para $\models (p \to q) \lor (q \to p)$

$$\begin{array}{ccccccc}
(p & \to & q) & \lor & (q & \to & p) \\
0 & 0 & 1 & 0 & 1 & 0 & 0 \\
 & 1* & & & & &
\end{array}$$

Algoritmos para SAT y TAUT

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

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

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

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

Selección de tautologías

- Selección de tautologías
  - 1. $F \to F$ (ley de identidad).
  - 2. $F \lor \neg F$ (ley del tercio excluso).
  - 3. $\neg(F \land \neg F)$ (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. $(F \to G) \land F \to G$ (modus ponens).
  - 8. $(F \to G) \land \neg G \to \neg F$ (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 $S = \{(p \lor q) \land (\neg q \lor r), q \to r\}$
    La valoración v_1 tal que $v_1(p) = 1, v_1(q) = 0, v_1(r) = 1$ es modelo de S ($v_1 \models S$).

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

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

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

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:
    - $\{(p \vee q) \wedge (\neg q \vee r), p \rightarrow r\}$ es consistente (con modelos v_4, v_6, v_8)
    - $\{(p \vee q) \wedge (\neg q \vee r), p \rightarrow r, \neg r\}$ es inconsistente

	p	q	r	$(p \vee q)$	$(\neg q \vee r)$	$(p \vee q) \wedge (\neg q \vee r)$	$p \rightarrow r$	$\neg r$
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: $\{p \rightarrow q, q \rightarrow r\} \models p \rightarrow r$ y $\{p\} \not\models p \wedge q$

	p	q	r	$p \rightarrow q$	$q \rightarrow r$	$p \rightarrow r$
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	$p \wedge q$
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).