Sintaxis y Semántica de la Lógica Proposicional

Dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

Contenido

Problema básico

Dados un conjunto de **afirmaciones** (hechos, hipótesis,...), $\mathcal{BC}$, y una afirmación, A, decidir si A ha de ser necesariamente cierta, suponiendo que todos las afirmaciones de $\mathcal{BC}$ lo son.

La **Lógica** proporciona formulaciones precisas de este problema y diferentes soluciones.
Para abordar este problema formalmente hemos de:

1. Dar un lenguaje para expresar las afirmaciones (*representación*).
2. Concretar qué entendemos por *afirmación cierta*.
3. Proporcionar mecanismos efectivos para garantizar la *corrección de las deducciones*.

A lo largo de este curso estudiaremos estas cuestiones en los dos casos más comunes: la **Lógica Proposicional**, y la **Lógica de Primer Orden**.

Lógica Proposicional

Características generales de la Lógica Proposicional:

- Sus expresiones (fórmulas) representan *afirmaciones* que pueden considerarse *verdaderas* o *falsas*.
- Se construyen a partir de expresiones básicas usando operadores (*conectivas*).
- Las conectivas se corresponden con formas sencillas de construir afirmaciones complejas en el lenguaje natural partiendo de otras más sencillas:
  - Conjunción: "…tal …**y**…cual…"
  - Disyunción: "…tal …**o**…cual …"
  - Implicación "**Si** …tal …**entonces**…cual …"
  - Negación: "**No** es cierto que tal …"
- Sólo permite analizar las formas de razonamiento ligadas a este tipo de construcciones.

El Lenguaje de la Lógica Proposicional

Formalmente, el *lenguaje de la Lógica Proposicional* consta de:

1. Un conjunto numerable de **variables proposicionales**: $VP = \{p, p_0, p_1, \ldots, q, q_0, q_1, \ldots, r, r_0, \ldots\}$
2. Operadores básicos de contrucción, **Conectivas lógicas**:
   - De aridad 1: $\neg$ (negación).
   - De aridad 2: $\vee$ (disyunción), $\wedge$ (conjunción), $\rightarrow$ (condicional) y $\leftrightarrow$ (bicondicional).
3. **Símbolos auxiliares**: "(" y ")".

Fórmulas

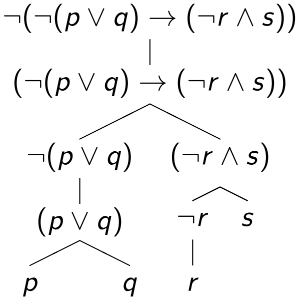
El conjunto de las *fórmulas proposicionales*, PROP, es el menor conjunto de expresiones que verifica:

- $VP \subseteq$ PROP,
- Es cerrado bajo las conectivas, es decir:
  - Si $F \in$ PROP, entonces $\neg F \in$ PROP.
  - Si $F, G \in$ PROP, entonces $(F \vee G)$, $(F \wedge G)$, $(F \rightarrow G)$, $(F \leftrightarrow G) \in$ PROP.

La sintaxis del lenguaje pretende evitar la ambigüedad en la interpretación de las fórmulas. Esa es la función de los símbolos auxiliares.

Árboles de formación

- Asociamos a cada fórmula un *árbol de formación* (esencialmente único) que describe el modo en que se construye la fórmula a partir de otras más sencillas.
- **Ejemplo**:



- Las fórmulas que aparecen en el árbol de formación de una fórmula F se denominan **subfórmulas** de F.

Reducción de paréntesis

Para facilitar la lectura de las fórmulas adoptaremos los siguientes convenios de notación:

1. Omitiremos los paréntesis externos.
2. Daremos a las conectivas una precedencia de asociación. De mayor a menor, están ordenadas por: $\neg, \wedge, \vee, \rightarrow$.
   - **Ejemplo**: $F \wedge G \rightarrow \neg F \vee G$ es $((F \wedge G) \rightarrow (\neg F \vee G))$.
3. Siempre se dejarán los paréntesis para la conectiva $\leftrightarrow$.
4. Cuando una conectiva se usa repetidamente, se asocia por la derecha: $F \vee G \vee H$ es $(F \vee (G \vee H))$.

Principio de Inducción sobre fórmulas

Gracias a la definición de PROP si deseamos probar que toda fórmula proposicional satisface cierta propiedad Ψ, podemos probarlo por **inducción sobre fórmulas**.

Para ello probamos:

1. **Caso base:** Todos los elementos de VP tienen la propiedad Ψ.
2. **Paso de inducción:**
   2.1 Si $F \in \text{PROP}$ tiene la propiedad Ψ, entonces $\neg F$ tiene la propiedad Ψ.
   2.2 Si $F, G \in \text{PROP}$ tienen la propiedad Ψ, entonces las fórmulas $(F \vee G)$, $(F \wedge G)$, $(F \to G)$ y $(F \leftrightarrow G)$ también tienen la propiedad Ψ.

Funciones de verdad

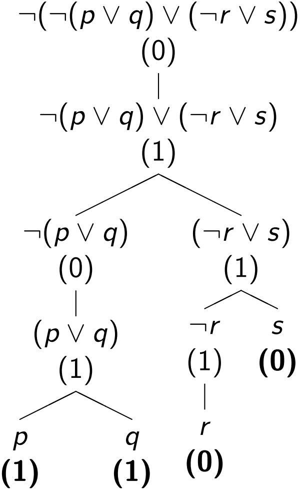
- Los elementos del conjunto $\{0, 1\}$ se llaman **valores de verdad**. Se dice que 0 es el valor **falso** y el 1 es el valor **verdadero**.
- El significado de una conectiva se determina mediante su **función de verdad**:
  - $H_\neg(i) = \begin{cases} 1, & \text{si } i = 0; \\ 0, & \text{si } i = 1. \end{cases}$
  - $H_\vee(i,j) = \begin{cases} 0, & \text{si } i = j = 0; \\ 1, & \text{en otro caso.} \end{cases}$
  - $H_\wedge(i,j) = \begin{cases} 1, & \text{si } i = j = 1; \\ 0, & \text{en otro caso.} \end{cases}$
  - $H_\to(i,j) = \begin{cases} 0, & \text{si } i = 1, j = 0; \\ 1, & \text{en otro caso.} \end{cases}$
  - $H_\leftrightarrow(i,j) = \begin{cases} 1, & \text{si } i = j; \\ 0, & \text{en otro caso.} \end{cases}$

Valoraciones

- Las variables proposicionales se interpretan mediante una **valoración de verdad** (o interpretación), es decir, una aplicación
$$v : VP \to \{0,1\}$$
- Podemos extender cada valoración, v, **de forma única**, al conjunto de todas las fórmulas de manera que para cada fórmula F se verifique:
  - $v(\neg F) = H_\neg(v(F))$.
  - $v((F \vee G)) = H_\vee(v(F), v(G))$.
  - $v((F \wedge G)) = H_\wedge(v(F), v(G))$.
  - $v((F \to G)) = H_\to(v(F), v(G))$.
  - $v((F \leftrightarrow G)) = H_\leftrightarrow(v(F), v(G))$.
- Se dice que $v(F)$ es el **valor de verdad** de F respecto de v.

Valor de verdad

Veamos el cálculo de $v(\neg(\neg(p \vee q) \vee (\neg r \vee s))$ en el árbol de formación (de abajo a arriba):



Tablas de verdad

Dada una valoración, v, el valor de verdad de una fórmula F respecto de v está determinado por los valores de verdad de las subfórmulas de F.

Ejemplo: si $v(p) = v(q) = 0$ y $v(r) = 1$, entonces

$$\begin{aligned} v(\neg((p \to q) \vee r)) &= H_\neg(H_\vee(v(p \to q), v(r))) = \\ &= H_\neg(H_\vee(H_\to(v(p), v(q)), 1)) = 0 \end{aligned}$$

Fijada v podemos presentar el cálculo de F mediante una tabla que recorre los valores de sus subfórmulas:

p	q	r	$p \to q$	$(p \to q) \vee r$	$\neg((p \to q) \vee r)$
0	0	1	1	1	0

Una **tabla de verdad** para F es una tabla similar que contiene una fila por cada posible valoración que asigne valores a las variables proposicionales que aparecen en F.

Validez y satisfactibilidad (I)

- Decimos que una fórmula F es **válida en** v, o que v es un **modelo** de F, si $v(F) = 1$.
  - Notación: $v \models F$.
  - Una valoración v es *modelo* de un conjunto de fórmulas U, $v \models U$, si v es modelo de todas las fórmulas de U.
- Una fórmula F es una **tautología** (o **válida**) si es válida para toda valoración (notación $\models F$).
- Una fórmula F es **satisfactible** (o consistente) si existe una valoración que es modelo de F. En caso contrario diremos que es **insatisfactible** (o inconsistente).
  - Análogamente, un conjunto de fórmulas U es satisfactible (o consistente) si existe una valoración que es modelo de U. En caso contrario diremos que es insatisfactible (o inconsistente).

Validez y satisfactibilidad (II)

Relación entre ambos conceptos:

Lema. Para cada $F \in \text{PROP}$ se verifica:

- Si F es un tautología entonces F es satisfactible.
- F es una tautología si y sólo si $\neg F$ insatisfactible.

Ejemplos:

- Son tautologías: $(p \vee \neg p)$ y $((p \to q) \to p) \to p$.
- $p \wedge \neg p$ es insatisfactible y, por tanto, $\neg(p \wedge \neg p)$ es una tautología.
- $(p \to q) \to p$ es satisfactible pero no es una tautología.

Consecuencia Lógica

- Una fórmula F es **consecuencia lógica** de un conjunto de fórmulas U, si todo modelo de U es modelo de F. Es decir, para toda valoración, v,

$$v \models U \quad \Longrightarrow \quad v \models F$$

  - Notación: $U \models F$.
  - La relación de consecuencia lógica permite formular el problema básico en el marco de la lógica proposicional.

Relación entre consecuencia lógica, consistencia y validez:

Proposición. Sea $\{F_1, \ldots F_n\} \subseteq \text{PROP}$. Son equivalentes:

- $\{F_1, \ldots, F_n\} \models F$
- $F_1 \wedge \cdots \wedge F_n \to F$ es un tautología.
- $\{F_1, \ldots, F_n, \neg F\}$ es insatisfactible.

Algoritmos de decisión (I)

Dado un conjunto de fórmulas proposicionales, U, un **algoritmo de decisión** para U es un algoritmo que dada $A \in PROP$, devuelve `SI` cuando $A \in U$, y `NO` si $A \notin U$.

Casos especialmente interesantes:

- SAT $= \{A \in PROP : \; A \text{ es satisfactible}\}$
- TAUT $= \{A \in PROP : \; A \text{ es una tautología}\}$
- Fijado $U \subseteq PROP$, la **Teoría de** U es

$$\mathcal{T}(U) = \{A \in PROP : \; U \models A\}$$

Un algoritmo de decisión para $\mathcal{T}(U)$ propociona una respuesta al Problema Básico expuesto al principio del tema.

Algoritmos de decisión (II)

Problema Básico:
Obtener un algoritmo que, dado un conjunto finito de fórmulas proposicionales, U, y una fórmula F, decida si $U \models F$.

El problema anterior se reduce a decidir la **satisfactibilidad** de una cierta fórmula (o si se prefiere, la **validez** de otra). Por tanto,

- La construcción de tablas de verdad proporciona un algoritmo (ineficiente) para decidir la consecuencia lógica.
- El Problema Básico es resoluble algorítmicamente, aunque no se conoce ninguna solución eficiente y se duda de la existencia de algoritmos de decisión eficientes para este problema, ya que ...
- ... determinar la satisfactibilidad de una fórmula proposicional es un problema **NP-completo**.

Algoritmos de decisión (III)

Problema Básico (bis):
Obtener un algoritmo eficiente que, dado un conjunto finito de fórmulas proposicionales, U, y una fórmula F, decida si $U \models F$.

Observaciones:

- Este problema es equivalente al de obtener un algoritmo eficiente para determinar la satisfactibilidad de una fórmula proposicional.
- Se trata de un **problema abierto**, que posiblemente tendrá una respuesta negativa (se cree que no existen algoritmos eficientes para resolver SAT).
- Para propósitos prácticos puede bastar con algoritmos eficientes para alguna clase especial de fórmulas.

Limitaciones de la lógica proposicional

- La lógica proposicional posee un semántica sencilla y existen algoritmos de decisión para sus problemas básicos, como SAT o la consecuencia lógica.
- Sin embargo, la expresividad de la lógica proposicional es bastante limitada.
- Existen problemas cuya descripción mediante lógica proposicional es complicada, ya que requieren un gran número de fórmulas o bien fórmulas de gran tamaño.
- Más aún, existen formas de razonamiento válido que no pueden ser expresadas mediante la lógica proposicional, por ejemplo:
  - Todos los hombres son mortales
  - Sócrates es un hombre.
  - Por tanto, Sócrates es mortal.
- La Lógica de Primer Orden extiende a la Lógica Proposicional proporcionando mayor expresividad.