

1.1. Propositiones y argumentos

La lógica es la disciplina que estudia los principios y métodos para distinguir razonamientos correctos de los incorrectos. Es una herramienta fundamental en la ciencia de la computación, ya que nos permite formalizar y analizar rigurosamente la estructura de los argumentos, la validez de las inferencias y la correctitud de los sistemas computacionales. En este capítulo estudiaremos sintaxis, semántica e inferencia: qué expresiones podemos escribir, qué significan y cuándo una conclusión se sigue de ciertas premisas.

A modo de motivación consideremos los siguientes dos argumentos:

- | | |
|---|---|
| 1. Una intensa ola de frío está afectando a gran parte de USA.
<hr/> 2. Por tanto, el calentamiento global no está ocurriendo. | 1. Todo déficit comercial es una pérdida monetaria.
2. Los estados unidos tienen un déficit comercial.
<hr/> 3. Por lo tanto, USA está sufriendo una pérdida monetaria. |
|---|---|

De manera intuitiva, el primero no es un razonamiento válido: una ola de frío localizada no permite concluir que el calentamiento global no está ocurriendo. El segundo tiene es un razonamiento válido: si todo déficit comercial es una pérdida monetaria y USA tiene un déficit comercial, entonces podemos concluir que USA está sufriendo una pérdida monetaria. Notemos que esto es independiente de que las premisas sean verdaderas o falsas: el argumento es válido o inválido incluso si USA no tiene un déficit comercial, o si el calentamiento global no está ocurriendo.

La lógica estudia esta forma del razonamiento, independientemente de que las premisas describan correctamente el mundo.

Existe una gran variedad de sistemas lógicos, cada uno con su propia sintaxis y semántica. En la primera parte del curso nos enfocaremos en la *lógica proposicional*, donde la unidad básica de análisis es una proposición.

Definición. Una *proposición* es un enunciado declarativo que sin ambigüedad es verdadero o falso, pero no ambos a la vez.

Para separar la condición de ser una proposición de su valor de verdad, consideremos las siguientes oraciones:

- Santiago es la capital de Chile.
- ¿Cuál es la capital de Chile?
- $2 + 2 = 5$.
- Silencio, por favor.
- Hoy llueve en Concepción.
- $x + 2 = 4$.
- Esta oración es falsa.

La primera, tercera y quinta son proposiciones, mientras que las demás no lo son. Que $2 + 2 = 5$ sea falsa no impide que sea una proposición; de la misma forma, el valor de verdad de la oración sobre la lluvia depende del día en que se evalúe, pero en ese contexto tiene un valor de verdad determinado. La segunda oración es una pregunta y la cuarta es una orden (no son enunciados declarativos), por lo que no son proposiciones. La sexta contiene una variable cuyo valor no ha sido fijado, mientras que la última conduce a una contradicción al intentar asignarle cualquiera de los dos valores de verdad.

Llamaremos *proposición atómica* a una proposición que tratamos como una unidad básica, sin descomponerla en proposiciones más pequeñas. Para abstraernos de su contenido usaremos *variables proposicionales*, que usualmente denotaremos por letras minúsculas como p, q, r y s .

1.1.1. Operadores lógicos

Dadas proposiciones atómicas, podemos construir proposiciones de mayor complejidad, conocidas como *proposiciones compuestas*, usando conectivos u operadores lógicos.

Definición. Dadas p y q proposiciones, definimos las siguientes proposiciones compuestas:

- La *negación* $\neg p$ es verdadera si y solo si p es falsa.
- La *conjunción* $p \wedge q$ es verdadera si y solo si p y q son ambas verdaderas.
- La *disyunción* $p \vee q$ es verdadera si y solo si al menos una entre p y q es verdadera, incluyendo el caso en que ambas lo son.
- La *implicancia* $p \rightarrow q$ es falsa si y solo si p es verdadera y q es falsa.
- La *equivalencia* $p \leftrightarrow q$ es verdadera si y solo si p y q tienen el mismo valor de verdad.

Por ejemplo, definamos las variables proposicionales

p : Santiago es la capital de Chile,

q : $2 + 2 = 5$,

r : hoy llueve en Concepción.

Podemos combinarlas para obtener las siguientes proposiciones:

- $\neg p$: “Santiago no es la capital de Chile”, que es falsa.
- $p \wedge q$: “Santiago es la capital de Chile y $2 + 2 = 5$ ”, que es falsa.
- $p \vee \neg q$: “Santiago es la capital de Chile o $2 + 2 \neq 5$ ”, que es verdadera.
- $p \rightarrow q$: “Si Santiago es la capital de Chile, entonces $2 + 2 = 5$ ”, que es falsa.

1.1.2. Fórmulas y su sintaxis

La sintaxis de la lógica proposicional corresponde a las reglas que determinan qué expresiones están bien formadas a partir de variables proposicionales y operadores lógicos. Podemos pensar en ella como las reglas que determinan qué expresiones “compilan”.

Definición. Las *fórmulas bien formadas* (FBF) se definen recursivamente de la siguiente manera:

- Si p es una variable proposicional, entonces p es una FBF.
- Si α y β son FBF, entonces $(\neg\alpha)$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$ y $(\alpha \rightarrow \beta)$ son FBF.

La definición anterior comienza con las variables proposicionales y permite aplicar sus reglas una cantidad finita de veces. Ninguna otra expresión es una FBF. Usaremos las letras griegas α , β y γ para representar fórmulas arbitrarias. Aunque $\leftrightarrow$ no aparece entre los conectivos de la definición recursiva, lo podemos tratar como una abreviación:

$$(\alpha \leftrightarrow \beta) \text{ representa a } ((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)).$$

Supongamos que p, q, r son variables proposicionales. La expresión

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

es una FBF: podemos construir primero $(p \vee q)$ y $(q \vee r)$, luego $((p \vee q) \rightarrow q)$ y $(\neg(q \vee r))$, y finalmente su conjunción. En cambio,

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

no es una FBF, pues hay conectivos binarios sin una fórmula a cada lado y los paréntesis no están balanceados. La expresión $(p \rightarrow (q \wedge r))$ sí es una FBF. Por otra parte, $(p \vee q) \rightarrow r$ no es literalmente producida por la definición anterior, que exige paréntesis alrededor de cada conectivo binario, pero la usaremos como abreviación de $((p \vee q) \rightarrow r)$.

Para evitar lo anterior y el exceso de paréntesis, usaremos un orden de precedencia para los conectivos:

$$\neg > \wedge > \vee > \rightarrow.$$

Esto se debe leer como “la negación tiene mayor precedencia que la conjunción, que tiene mayor precedencia que la disyunción, que tiene mayor precedencia que la implicancia”. A pesar de esto, siempre podemos usar paréntesis cuando eviten confusión.

Otra simplificación es la omisión de los paréntesis exteriores de una fórmula. No adoptaremos una convención de asociatividad para cadenas de implicancias, donde mantendremos los paréntesis necesarios. Con estas convenciones, podemos abreviar los siguientes ejemplos.

Fórmula completamente parentizada	Forma abreviada
$((p \vee q) \wedge r)$	$(p \vee q) \wedge r$
$((p \rightarrow q) \vee (r \rightarrow s))$	$(p \rightarrow q) \vee (r \rightarrow s)$
$((p \wedge q) \rightarrow r)$	$p \wedge q \rightarrow r$
$((p \vee q) \rightarrow (r \vee s))$	$p \vee q \rightarrow r \vee s$

En la segunda fila los paréntesis interiores son necesarios. La precedencia obliga a agrupar $q \vee r$ antes que las implicancias y no hemos fijado cómo agrupar dos implicancias consecutivas; por lo tanto, $p \rightarrow q \vee r \rightarrow s$ no representa la fórmula original.

1.1.3. Fórmulas y su semántica

La semántica de la lógica proposicional determina el significado de una FBF a partir del significado de sus variables proposicionales. Para esto, primero asignamos valores de verdad a las variables y luego extendemos esta asignación recursivamente a todas las fórmulas. Representaremos falso por 0 y verdadero por 1.

Definición. Sean $p_1, \dots, p_n$ variables proposicionales. Una *valuación* de $\{p_1, \dots, p_n\}$ es una función $v: \{p_1, \dots, p_n\} \rightarrow \{0, 1\}$ que asocia un valor de verdad a cada variable proposicional.

Por ejemplo, para las variables p, q, r podemos considerar las valuaciones v_1 y v_2 dadas por

$$v_1(p) = 1, \quad v_1(q) = 1, \quad v_1(r) = 0, \quad v_2(p) = 1, \quad v_2(q) = 0, \quad v_2(r) = 1.$$

Estas funciones todavía no asignan directamente un valor a una fórmula compuesta. La definición siguiente nos dice cómo hacerlo usando su construcción sintáctica.

Definición. Sean $p_1, \dots, p_n$ variables proposicionales, sea $v: \{p_1, \dots, p_n\} \rightarrow \{0, 1\}$ una valuación y sea $\mathcal{F}$ el conjunto de FBF cuyas variables pertenecen a $\{p_1, \dots, p_n\}$. La *extensión de la valuación v* es la función $\hat{v}: \mathcal{F} \rightarrow \{0, 1\}$ definida recursivamente por

$$\begin{aligned} \hat{v}(p) &= v(p) && \text{para cada } p \in \{p_1, \dots, p_n\}, \\ \hat{v}(\neg\alpha) &= 1 - \hat{v}(\alpha), \\ \hat{v}(\alpha \wedge \beta) &= \min\{\hat{v}(\alpha), \hat{v}(\beta)\}, \\ \hat{v}(\alpha \vee \beta) &= \max\{\hat{v}(\alpha), \hat{v}(\beta)\}, \\ \hat{v}(\alpha \rightarrow \beta) &= \max\{1 - \hat{v}(\alpha), \hat{v}(\beta)\}. \end{aligned}$$

Evaluemos la fórmula $(p \wedge q) \rightarrow r$ bajo las dos valuaciones anteriores. Para v_1 , primero obtenemos

$$\hat{v}_1(p \wedge q) = \min\{1, 1\} = 1,$$

y luego

$$\hat{v}_1((p \wedge q) \rightarrow r) = \max\{1 - 1, 0\} = 0.$$

Para v_2 , el mismo cálculo da

$$\hat{v}_2(p \wedge q) = \min\{1, 0\} = 0, \quad \hat{v}_2((p \wedge q) \rightarrow r) = \max\{1 - 0, 1\} = 1.$$

Otra manera de describir la semántica de los operadores es mediante sus *tablas de verdad*. Las reglas de la definición anterior producen la tabla siguiente:

α	β	$\neg\alpha$	$\alpha \wedge \beta$	$\alpha \vee \beta$	$\alpha \rightarrow \beta$
0	0	1	0	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	0	1	1	1

Como $\alpha \leftrightarrow \beta$ abrevia $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$, su valor es 1 precisamente cuando α y β tienen el mismo valor.

Podemos calcular la tabla de verdad de una fórmula más compleja evaluando primero sus subfórmulas. Para $(p \wedge q) \rightarrow r$ conviene agregar una columna para $p \wedge q$:

p	q	r	$p \wedge q$	$(p \wedge q) \rightarrow r$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

Notemos que la fórmula es falsa solamente cuando p y q son verdaderas y r es falsa, tal como indica la definición de la implicancia.

1.1.4. Propiedades de las fórmulas proposicionales

Las tablas de verdad nos permiten distinguir fórmulas cuyo valor no depende de la valuación escogida.

Definición. Sea α una FBF. Diremos que α es una *tautología* si, para toda valuación v de las variables de α , tenemos $\hat{v}(\alpha) = 1$. Diremos que α es una *contradicción* si, para toda valuación v de las variables de α , tenemos $\hat{v}(\alpha) = 0$.

Consideremos las fórmulas $p \vee \neg p$ y $p \wedge \neg p$. Sus valores aparecen en la tabla siguiente:

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

Por lo tanto, $p \vee \neg p$ es una tautología y $p \wedge \neg p$ es una contradicción.

Hasta ahora hemos usado fórmulas para representar proposiciones individuales. También podemos usarlas para precisar cuándo un razonamiento es válido.

Definición. Un *argumento* consiste en fórmulas $P_1, \dots, P_n$, llamadas *premisas*, y una fórmula C , llamada *conclusión*. El argumento es *válido* si toda valuación que hace verdaderas todas las premisas también hace verdadera la conclusión; es decir, si para toda valuación v ,

$$\hat{v}(P_1) = \dots = \hat{v}(P_n) = 1 \implies \hat{v}(C) = 1.$$

Analicemos los dos argumentos siguientes:

- | | |
|--|---|
| 1. La luna es un planeta o 2 es impar.
2. La luna no es un planeta.
3. Por lo tanto, 2 es impar. | 1. Si Santiago está en Chile, entonces está en Sudamérica.
2. Santiago está en Sudamérica.
3. Por lo tanto, Santiago está en Chile. |
|--|---|

En el argumento de la izquierda, sea p la proposición “la luna es un planeta” y sea q la proposición “2 es impar”. Las premisas son $p \vee q$ y $\neg p$, y la conclusión es q . No existe una valuación que haga verdaderas ambas premisas y falsa la conclusión: si $\neg p$ es verdadera, entonces p es falsa, y para que $p \vee q$ sea verdadera debemos tener que q es verdadera. Por lo tanto, el argumento es válido.

En el argumento de la derecha, sea p la proposición “Santiago está en Chile” y sea q la proposición “Santiago está en Sudamérica”. Las premisas son $p \rightarrow q$ y q , y la conclusión es p . La valuación $v(p) = 0$ y $v(q) = 1$ hace verdaderas ambas premisas y falsa la conclusión, por lo que el argumento no es válido. Este ejemplo muestra por qué saber que una implicancia y su consecuente son verdaderos no basta para deducir su antecedente.

La definición de argumento válido también puede escribirse como una propiedad de una sola fórmula. El argumento con premisas $P_1, \dots, P_n$ y conclusión C es válido si y solo si

$$(P_1 \wedge \dots \wedge P_n) \rightarrow C$$

es una tautología. Aquí la conjunción de las premisas se agrupa de izquierda a derecha según nuestra convención. En efecto, esta implicancia solamente podría ser falsa bajo una valuación que hiciera verdaderas todas las premisas y falsa la conclusión, que es precisamente lo que la validez prohíbe.

1.1.5. Ejercicios

1. Considere el argumento

$$\begin{array}{c} p \rightarrow q \\ q \rightarrow r \\ \hline p \\ \hline r \end{array}$$

Demuestre que el argumento es válido. ¿Puede eliminar algunas de las premisas y mantener la validez del argumento? Justifique.

2. Considere un sistema de despliegue de software con las siguientes proposiciones:

- S : el despliegue satisface las condiciones de seguridad;
- R : la infraestructura requerida está disponible;
- E : un administrador emite una autorización de emergencia;
- Q : se alcanza el quórum de aprobaciones automáticas;
- D : el despliegue se ejecuta.

Las reglas del sistema son:

- R1.** Las condiciones de seguridad e infraestructura se cumplen si y solo si existe autorización de emergencia o se alcanza el quórum.
- R2.** Si se alcanza el quórum, el despliegue se ejecuta.
- R3.** Si existe autorización de emergencia, el despliegue se ejecuta.

- (a) Escriba las tres reglas como fórmulas proposicionales.
- (b) ¿Son consistentes las reglas con un despliegue ejecutado que no satisface las condiciones de seguridad? Justifique.
- (c) Demuestre que las reglas implican

$$S \wedge R \rightarrow D.$$

3. El operador *NOR*, denotado por $\downarrow$, se interpreta como “ni α ni β ”; es decir, $\alpha \downarrow \beta$ es verdadera exactamente cuando α y β son ambas falsas.
- (a) Defina inductivamente la sintaxis de las fórmulas construidas usando únicamente variables proposicionales, paréntesis y el operador $\downarrow$.
 - (b) Extienda una valuación v a estas fórmulas mediante una definición recursiva. Dé la regla para $\hat{v}(\alpha \downarrow \beta)$ tanto en palabras como usando una expresión.
 - (c) ¿El lenguaje que usa solamente $\downarrow$ es más expresivo, menos expresivo o igual de expresivo que la lógica proposicional que usa $\neg$, $\wedge$, $\vee$ y $\rightarrow$? Justifique.