Capítulo 1: Fundamentos: Lógica y Demostraciones

Clase 1: Lógica Proposicional

Matemática Discreta - CC3101
Profesor: Pablo Barceló

¿Qué es la lógica?

La lógica es el estudio de las leyes del pensamiento (Kant, 1785).

En la actualidad se considera que la lógica es el estudio de qué es lo que hace que un argumento se considere correcto (en forma y no en contenido).

En otros términos, de cuándo una conclusión se deduce lógicamente de ciertas premisas.

Ejemplo: ¿Es el siguiente argumento válido?

Todos los hombres son mortales.
Sócrates es hombre.

Por lo tanto, Sócrates es mortal.

¿Hay algo en este argumento que dependa de Sócrates mismo?

Ejemplo de razonamiento lógico

Ejercicio: ¿Cuál es la forma del siguiente argumento?

Si Pedro estudia en el DCC o en el CMM, entonces tomará CC4OC.
Pedro estudia en el DCC.

Por tanto, Pedro tomará CC4OC.

Ejemplo de razonamiento lógico

Ejercicio: ¿Cuál es la forma del siguiente argumento?

Si Pedro estudia en el DCC o en el CMM, entonces tomará CC4OC.
Pedro estudia en el DCC.

Por tanto, Pedro tomará CC4OC.

$$\begin{array}{l} (p \vee q) \rightarrow r \\ p \\ \hline r \end{array}$$

Aplicaciones de la lógica en CS

La lógica es la base de todo el razonamiento matemático, y también de todo el razonamiento automatizado.

Tiene aplicaciones práticas en CS en los siguientes campos (entre muchos otros):

- Diseño de hardware;
- ingeniería de software;
- bases de datos;
- inteligencia artificial;
- lenguajes de programación.

Lógica proposicional

Empezaremos con el ejemplo más básico de lenguaje lógico: la lógica proposicional.

Esta está construida a partir de proposiciones, que son oraciones que son verdaderas o falsas (pero no ambas).

- Santiago es la capital de Chile, $(1 + 1 = 3)$, $(1 + 1 = \frac{6}{3})$.

Las siguientes, en cambio, no son proposiciones:

- ¿Qué hora es?, $(x + 1 = 2)$.

Valores de verdad

Nos evitaremos problemas semánticos, y denotaremos nuestras proposiciones por letras minúsulas $p, q, r, \ldots, p_1, p_2, \ldots$.

Cada proposición tiene un valor de verdad asignado, que puede ser 1 (si la proposición es verdadera) o 0 (si la proposición es falsa).

Oraciones

Una oración se construye a partir de las proposiciones $p, q, r, \ldots,$ usando además tres símbolos nuevos: $\neg$, $\vee$, $\wedge$.

- Dada una proposición p, la oración $\neg p$ denota que 'p es falso'. Se llama la negación de p.
- Dadas proposiciones p y q, la oración $p \vee q$ denota que 'p o q', y se llama la disyunción de p y q.
- Dadas proposiciones p y q, la oración $p \wedge q$ denota que 'p y q', y se llama la conjunción de p y q.

En general los símbolos $\neg, \vee, \wedge$ se aplican no sólo a las proposiciones sino también a las oraciones:

- $(\neg(p \vee (q \wedge r))) \vee s$, $(p \vee q) \vee r$.

Paréntesis

En las oraciones anteriores utilizamos paréntesis para evitar ambiguedades en el orden de aplicación de los símbolos $\neg, \vee, \wedge$:

- ¿Qué significa $p \vee q \vee r$?

Usualmente asumimos que $\neg$ siempre se aplica antes que cualquier otra operación, y por tanto, $\neg p \wedge q$ es lo mismo que $(\neg p) \wedge q$.

Traducciones

Con los nuevos conectivos podemos traducir muchas de nuestras oraciones en español al lenguaje de la lógica proposicional.

Por ejemplo, 'Juan es alto pero flaco' puede traducirse como $'p \wedge q'$, donde:

- $p =$ 'Juan es alto'.
- $q =$ 'Juan es flaco'.

Ejercicio: ¿Cómo traduciría 'Juan no es alto ni flaco'?

Tablas de verdad

Las oraciones se forman entonces desde las proposiciones, usando los símbolos $\neg, \vee$ and $\wedge$.

Pero las oraciones como las proposiciones también tienen un valor de verdad: deben ser verdaderas (1) o falsas (0).

La idea es que el valor de verdad de una oración se puede determinar únicamente a partir del valor de verdad de las proposiciones que la componenen y de las tablas de verdad de los símbolos $\neg, \vee$ and $\wedge$.

Tabla de verdad de $\neg$

Si α es una oración, entonces el valor de verdad de $\neg\alpha$ es el contrario que el de α.

La tabla de verdad de $\neg\alpha$ es como sigue:

α	$\neg\alpha$
1	0
0	1

Tabla de verdad de $\alpha \vee \beta$

Si α y β son oraciones, entonces $\alpha \vee \beta$ es verdadero si y sólo si α o β son verdaderos.

La tabla de verdad de $\alpha \vee \beta$ es como sigue:

α	β	$\alpha \vee \beta$
1	1	1
1	0	1
0	1	1
0	0	0

Tabla de verdad de $\alpha \vee \beta$

Si α y β son oraciones, entonces $\alpha \vee \beta$ es verdadero si y sólo si α o β son verdaderos.

La tabla de verdad de $\alpha \vee \beta$ es como sigue:

α	β	$\alpha \vee \beta$
1	1	1
1	0	1
0	1	1
0	0	0

Pregunta: ¿Formaliza ésto lo que entendemos por disyunción en el lenguaje natural?

Tabla de verdad de $\alpha \wedge \beta$

Si α y β son oraciones, entonces $\alpha \wedge \beta$ es verdadero si y sólo si α y β son verdaderos.

La tabla de verdad de $\alpha \wedge \beta$ es como sigue:

α	β	$\alpha \wedge \beta$
1	1	1
1	0	0
0	1	0
0	0	0

Tabla de verdad de $\alpha \wedge \beta$

Si α y β son oraciones, entonces $\alpha \wedge \beta$ es verdadero si y sólo si α y β son verdaderos.

La tabla de verdad de $\alpha \wedge \beta$ es como sigue:

α	β	$\alpha \wedge \beta$
1	1	1
1	0	0
0	1	0
0	0	0

Pregunta: ¿Formaliza ésto lo que entendemos por conjunción en el lenguaje natural?

Tabla de verdad de oraciones más complejas

Ahora podemos construir la tabla de verdad de oraciones más complejas:

Ejercicio: Construya la tabla de verdad de $(p \vee q) \wedge \neg(p \wedge q)$.

¿Cuál es el 'significado' de esta oración?

Implicación lógica

Cuando decimos que hacemos una deducción lógica en el lenguaje natural, razonamos desde una hipótesis hasta una conclusión.

- Si tal y tal cosa son ciertas, entonces tal otra también es cierta.

Este tipo de argumentos se representan de la forma $\alpha \rightarrow \beta$ en nuestro lenguaje.

Intituitivamente, ¿cuándo una oración de la forma $\alpha \rightarrow \beta$ debería ser falsa?

Implicación lógica

Cuando decimos que hacemos una deducción lógica en el lenguaje natural, razonamos desde una hipótesis hasta una conclusión.

- Si tal y tal cosa son ciertas, entonces tal otra también es cierta.

Este tipo de argumentos se representan de la forma $\alpha \to \beta$ en nuestro lenguaje.

Intituitivamente, ¿cuándo una oración de la forma $\alpha \to \beta$ debería ser falsa?

Cuando α es verdadero y β es falso!

Tabla de verdad para $\rightarrow$

Por tanto, la tabla de verdad para el conectivo $\rightarrow$ es la siguiente:

α	β	$\alpha \rightarrow \beta$
1	1	1
1	0	0
0	1	1
0	0	1

Como veremos más tarde, $\rightarrow$ no le provee mayor poder expresivo a nuestro lenguaje inicial.

Tabla de verdad para $\rightarrow$

Por tanto, la tabla de verdad para el conectivo $\rightarrow$ es la siguiente:

α	β	$\alpha \rightarrow \beta$
1	1	1
1	0	0
0	1	1
0	0	1

Como veremos más tarde, $\rightarrow$ no le provee mayor poder expresivo a nuestro lenguaje inicial.

Pregunta: ¿Qué elemento llama la atención de la tabla de verdad de $\rightarrow$?

Equivalencia lógica

Decimos que las oraciones α y β son equivalentes, y lo denotamos por $\alpha \equiv \beta$, si las tablas de verdad de α y β son iguales.

Algunas equivalencias útiles:

$$\begin{aligned}
(\neg(\varphi \wedge \psi)) &\equiv ((\neg\varphi) \vee (\neg\psi)) & (\varphi \rightarrow \psi) &\equiv ((\neg\varphi) \vee \psi)\\
(\neg(\varphi \vee \psi)) &\equiv ((\neg\varphi) \wedge (\neg\psi)) & (\varphi \leftrightarrow \psi) &\equiv ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))\\
(\varphi \wedge (\psi \wedge \theta)) &\equiv ((\varphi \wedge \psi) \wedge \theta) & (\neg(\neg\varphi)) &\equiv \varphi\\
(\varphi \vee (\psi \vee \theta)) &\equiv ((\varphi \vee \psi) \vee \theta) & &
\end{aligned}$$

Ejercicio: ¿Es cierto que $\rightarrow$ es asociativo?

Consecuencia lógica

Llegamos a nuestra noción más importante, la que captura cuando una conclusión se sigue lógicamente desde un conjunto de premisas.

Sea Σ un conjunto de oraciones y α una oración. Decimos que α es consecuencia lógica de Σ, si cada vez que cada fórmula de Σ es verdadera entonces α también es verdadera.

En ese caso escribimos $\Sigma \models \alpha$.

Ejercicio: Demuestre que $\{p, p \to q\} \models q$ (modus ponens) y $\{\neg q, p \to q\} \models \neg p$ (modus tollens).

Demuestre que $\{p \to q, q \to r\} \models p \to r$ (transitividad).

Consecuencia lógica

Llegamos a nuestra noción más importante, la que captura cuando una conclusión se sigue lógicamente desde un conjunto de premisas.

Sea Σ un conjunto de oraciones y α una oración. Decimos que α es consecuencia lógica de Σ, si cada vez que cada fórmula de Σ es verdadera entonces α también es verdadera.

En ese caso escribimos $\Sigma \models \alpha$.

Ejercicio: Demuestre que $\{p, p \to q\} \models q$ (modus ponens) y $\{\neg q, p \to q\} \models \neg p$ (modus tollens).

Demuestre que $\{p \to q, q \to r\} \models p \to r$ (transitividad).

¿Puede encontrar ejemplos de este tipo de argumentación en el lenguaje natural o matemático?

Consecuencia lógica: Otras propiedades interesantes

Ejercicio: Demuestre que $\alpha \equiv \beta$ si y sólo si $\alpha \models \beta$ and $\beta \models \alpha$.

Decimos que un conjunto de oraciones Σ es insatisfacible si en ninguna fila de la tabla de verdad se tiene que todas las oraciones en Σ son verdaderas.

Ejercicio: Demuestre que $\Sigma \models \alpha$ si y sólo si $\Sigma \cup \{\neg\alpha\}$ es insatisfacible.

Ejercicio: Demuestre que Σ es insatisfacible si y sólo si $\Sigma \models \alpha$, para cualquier oración α.

Ejercicio: Demuestre que $\Sigma \models \alpha \rightarrow \beta$ si y sólo si $\Sigma \cup \{\alpha\} \models \beta$ (regla de deducción).

Consecuencia lógica: Monotonía

Ejercicio: Si $\Sigma \models \alpha$, entonces para cada oración β se tiene que $\Sigma \cup \{\beta\} \models \alpha$ (monotonía).

Ejercicio: Si $\Sigma \models \alpha$, entonces para cada oración β se tiene que $\Sigma \cup \{\beta\} \models \alpha$ (monotonía).

Sabemos que $\{p,\ p \rightarrow q\} \models q$. Usando el teorema de monotonía deducimos que $\{p,\ p \rightarrow q,\ \neg q\} \models q$. ¿Cómo es esto posible?

Consecuencia lógica: Monotonía

Ejercicio: Si $\Sigma \models \alpha$, entonces para cada oración β se tiene que $\Sigma \cup \{\beta\} \models \alpha$ (monotonía).

Sabemos que $\{p,\ p \rightarrow q\} \models q$. Usando el teorema de monotonía deducimos que $\{p,\ p \rightarrow q,\ \neg q\} \models q$. ¿Cómo es esto posible?

¿Puede usarse la lógica proposicional para modelar razonamiento con sentido común?

Ejercicios finales

Ejercicio: Demuestre que $\neg$, $\vee$ son suficientes para expresar todas las oraciones de la lógica proposicional (i.e. todas las oraciones que utilizan conectivos $\neg, \vee, \wedge, \rightarrow$).

Ejercicio: Demuestre que $\wedge$, $\vee$ no son suficientes para expresar todas las oraciones de la lógica proposicional sobre variables propisicionales p y q.

Circuitos digitales

¿Por qué hasta aquí hemos ocupado 1 y 0 para denotar verdadero y falso?

Porque estos valores también representan a los dos valores binarios.

La lógica proposicional también puede ser utilizada para diseñar circuitos digitales: transformen secuencias de señales de 1s y 0s en otras secuencias de señales de 1s y 0s (idea de Shannon, '38).

Por ejemplo, un sumador (lo veremos luego).

Compuertas

Un circuto digital se piensa abstractamente como una caja negra que establece una relación entre ciertas entradas y la salida:

La operación del circuito se halla completamente especificada al construir una tabla entrada/salida que liste todos los posibles valores de entrada con su respectivo valor de salida:

P	Q	R
1	1	0
1	0	1
0	1	0
0	0	1

Una tabla entrada/salida se ve igual que una tabla de verdad.

Probablemente entonces, los circuitos digitales puedan ser representados por oraciones de la lógica proposicional.

Tablas de verdad y oraciones

Una tabla entrada/salida se ve igual que una tabla de verdad.

Probablemente entonces, los circuitos digitales puedan ser representados por oraciones de la lógica proposicional.

Ejercicio: Encuentre la oración que representa al circuito digital del ejemplo anterior.

Asumamos el caso general en el que nuestra circuto digital está dado por la siguiente tabla entrada/salida:

p_1	p_2	$\cdots$	p_{n-1}	p_n	$C(p_1, p_2, \ldots, p_{n-1}, p_n)$
0	0	$\cdots$	0	0	b_1
0	0	$\cdots$	0	1	b_2
$\vdots$	$\vdots$	$\ldots$	$\vdots$	$\vdots$	$\vdots$
1	1	$\cdots$	1	1	b_{2^n}

¿Qué oración de la lógica proposicional tiene exactamente esta tabla de verdad?

Tablas de verdad y oraciones

Asumiendo que σ_i es la valuación correspondiente a la fila i de la tabla de verdad de $C(p_1, \ldots, p_n)$, este conectivo es equivalente a:

$$\bigvee_{i\,:\,b_i=1} \left(\left(\bigwedge_{j\,:\,\sigma_i(p_j)=1} p_j \right) \wedge \left(\bigwedge_{k\,:\,\sigma_i(p_k)=0} \neg p_k \right) \right).$$

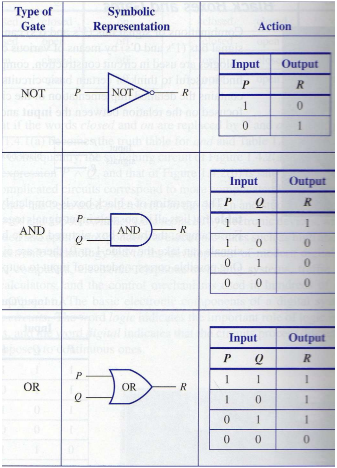
Tablas de verdad y oraciones

Asumiendo que σ_i es la valuación correspondiente a la fila i de la tabla de verdad de $C(p_1, \ldots, p_n)$, este conectivo es equivalente a:

$$\bigvee_{i\,:\,b_i=1} \Bigg(\Bigg(\bigwedge_{j\,:\,\sigma_i(p_j)=1} p_j \Bigg) \wedge \Bigg(\bigwedge_{k\,:\,\sigma_i(p_k)=0} \neg p_k \Bigg) \Bigg).$$

Conclusión: Basta con los conectivos lógicos $\neg, \vee, \wedge$ para representar cualquier tabla de verdad (y, por tanto, cualquier circuto digital).

Para representar una oración de la lógica proposicional como circuito digital utilizamos las siguientes compuertas lógicas:



Type of Gate	Symbolic Representation	Action
NOT	P — NOT — R	see table NOT
AND	P, Q — AND — R	see table AND
OR	P, Q — OR — R	see table OR

NOT:

Input	Output
P	R
1	0
0	1

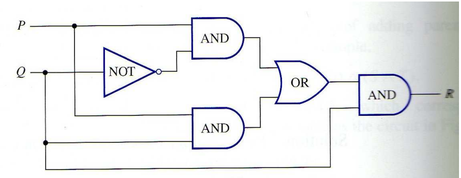
AND:

Input		Output
P	Q	R
1	1	1
1	0	0
0	1	0
0	0	0

OR:

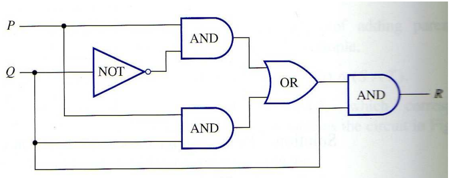
Input		Output
P	Q	R
1	1	1
1	0	1
0	1	1
0	0	0

El siguiente es un ejemplo de un circuito digital construido utilizando las compuertas lógicas:



(a)

El siguiente es un ejemplo de un circuito digital construido utilizando las compuertas lógicas:



(a)

Pregunta: ¿Cuál es la tabla entrada/salida de este circuito?

Notación binaria

Recordemos que todo número natural puede ser representado en notación binaria, i.e. de la forma $d_n d_{n-1} \cdots d_0$, donde cada d_i $(0 \leq i \leq n)$ es el bit 0 o 1.

Pregunta: ¿Cuál es el número binario que es equivalente al número natural m?

- $d_n d_{n-1} \cdots d_0$ tal que $\sum_{i=0}^{n} d_i \cdot 2^i = m$ and $d_n = 1$.

Ejemplo: 11011 es el equivalente a 27 en notación binaria.

Ejercicio: Represente los siguientes números naturales en notación binaria: 19, 458, 287, 55.

Ejercicio: Represente los soiguientes números binarios en notación decimal: 10111, 110111, 10110011.

Circuito digital para la suma

Ejercicio: ¿Cómo se resta en binario?

Ejercicio: Construya un circuito digital con $2n$ entradas y $n + 1$ salidas que compute la suma binaria de dos números binarios de n bits.

Ejercicio: ¿Cómo se multiplica en binario? Construya un circuito (más bien, una familia) que lo haga.