Tema 2: Lógica

- La Lógica es la parte de las Matemáticas y de la Filosofía que estudia qué razonamientos son válidos.
- Si echamos un vistazo al desarrollo histórico de la Informática, ésta siempre estuvo ligada a la Lógica, tanto en el software como en el hardware.

- **Enunciado** es toda frase declarativa con un significado completo de la que podemos afirmar que es verdadera o falsa.
- ¿Cuántos años tienes?"
- ¡No te fastidia la niña ésta!
- Esta frase es falsa
- Todos los pasajeros se han abrochado los cinturones

- Abandonamos el lenguaje natural para ir hacia un lenguaje formal, como los lenguajes de programación
- “Llueve o están regando" se formaliza en **Lógica de Proposiciones** mediante la fórmula $\boldsymbol{p} \vee \boldsymbol{q}$

- Los elementos y reglas necesarios para determinar un lenguaje formal, que forman parte de la **Sintaxis** de la Lógica, son:
  - Un **alfabeto** de símbolos primitivos.
  - Unas **reglas de formación** para combinar esos símbolos.
- Una vez finalizada la formalización se asocian a las fórmulas obtenidas valores de verdad que les dan significado.

Lógica Proposicional. Enunciados proposicionales

- **Enunciados simples o atómicos**: Son aquellos de los que no se pueden extraer otros enunciados, es decir, contienen un único enunciado.

-

“El tren llegó puntual"	“Estaba esperándole"
“El examen era difícil"	“Él había estudiado"
“Llegó tarde"	“Se encontró con su amigo"
“Hay examen"	“Le gusta lo que ve en clase"
"Juan estudia"	“La peli resultó tan buena como decían"“

Lógica Proposicional. Enunciados proposicionales

- **Enunciados compuestos o moleculares**: Son aquellos en los que aparece uno o varios enunciados unidos, explícita o implícitamente, por palabras especiales del lenguaje.

-

  “El tren llegó puntual y no estaba esperándole"

  “El examen era difícil o él no había estudiado"

  “No se encontró con su amigo porque llegó tarde"

  “La película no resultó tan buena como decían"

  “Juan estudia si hay examen o le gusta lo que se ve en clase"

Lógica. Formalización

- **variables proposicionales**: $p, q, r, \ldots$
- fórmulas proposicionales constituidas por variables proposicionales y **conectivos**:
  - $\neg$ es la **negación**.
  - $\wedge$ es la **conjunción** lógica.
  - $\vee$ es la **disyunción** lógica.
  - $\longrightarrow$ es la **implicación** o **condicional** lógica.
  - $\longleftrightarrow$ es la **doble implicación** o **bicondicional** lógica.

Fórmulas proposicionales básicas

- Aparece un único conectivo y una o dos variables proposicionales
-

NOMBRE	SÍMBOLO	FÓRMULA	ENUNCIADOS
NEGACIÓN	$\neg$	$\neg p$	no p.
CONJUNCIÓN	$\wedge$	$(p \wedge q)$	p y q, p pero q, p y sin embargo q.
DISYUNCIÓN	$\vee$	$(p \vee q)$	p o q, p o q o ambos.
IMPLICACIÓN O **CONDICIONAL**	$\longrightarrow$	$(p \longrightarrow q)$,	si p entonces q, p sólo si q, p es suficiente para q, q es necesario para p.
DOBLE IMPLICACIÓN O **BICONDICIONAL**	$\longleftrightarrow$	$(p \longleftrightarrow q)$	p si y sólo si q, si p entonces q y si q entonces p, p es necesario y suficiente para q.

Sintaxis en Lógica Proposicional

- Símbolos primitivos del lenguaje
  - **Variables proposicionales**: $\boldsymbol{\Sigma = \{p, q, r, \dots\}}$.
  - **Constantes lógicas**: $\{\top, \bot\}$, cuyo significado concreto se introducirá al estudiar la Semántica.
  - **Conectivos**: $\{\neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$.
  - **Símbolos auxiliares**: $\{(,)\}$.
- Las fórmulas de la Lógica Proposicional serán palabras del **alfabeto** $\boldsymbol{A_\Sigma} = \boldsymbol{\Sigma} \cup \{\top, \bot\} \cup \{\neg, \vee, \wedge, \rightarrow, \leftrightarrow\} \cup \{(,)\}$.

Reglas de formación

- Son **fórmulas proposicionales** sobre el alfabeto $\mathcal{A}_\Sigma$ aquellas palabras que podamos construir con los símbolos del alfabeto, en un número finito de pasos, aplicando las siguientes reglas:
  1. $\top, \bot$ y las variables proposicionales son fórmulas proposicionales, llamadas **fórmulas atómicas**.
  2. Si F es una fórmula proposicional, entonces $\neg F$ es también fórmula proposicional.
  3. Si F y G son fórmulas proposicionales, entonces $(F \oplus G)$ es también fórmula proposicional, es decir, $(F \vee G)$, $(F \wedge G)$, $(F \to G)$ y $(F \leftrightarrow G)$ son fórmulas proposicionales.
- El conjunto de todas las fórmulas proposicionales sobre el alfabeto $\mathcal{A}_\Sigma$ se representará por $\mathcal{L}_\Sigma$.

- Una fórmula se dice que es un **literal** si es una variable proposicional o bien una variable proposicional negada.
- Dada una fórmula $F \in L_{\Sigma}$ se dice que:
  - $\neg$ es el **conectivo principal** de F si $F = \neg A$ para alguna fórmula $A \in L_{\Sigma}$.
  - $\oplus$ es el **conectivo principal** de F si $F = (A \oplus B)$ para ciertas fórmulas A y B de L_{Σ}.

Convenio de precedencia

- Nivel 1: $\neg$
  - Nivel 2: $\wedge, \vee$
  - Nivel 3: $\rightarrow, \leftrightarrow$
- Los criterios para la **simplificación de paréntesis** son los siguientes:
  1. Los paréntesis exteriores de una fórmula se pueden omitir.
  2. En caso de que haya dos o más conectivos que puedan ser el conectivo principal se tomará el de mayor nivel.
  3. Si hay al menos dos conectivos de este nivel los paréntesis son imprescindibles para evitar la ambigüedad.

Principio de Recursión Estructural

- Sea $f : \Sigma \cup \{\top, \bot\} \rightarrow E$ una función, entonces f se extiende a $\tilde{f} : L_\Sigma \rightarrow E$ utilizando el siguiente esquema recursivo:

- 1. Si F es una fórmula atómica, entonces $\tilde{f}(F) = f(F)$.
  2. El valor de $\tilde{f}(\neg F)$ se define en función de $\tilde{f}(F)$, para $F \in L_\Sigma$.
  3. El valor de $\tilde{f}(F \oplus G)$ se define en función de $\tilde{f}(F)$ y $\tilde{f}(G)$, para $F, G \in L_\Sigma$.

- El Principio de Recursión Estructural asegura que para cada esquema recursivo la extensión $\tilde{f}$ es única.

- Se define el **árbol estructural** de una fórmula $F \in L_{\Sigma}$ mediante la función $A : L_{\Sigma} \longrightarrow \{\text{Árboles}\}$ definida recursivamente como sigue:
- 1. Si F es una fórmula atómica $A(F) = F$

  2. $$A(\neg F) = \begin{array}{c} \neg F \\ \downarrow \\ A(F) \end{array}$$

  3. $$A(F \oplus G) = \begin{array}{ccc} & F \oplus G & \\ \swarrow & & \searrow \\ A(F) & & A(G) \end{array}$$

- **subfórmulas**

Semántica en Lógica Proposicional

- Lo que vamos a estudiar es si una fórmula es verdadera o falsa
- y si un razonamiento es correcto o no
- Estudiamos si cada una de las partes de la fórmula es correcta.
- Una *valoración* es cualquier aplicación $V : \Sigma \rightarrow \{0, 1\}$

Valor veritativo de una fórmula

- Sea $V : \Sigma \longrightarrow \{0,1\}$ una valoración. Se define la función $\tilde{V} : L_\Sigma \longrightarrow \{0,1\}$, llamada **valor veritativo** para la valoración V,mediante recursión estructural:

1. $\tilde{V}(p) = V(p)$ si p es una var propos, $\begin{cases} \tilde{V}(\perp) = 0 \\ \tilde{V}(\top) = 1 \end{cases}$
2. $\tilde{V}(\neg F) = \begin{cases} 1 & \text{si } \tilde{V}(F) = 0 \\ 0 & \text{si } \tilde{V}(F) = 1 \end{cases}$
3. $\tilde{V}(F \wedge G) = \begin{cases} 1 & \text{si } \tilde{V}(F) = \tilde{V}(G) = 1 \\ 0 & \text{en caso contrario} \end{cases}$
4. $\tilde{V}(F \vee G) = \begin{cases} 0 & \text{si } \tilde{V}(F) = \tilde{V}(G) = 0 \\ 1 & \text{en caso contrario} \end{cases}$
5. $\tilde{V}(F \rightarrow G) = \begin{cases} 0 & \text{si } \tilde{V}(F) = 1,\ \tilde{V}(G) = 0 \\ 1 & \text{en caso contrario} \end{cases}$
6. $\tilde{V}(F \leftrightarrow G) = \begin{cases} 1 & \text{si } \tilde{V}(F) = \tilde{V}(G) \\ 0 & \text{en caso contrario} \end{cases}$

NO Valor veritativo de una fórmula

- Sea V una valoración que cumple: $V(p) = 0, \ V(q) = 1$ y $V(r) = 1$.
- Para calcular el valor veritativo de $\neg p \rightarrow p \vee q$ se procede como sigue:
  - $\tilde{V}(p) = 0$ implica que $\tilde{V}(\neg p) = 1$, por (2), y
  - $\tilde{V}(p) = 0$ y $\tilde{V}(q) = 1$ implican que $\tilde{V}(p \vee q) = 1$, por (4).
  - Finalmente $\tilde{V}(\neg p) = 1$ y $\tilde{V}(p \vee q) = 1$ implican que $\tilde{V}(\neg p \rightarrow p \vee q) = 1$, por (5).
- Para calcular el valor veritativo de $r \rightarrow p \wedge q$ se procede de manera análoga
- Para una fórmula F, con variables proposicionales $\{p_1, p_2, \ldots, p_n\}$, el número de “valoraciones significativas" es finito, concretamente 2^n.

Valores veritativos de las fórmulas básicas

- $$V(\neg p) = 1 - V(p)$$
- $$V(p \wedge q) = min\{V(p), V(q)\}$$
- $$V(p \vee q) = max\{V(p), V(q)\}$$
- $$V(p \longrightarrow q) = max\{1 - V(p), V(q)\}$$
- $$V(p \leftrightarrow q) = 1 - |V(p) - V(q)|$$
- Para una fórmula F, con variables proposicionales $\{\, p_1, p_2, \ldots, p_n \,\}$, el número de “valoraciones significativas" es finito, concretamente 2^n.

Modelo

- Si $V(F) = 1$ diremos que la valoración V es *modelo* de $F \in L_{\Sigma}$, .
- Si $V(F) = 1$, V *satisface* a F.
- Si $V(F) = 0$ diremos que la valoración V *no es modelo*, o es no modelo, de $F \in L_{\Sigma}$, .
- Si $V(F) = 0$, V *no satisface* a F.

tautología

- Se dice que $F \in L_\Sigma$ es una *tautología* si toda valoración V es modelo de F.
- Se dice que $F \in L_\Sigma$ es una *contradicción* si ninguna valoración V es modelo de F.
- Se dice que $F \in L_\Sigma$ es *satisfactible* si existe alguna valoración que sea modelo de F.

Equivalencia lógica

- Se dice que $F, G \in L_\Sigma$ son *lógicamente equivalentes*, y se denotará $F \equiv G$, sii $V(F) = V(G)$ para toda valoración.
- $F \equiv G$ si F y G tienen exactamente los mismos modelos.

$F \equiv G$ propiedades

- Sean $F,\ G,\ H \in L_\Sigma$. Se verifica:

1. $$F \equiv F$$

2. $$\text{Si } F \equiv G, \text{ entonces } G \equiv F$$

3. $$\text{Si } F \equiv G \text{ y } G \equiv H, \text{ entonces } F \equiv H$$

4. $$\neg\neg F \equiv F$$

5. $$F \equiv \top \text{ si y sólo si } \neg F \equiv \bot$$

$$\neg\top \equiv \bot \text{ y } \neg\bot \equiv \top$$

- Leyes conmutativas

  $\boldsymbol{F} \vee \boldsymbol{G} \equiv \boldsymbol{G} \vee \boldsymbol{F}$.

  $\boldsymbol{F} \wedge \boldsymbol{G} \equiv \boldsymbol{G} \wedge \boldsymbol{F}$.

- Leyes distributivas

  $(\boldsymbol{F} \vee \boldsymbol{G}) \wedge \boldsymbol{H} \equiv (\boldsymbol{F} \wedge \boldsymbol{H}) \vee (\boldsymbol{G} \wedge \boldsymbol{H})$.

  $(\boldsymbol{F} \wedge \boldsymbol{G}) \vee \boldsymbol{H} \equiv (\boldsymbol{F} \vee \boldsymbol{H}) \wedge (\boldsymbol{G} \vee \boldsymbol{H})$.

- Leyes de identidad

  $\boldsymbol{F} \wedge \top \equiv \boldsymbol{F}$.

  $\boldsymbol{F} \vee \bot \equiv \boldsymbol{F}$.

- Leyes del complementario

  $\boldsymbol{F} \vee \neg \boldsymbol{F} \equiv \top$.

  $\boldsymbol{F} \wedge \neg \boldsymbol{F} \equiv \bot$.

Equivalencias

- Leyes de idempotencia

  $F \vee F \equiv F$.

  $F \wedge F \equiv F$.

- Leyes de acotación

  $F \vee \top \equiv \top$.

  $F \wedge \bot \equiv \bot$.

- Leyes de absorción

  $F \vee (F \wedge G) \equiv F$.

  $F \wedge (F \vee G) \equiv F$.

- Leyes asociativas

  $(F \vee G) \vee H \equiv F \vee (G \vee H)$.

  $(F \wedge G) \wedge H \equiv F \wedge (G \wedge H)$.

Equivalencias

- Leyes de De Morgan

  $\neg(F \vee G) \equiv \neg F \wedge \neg G$.

  $\neg(F \wedge G) \equiv \neg F \vee \neg G$.

- Relación entre conectivos

  $F \rightarrow G \equiv \neg F \vee G$.

  $F \leftrightarrow G \equiv (F \rightarrow G) \wedge (G \rightarrow F)$.

Equivalencias

- $$F \to G \equiv \neg G \to \neg F$$
- $$\neg(F \to G) \equiv F \land \neg G$$
- $$F \leftrightarrow G \equiv (F \land G) \lor (\neg F \land \neg G)$$
- $$\neg(F \leftrightarrow G) \equiv (F \land \neg G) \lor (\neg F \land G)$$
- $$\bot \to F \equiv \top$$
- $$F \to \top \equiv \top$$

Conjuntos de fórmulas

- Sea $\Phi \subset L_{\Sigma}$ un conjunto de fórmulas proposicionales y V una valoración.
- V *satisface* a Φ, o V es *modelo* de Φ, si $V(F) = 1$ para cada $F \in \Phi$.
- Φ es un *conjunto satisfactible* si existe una valoración V que es modelo de Φ.
- En caso contrario se dice que Φ es *insatisfactible*.