TEMA 2 Definición de la lógica proposicional

1. ¿Qué es una lógica?

Consideraremos en esta asignatura que una lógica es la unión de:

1. **Sintaxis**: ¿De qué símbolos dispone el lenguaje y de qué maneras los puedo combinar para obtener una fórmula F?
2. **Semántica**:
    - Una definición de interpretación I: ¿Qué posibles significados existen para los símbolos?
    - Una definición del concepto de satisfacción: ¿Cuando una interpretación I satisface una fórmula F?

En una lógica también suele haber (pero no necesariamente) métodos de deducción, que describen cómo a partir de unas fórmulas dadas se pueden inferir otras nuevas, dando lugar a gran diversidad de aplicaciones prácticas de las lógicas. Aquí, para comenzar, veremos un ejemplo de lógica muy simple, la lógica proposicional, que, a pesar de su sencillez, permite expresar conceptos muy importantes en el mundo de la informática y tiene muchas aplicaciones.

2. Definición de la Lógica Proposicional

Sintaxis: En esta lógica, el vocabulario es un conjunto de símbolos proposicionales o de predicado P (cuyos elementos escribiremos normalmente como p, q,r, . . .). Una fórmula de lógica proposicional sobre P se define como:

- Todo símbolo de predicado de P es una fórmula.
- Si F y G son fórmulas, entonces $(F \land G)$ y $(F \lor G)$ son fórmulas.
- Si F es una fórmula, entonces $\neg F$ es una fórmula.
- Nada más es una fórmula.

Interpretación: Una interpretación I sobre el vocabulario P es una función $I: P \to \{0, 1\}$, es decir, I es una función que, para cada símbolo de predicado, nos dice si es 1 (cierto, true) o 0 (falso, false).

Satisfacción: Sean I una interpretación y F una fórmula, ambas sobre el vocabulario P. La evaluación en I de F, denotada $eval_I(F)$, es una función que por cada fórmula da un valor de $\{0, 1\}$. Definimos $eval_I(F)$ para todos los casos posibles de F, usando min, max y − que denotan el mínimo, el máximo, y la resta (sobre números del conjunto {0, 1}), donde, como sabemos:

min(0, 0) = min(0, 1) = min(1, 0) = 0 y min(1, 1) = 1

max(1, 1) = max(0, 1) = max(1, 0) = 1 y max(0, 0) = 0

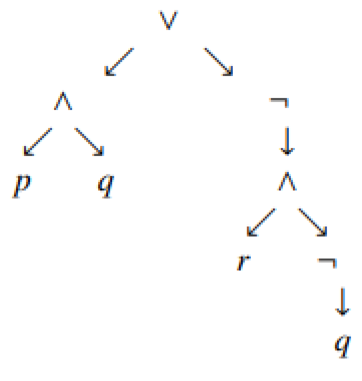
- si F es un símbolo p de P entonces $eval_I(F) = I(p)$
    - $eval_I((F \land G)) = min(eval_I(F), eval_I(G))$
    - $eval_I((F \lor G)) = max(eval_I(F), eval_I(G))$
    - $evalI(\neg F) = 1 - eval_I(F)$

Definimos: si $eval_I(F) = 1$ entonces I satisface F, denotado I |= F. En este caso también se dice que F es cierto en I, o que I es un **modelo** de F.

3. Explicaciones sobre la definición de la lógica proposicional

Otras conectivas: En lógica proposicional, a ∧, ∨ y ¬ se les llama **conectivas lógicas**. A menudo también se consideran otras conectivas, como → o ↔. Aquí no las hemos introducido porque son expresables mediante las tres conectivas dadas. Por ejemplo, las fórmulas F → G y F ↔ G pueden verse como abreviaturas de (¬F ∨ G) y de ((¬F ∨ G) ∧ (¬G ∨ F)) respectivamente.

No ambigüedad, representación en árbol: Las fórmulas, vistas como una cadena de símbolos (conectivas, símbolos de predicado y paréntesis), no son ambiguas: existe una única forma de entenderlas. Formalmente: para cada fórmula F existe un único árbol que representa la construcción de F según la definición de la sintaxis de la lógica proposicional. Por ejemplo, la fórmula ((p ∧ q) ∨ ¬(r ∧ ¬q)) solo puede leerse como un ∨ de la fórmula (p∧q) y de la fórmula ¬(r ∧ ¬q). Estas a su vez sólo son legibles de una única manera, y así sucesivamente. El árbol correspondiente es:



Prioridades de conectivas y paréntesis: A veces podemos omitir paréntesis innecesarios: quitarlos no introduce ambigüedad. Por ejemplo, podemos escribir p ∧ q en vez de (p ∧ q). Además se asumen las siguientes prioridades entre las conectivas; de más prioritario a menos, tenemos: ¬ ∧ ∨ → ↔. Así, podemos escribir p ∧ q ∨ r para referirnos a (p ∧

q) ∨ r, pero no podemos omitir paréntesis en ¬(p ∧ (q ∨ r)). Nótese la similitud con la suma, el producto, y el "−" unario en aritmética: con $-x * y + z$ nos referimos a $((-x) * y) + z$ y no a, por ejemplo, $-(x * (y + z))$, ni a $(-x) * (y + z)$.

¿Qué cosas se modelan con esta lógica?: la idea intuitiva de esta lógica es que sirve para modelar (y razonar formalmente sobre) situaciones de la vida real en las que tratemos con enunciados o proposiciones que solo pueden ser o bien ciertas o bien falsas.

Por ejemplo, si P es { llueve, hace sol, está nublado }, cada interpretación I para esta P modela una situación real del tiempo del estilo de "sí llueve, no hace sol y no está nublado", es decir, I(llueve) = 1, I(hace sol) = 0, I(está nublado) = 0. Con esta interpretación I, si F es la fórmula

llueve ∧ ¬(hace sol ∨ está nublado)

entonces, por la definición de $eval_I$, tenemos

$eval_I(F) = \min(I(llueve), 1 - \max(I(hace\ sol), I(esta\ nublado))) =$

$\min(1, 1 - \max(0, 0)) = 1$

luego F es cierta en I, es decir, tenemos I |= F. Similarmente, si G es la fórmula

¬llueve ∨ (hace sol ∨ está nublado),

entonces I no satisface G, es decir, I |≠ G.

Podemos listar todas las posibles interpretaciones y la evaluación en cada una de ellas de la fórmula *llueve ∧ ¬(hace sol ∨ está nublado)* mediante una tabla de verdad (escribiendo p, q, r en vez de llueve, hace sol, y está nublado):

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

y vemos que en este ejemplo solo hay una interpretación que satisface la fórmula, que es cuando llueve, no hace sol y no está nublado.

4. Satisfactibilidad, tautología, consecuencia y equivalencia

Ahora podemos introducir la siguiente nomenclatura básica. Estas nociones no dependen de la lógica concreta. Se definen de manera idéntica en otras lógicas distintas a la lógica proposicional, como la lógica de primer orden. Por eso no se repetirán cuando tratemos la lógica de primer orden.

1. Una fórmula F es **satisfactible** si tiene algun modelo, es decir, si existe alguna interpretación I tal que I |= F. Una fórmula F es insatisfacible (o es una contradicción) si no es satisfactible.
2. Una fórmula F es una **tautología** (o es válida), si toda interpretación es modelo de F, es decir, si para toda interpretación I tenemos I |= F.
3. Sean F y G fórmulas construidas sobre un vocabulario P. Definimos: F es **consecuencia lógica** de G si todo modelo de G también lo es de F, es decir, si para toda interpretación I sobre P tal que I |= G tenemos I |= F. Sobrecargando el operador "|=", esto se denotará por G |= F.
4. Sean F y G fórmulas construidas sobre un vocabulario P. Definimos: F y G son **lógicamente equivalentes** si tienen los mismos modelos, es decir, si para toda interpretación I sobre P tenemos I |= G si y sólo si I |= F. Esto se denotará por G ≡ F.

Nótese que el símbolo "**|=**" denota **satisfacción** cuando a su izquierda hay una interpretación I; en ese caso, I |= F denota que I satisface F. En cambio, cuando a la izquierda hay una fórmula G, el símbolo "**|=**" denota **consecuencia lógica**; así, G |= F denota que todo modelo de G satisface F.

Este tipo de propiedades (satisfactibilidad o tautología de una fórmula, consecuencia lógica o equivalencia entre fórmulas, etc.) se plantean en diversas aplicaciones prácticas de una lógica. Por ejemplo, en Intel pueden estar interesados en saber si dos circuitos (dos fórmulas) son equivalentes, o saber si un circuito F cumple cierta propiedad G (es decir, si F |= G).

La lógica proposicional es especialmente interesante porque todas estas propiedades son decidibles: para cada una de ellas hay algún programa de ordenador que siempre termina y nos da una respuesta correcta sí/no. Por ejemplo, para saber si una fórmula F dada es satisfactible, podemos construir su tabla de verdad con la lista de todas las posibles interpretaciones I para el vocabulario P sobre el que F está construido, y averiguar si hay algún 1 en la columna de F, esto es, si $eval_I(F) = 1$ para alguna I.

Similarmente, tenemos F |= G para dos fórmulas F y G dadas si y sólo si en la tabla de verdad (para el vocabulario P de F y G), por cada fila con un 1 en la columna de F también hay un 1 en la columna de G.

Sin embargo, estos métodos basados en la tabla de verdad no son los más eficientes. En los ejercicios veremos que cualquiera de estas propiedades se puede expresar en términos del problema de la satisfactibilidad (abreviado, SAT), esto es, el de determinar si una fórmula proposicional dada es satisfacible o no. Por ejemplo, para dos fórmulas F y G tenemos $F \models G$ si, y sólo si, $F \land \neg G$ es insatisfacible. Por eso, para las aplicaciones prácticas se usa un SAT solver, esto es, un programa de ordenador que, dada una fórmula F, decide si F es satisfacible o no.