Cálculo Proposicional

Argumentos de tipo informal sobre por qué un sistema funciona no capta debilidades cruciales. El sistema resulta frágil. Un enfoque alternativo para el diseño e implementación de sistemas se basa en usar un lenguaje formal para su especificación, y así poder razonar sobre sistemas de software.
Preguntas del tipo:

- ¿Este programa que acepta un arreglo de enteros produce un arreglo ordenado?
- ¿Este programa accede a una localidad de memoria no disponible?
- ¿Esta función siempre para?

son frecuentes. Para responderlas necesitamos de un "sistema de cálculo": lógica proposicional (PL) y luego lógica de primer orden (FOL).

El tipo de afirmaciones que se estudian en PL son del tipo:

PL : "el cielo es azul", "esta afirmación se refiere a sí misma": éstas se llaman *proposiciones*.

FOL : "x es azul", "y se refiere a z": éstas se llaman *predicados*.

En Lógica los conceptos más fundamentales son:

- **satisfactibilidad:** (¿esta fórmula es alguna vez cierta?)
- **validez:** (¿esta fórmula es siempre cierta?)

Sintaxis de PL

Sintaxis: se refiere a un conjunto de símbolos y reglas para combinarlos y formar "afirmaciones" ó fórmulas de un lenguajes (sistema formal).
Los símbolos que se usan en la Lógica Proposicional son:

1. $\top$ (top,cierto), $\bot$ (bottom, falso)
2. Letras proposicionales: $P, Q, R, \ldots, P_1, P_2, \ldots$. En general, las letras proposicionales son las primeras letras mayúsculas del alfabeto con o sin subíndices escritas en itálica.
3. Conectivos lógicos o booleanos: si $\mathcal{F}, \mathcal{F}_1, \mathcal{F}_2$ son fórmulas, entonces
   - $(\neg\mathcal{F})$: negación (no)
   - $(\mathcal{F}_1 \wedge \mathcal{F}_2)$: conjunción (y)
   - $(\mathcal{F}_1 \vee \mathcal{F}_2)$: disyunción (o)
   - $(\mathcal{F}_1 \rightarrow \mathcal{F}_2)$: implicación (implica), $\mathcal{F}_1$ antecedente y $\mathcal{F}_2$ consecuente.
   - $(\mathcal{F}_1 \leftrightarrow \mathcal{F}_2)$: ssi (si y sólo si)

Nótese que $\neg$ es un conectivo de aridad uno; los demás conectivos tienen aridad dos.

Definición

1. *Los* **átomos** *son:* $\top, \bot$ *y las letras proposicionales* $P, Q, \ldots$
2. *Una* **literal** *es un átomo o su negación.*
3. *Una* **fórmula** *es una literal o la aplicación de un conectivo lógico a una o varias fórmulas.*

Definición

Una fórmula $\mathcal{G}$ es una **subfórmula** *de una fórmula $\mathcal{F}$ si aparece sintácticamente en $\mathcal{F}$. Más precisamente:*

1. *La única subfórmula de un átomo es el propio átomo.*
2. *Las subfórmulas de $(\neg\mathcal{F})$ son $(\neg\mathcal{F})$ y $\mathcal{F}$.*
3. *Las subfórmulas de $(\mathcal{F}_1 \wedge \mathcal{F}_2)$, $(\mathcal{F}_1 \vee \mathcal{F}_2)$, $(\mathcal{F}_1 \to \mathcal{F}_2)$, $(\mathcal{F}_1 \leftrightarrow F_2)$ son las fórmulas mismas y $\mathcal{F}_1$, $\mathcal{F}_2$.*

Una **subfórmula estricta** es una fórmula que no es la fórmula misma.

Ejemplo

Sea la fórmula

$$\mathcal{F} : ((P \wedge Q) \rightarrow (P \vee \neg Q))$$

entonces tenemos:

- letras proposicionales: P, Q.
- átomos: P, Q
- literales: $P, Q, (\neg Q)$ ($(\neg Q)$ no es un átomo).
- subfórmulas: $\mathcal{F}, (P \wedge Q), (P \vee (\neg Q)), (\neg Q), P, Q$.

Los paréntesis son a veces innecesarios. Para esto se define la **precedencia** de los conectivos: de alto a bajo,

$$\neg, \wedge, \vee, \rightarrow, \leftrightarrow .$$

En caso de empate se asocia a la derecha (precaución: esta convención cambia de texto en texto).

Ejemplo

Sea

$$\mathcal{F} : P \wedge Q \rightarrow P \vee Q.$$

Tal $\mathcal{F}$ es la abreviatura de la fórmula que se obtiene al restaurar paréntesis:

1. $(P \wedge Q) \rightarrow P \vee Q$
2. $(P \wedge Q) \rightarrow (P \vee Q)$
3. $((P \wedge Q) \rightarrow (P \vee Q))$

Es decir $\mathcal{F}$ abrevia a

$$\mathcal{F}' : ((P \wedge Q) \rightarrow (P \vee Q))$$

Ejemplo

Restaurar los paréntesis de

$$P_1 \wedge \neg P_2 \wedge \top \vee \neg P_1 \wedge P_2.$$

Sol.

1. $P_1 \wedge \neg P_2 \wedge \top \vee \neg P_1 \wedge P_2$
2. $P_1 \wedge \neg P_2 \wedge \top \vee (\neg P_1) \wedge P_2$
3. $P_1 \wedge \neg P_2 \wedge \top \vee (\neg P_1) \wedge P_2$
4. $P_1 \wedge (\neg P_2) \wedge \top \vee (\neg P_1) \wedge P_2$
5. $P_1 \wedge (\neg P_2) \wedge \top \vee (\neg P_1) \wedge P_2$
6. $P_1 \wedge (\neg P_2) \wedge \top \vee ((\neg P_1) \wedge P_2)$
7. $P_1 \wedge (\neg P_2) \wedge \top \vee ((\neg P_1) \wedge P_2)$
8. $P_1 \wedge ((\neg P_2) \wedge \top) \vee ((\neg P_1) \wedge P_2)$
9. $P_1 \wedge ((\neg P_2) \wedge \top) \vee ((\neg P_1) \wedge P_2)$
10. $(P_1 \wedge ((\neg P_2) \wedge \top)) \vee ((\neg P_1) \wedge P_2)$
11. $(P_1 \wedge ((\neg P_2) \wedge \top)) \vee ((\neg P_1) \wedge P_2)$

12. $((P_1 \wedge ((\neg P_2) \wedge \top)) \vee ((\neg P_1) \wedge P_2))$

Ejemplo

Restaurar paréntesis:

$$A \leftrightarrow \neg B \vee C \rightarrow A$$

Sol.

1. $A \leftrightarrow \neg B \vee C \rightarrow A$
2. $A \leftrightarrow (\neg B) \vee C \rightarrow A$
3. $A \leftrightarrow (\neg B) \vee C \rightarrow A$
4. $A \leftrightarrow ((\neg B) \vee C) \rightarrow A$
5. $A \leftrightarrow ((\neg B) \vee C) \rightarrow A$
6. $A \leftrightarrow (((\neg B) \vee C) \rightarrow A)$
7. $A \leftrightarrow (((\neg B) \vee C) \rightarrow A)$
8. $(A \leftrightarrow (((\neg B) \vee C) \rightarrow A))$
9. $(A \leftrightarrow (((\neg B) \vee C) \rightarrow A))$