Equivalencia, validez y satisfacibilidad

La **equivalencia lógica** se define como sigue: dos sentencias α y β son equivalentes lógicamente si tienen los mismos valores de verdad en el mismo conjunto de modelos. Esto se denota como

$$\alpha \Leftrightarrow \beta$$

Por ejemplo ¬(P^Q) es equivalente a ¬Q v ¬P, lo podemos comprovar rápidamente con una tabla de verdad

P	Q	¬	P^Q	¬Q	V	¬ P
V	V	**F**	V	F	**F**	F
V	F	**V**	F	V	**V**	F
F	V	**V**	F	F	**V**	V
F	F	**V**	F	V	**V**	V

El segundo concepto es la **validez**. Una sentencia es valida si es verdadera en todos los modelos. Por ejemplo:

p	q	$p \wedge q$	$(p \wedge q) \rightarrow p$
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	V

Las sentencias validas tambiéns e conocen como tautologías, son necesariamente verdaderas por lo tanto vaciás de significado. ¿Qué utilidad tienen= De la definición de implicación podemos derivar el teorema de la deducción

Para cualquier sentencia α y β, $\alpha \models \beta$ si y sólo si la sentencia $(\alpha \Rightarrow \beta)$ es válida.

El último concepto es el de la satisfacibilidad. Una sentencia es satisfactoria si es verdadera para algún modelo. Por ejemplo:

1	2	3	4	5
A	B	C	$B \vee C$	$A \wedge (B \vee C)$
V	V	V	V	V
V	V	F	V	V
V	F	V	V	V
V	F	F	F	F
F	V	V	V	F
F	V	F	V	F
F	F	V	V	F
F	F	F	F	F

La determinación de la satisfacibilidad d sentencias en lógica proposicional fue el primer problema que se demostró que era NP-completo. Muchos problemas en ciencias de la computación son en realidad problemas de satisfacibilidad. Por ejemplo, todos los problemas de satisfacción de restricciones lo son. Con algunas transformaciones adecuadas, los problemas de búsqueda también se pueden resolver mediante satisfacibilidad. La validez y la satisfacibilidad están íntimamente relacionadas α es válida si y sólo si ¬α es insatisfacible y viceversa.

Forma normal conjuntiva

Toda sentencia en lógica proposicional es equivalente lógicamente a una conjunción de disyunciones de literales. Una sentencia representada mediante una conjunción de disyunciones de literales se dice que esta en forma normal conjuntiva o FNC. Por ejemplo:

$$\neg A \wedge (B \vee C)$$
$$(A \vee B) \wedge (\neg B \vee C \vee \neg D) \wedge (D \vee \neg E)$$
$$A \wedge B.$$

La transformación de una sentencia a FNC se basa en reglas sobre equivalencias lógicas: la doble negación, las leyes de De Morgan, y la distributividad.

- **Algoritmo de cálculo de forma normal conjuntiva:**
  - **Algoritmo: Aplicando a una fórmula F los siguientes pasos se obtiene una forma normal conjuntiva de F:**

1. **Eliminar los bicondicionales usando la equivalencia**
$$A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A) \quad (1)$$
2. **Eliminar los condicionales usando la equivalencia**
$$A \rightarrow B \equiv \neg A \vee B \quad (2)$$
3. **Interiorizar las negaciones usando las equivalencias**
$$\neg(A \wedge B) \equiv \neg A \vee \neg B \quad (3)$$
$$\neg(A \vee B) \equiv \neg A \wedge \neg B \quad (4)$$
$$\neg\neg A \equiv A \quad (5)$$
4. **Interiorizar las disyunciones usando las equivalencias**
$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C) \quad (6)$$
$$(A \wedge B) \vee C \equiv (A \vee C) \wedge (B \vee C) \quad (7)$$

- **Ejemplo de cálculo de una FNC de $(p \rightarrow q) \vee (q \rightarrow p)$:**

$$\begin{aligned} & (p \rightarrow q) \vee (q \rightarrow p) \\ \equiv\ & (\neg p \vee q) \vee (\neg q \vee p) \ \ [\text{por } (2)] \\ \equiv\ & \neg p \vee q \vee \neg q \vee p \end{aligned}$$

- **Ejemplo de cálculo de una FNC de $\neg(p \wedge (q \rightarrow r))$:**

$$
\begin{array}{lll}
 & \neg(p \wedge (q \rightarrow r)) & \\
\equiv & \neg(p \wedge (\neg q \vee r)) & [\text{por } (2)] \\
\equiv & \neg p \vee \neg(\neg q \vee r) & [\text{por } (3)] \\
\equiv & \neg p \vee (\neg\neg q \wedge \neg r) & [\text{por } (4)] \\
\equiv & \neg p \vee (q \wedge \neg r) & [\text{por } (5)] \\
\equiv & (\neg p \vee q) \wedge (\neg p \vee \neg r) & [\text{por } (6)]
\end{array}
$$

- **Ejemplo de cálculo de una FNC de $(p \leftrightarrow q) \rightarrow r$:**

$$
\begin{array}{lll}
 & (p \leftrightarrow q) \rightarrow r & \\
\equiv & (p \rightarrow q) \wedge (q \rightarrow p) \rightarrow r & [\text{por } (1)] \\
\equiv & \neg((\neg p \vee q) \wedge (\neg q \vee p)) \vee r & [\text{por } (2)] \\
\equiv & (\neg(\neg p \vee q) \vee \neg(\neg q \vee p)) \vee r & [\text{por } (3)] \\
\equiv & ((\neg\neg p \wedge \neg q) \vee (\neg\neg q \wedge \neg p)) \vee r & [\text{por } (4)] \\
\equiv & ((p \wedge \neg q) \vee (q \wedge \neg p)) \vee r & [\text{por } (5)] \\
\equiv & (((p \wedge \neg q) \vee q) \wedge ((p \wedge \neg q) \vee \neg p)) \vee r & [\text{por } (6)] \\
\equiv & (((p \vee q) \wedge (\neg q \vee q)) \wedge ((p \vee \neg p) \wedge (\neg q \vee \neg p))) \vee r & [\text{por } (7)] \\
\equiv & (((p \vee q) \wedge (\neg q \vee q)) \vee r) \wedge (((p \vee \neg p) \wedge (\neg q \vee \neg p)) \vee r) & [\text{por } (7)] \\
\equiv & (((p \vee q) \vee r) \wedge ((\neg q \vee q) \vee r)) \wedge (((p \vee \neg p) \vee r) \wedge ((\neg q \vee \neg p) \vee r)) & [\text{por } (7)] \\
\equiv & (p \vee q \vee r) \wedge (\neg q \vee q \vee r) \wedge (p \vee \neg p \vee r) \wedge (\neg q \vee \neg p \vee r) & \\
\equiv & (p \vee q \vee r) \wedge (\neg q \vee \neg p \vee r) &
\end{array}
$$

Ejemplo 8: Para transformar la fórmula $\neg(p \rightarrow q) \leftrightarrow p \vee r$ a Forma Normal Conjuntiva, se pueden emplear los siguientes pasos:

$$
\begin{array}{ll}
 & \neg(p \rightarrow q) \leftrightarrow p \vee r \\
\equiv & \quad \{ \text{Eliminación} \leftrightarrow \} \\
 & (\neg (p \rightarrow q) \rightarrow p \vee r) \wedge ((p \vee r) \rightarrow \neg(p \rightarrow q)) \\
\equiv & \quad \{ \text{Eliminación} \rightarrow \} \\
 & (\neg (\neg (\neg p \vee q) \vee p \vee r) \wedge (\neg (p \vee r) \vee \neg(\neg p \vee q)) \\
\equiv & \quad \{ \text{Eliminación doble negación} \} \\
 & (\neg p \vee q \vee p \vee r) \wedge (\neg (p \vee r) \vee \neg(\neg p \vee q)) \\
\equiv & \quad \{ \text{Eliminación disyunción con literal y su opuesto} \} \\
 & (\neg (p \vee r) \vee \neg(\neg p \vee q)) \\
\equiv & \quad \{ \text{De Morgan} \} \\
 & (\neg p \wedge \neg r) \vee (\neg\neg p \wedge \neg q) \\
\equiv & \quad \{ \text{Eliminación doble negación} \} \\
 & (\neg p \wedge \neg r) \vee (p \wedge \neg q) \\
\equiv & \quad \{ \text{Distributiva} \vee \} \\
 & ((\neg p \wedge \neg r) \vee p) \wedge ((\neg p \wedge \neg r) \vee \neg q) \\
\equiv & \quad \{ \text{Distributiva} \vee \} \\
 & ((\neg p \vee p) \vee \neg r) \vee p) \wedge ((\neg p \wedge \neg r) \vee \neg q)
\end{array}
$$

Algoritmo de Resolución Proposicional

El algoritmo se basa en una regla de inferencia sencilla y, a la vez, de gran potencia: la regla de resolución. Puesto que se utiliza una sola regla, el algoritmo es fácil de analizar e implementar.

La idea del principio de resolución es simple: Si se sabe que se cumple: "P ó Q" y también se sabe que se cumple "no P ó R" entonces se puede deducir que se cumplirá "Q ó R". Ejemplo:

Si se tiene: "Gana o Pierde o Empata" y "Si Gana entonces da una Fiesta o Va de Viaje". Se puede deducir que: "O Pierde o Empata o da una Fiesta o va de Viaje".

Formalizando, la primera frase sería:

$$G \vee P \vee E \text{ y la segunda: } G \rightarrow F \vee V \equiv \neg G \vee F \vee V$$

La regla de resolución inferirá:

$$P \vee E \vee F \vee V$$