EJEMPLO 5 Justifica que las proposiciones $\neg(p \vee (\neg p \wedge q))$ y $\neg p \wedge \neg q$ son lógicamente equivalentes.

Solución: Podríamos utilizar una tabla de verdad para mostrar que estas fórmulas son equivalentes. En vez de ello, estableceremos la equivalencia desarrollando una serie de equivalencias lógicas intermedias usando una de las equivalencias de la Tabla 5 cada vez, comenzando con $\neg(p \vee (\neg p \wedge q))$ y finalizando con $\neg p \wedge \neg q$. Tenemos las siguientes equivalencias:

$\neg(p \vee (\neg p \wedge q))$	$\equiv \neg p \wedge \neg(\neg p \wedge q)$	por la segunda ley de De Morgan
	$\equiv \neg p \wedge [\neg(\neg p) \vee \neg q)]$	por la primera ley de De Morgan
	$\equiv \neg p \wedge (p \vee \neg q)$	por la ley de la doble negación
	$\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q)$	por la segunda ley distributiva
	$\equiv \mathbf{F} \vee (\neg p \wedge \neg q)$	puesto que $\neg p \wedge p \equiv \mathbf{F}$
	$\equiv (\neg p \wedge \neg q) \vee \mathbf{F}$	por la ley conmutativa para la disyunción
	$\equiv \neg p \wedge \neg q$	por la ley de identidad para $\mathbf{F}$

Consecuentemente, $\neg(p \vee (\neg p \wedge q))$ y $\neg p \wedge \neg q$ son lógicamente equivalentes. ◀

EJEMPLO 6 Muestra que $(p \wedge q) \rightarrow (p \vee q)$ es una tautología.

Solución: Para mostrar que esta sentencia es una tautología, usaremos equivalencias lógicas para demostrar que es lógicamente equivalente a $\mathbf{V}$. (*Nota:* Se podría haber hecho también mediante una tabla de verdad).

$(p \wedge q) \rightarrow (p \vee q)$	$\equiv \neg(p \wedge q) \vee (p \vee q)$	por el Ejemplo 3
	$\equiv (\neg p \vee \neg q) \vee (p \vee q)$	por la primera ley de De Morgan
	$\equiv (\neg p \vee p) \vee (\neg q \vee q)$	por las leyes asociativa y conmutativa para la disyunción
	$\equiv \mathbf{V} \vee \mathbf{V}$	por el Ejemplo 1 y la ley conmutativa para la disyunción
	$\equiv \mathbf{V}$	por la ley de dominación ◀

Se puede usar una tabla de verdad para determinar si una fórmula es una tautología. Esta tabla se puede construir a mano para una proposición con un número reducido de variables, pero cuando el número de variables crece, el método manual se vuelve impracticable. Por ejemplo, hay más de $2^{20} = 1\,048\,576$ filas en la tabla de verdad de una proposición de veinte variables. Claramente, se necesita la ayuda de un ordenador si queremos determinar de esta forma si la fórmula de 20 variables es una tautología. Pero cuando hay mil variables, ¿puede un ordenador determinar en un plazo de tiempo razonable si una fórmula es una tautología? Revisar cada una de las 2^{1000} (un número con más de trescientas cifras) combinaciones de valores de verdad no puede hacerse en un ordenador ni en billones de años. Además, no se conoce ningún otro procedimiento que pueda seguir un ordenador para determinar en un plazo razonable de tiempo si una fórmula con un número tan grande de variables es una tautología. Contestaremos a preguntas como éstas en el Capítulo 2, cuando estudiemos la complejidad de algoritmos.

Enlaces

ADA AUGUSTA, CONDESA DE LOVELACE (1815-1852) Ada Augusta fue la única hija del matrimonio del famoso poeta Lord Byron y Annabella Millbanke, los cuales se separaron cuando Ada tenía un mes. La crió su madre, que potenció su talento intelectual. Fue educada por los matemáticos William Frend y Augustus de Morgan. En 1838 se casó con Lord King, más tarde nombrado conde de Lovelace.

Ada Augusta continuó sus estudios de matemáticas tras su matrimonio, ayudando a Charles Babbage en su trabajo sobre una de las primeras máquinas calculadoras, llamada la máquina analítica. En sus escritos se encuentra la más completa descripción de esta máquina. Tras 1845, ella y Babbage trabajaron juntos en el desarrollo de un sistema para predecir carreras de caballos. Lamentablemente, su sistema no funcionó bien, dejando a Ada importantes deudas hasta su muerte. El lenguaje de programación Ada se nombró así en honor a la condesa de Lovelace.