

ESPECIFICACIONES DE SISTEMA

Traducir oraciones del lenguaje natural, como el español, a expresiones lógicas es una parte esencial de la especificación tanto de sistemas *hardware* como *software*. Los ingenieros de *software* y de sistemas reciben los requerimientos en lenguaje natural y producen especificaciones precisas y sin ambigüedades que pueden usarse como base para desarrollo de sistemas. El Ejemplo 11 muestra cómo se pueden utilizar las variables proposicionales en este proceso.

EJEMPLO 11 Expresa la especificación «La respuesta automatizada no se puede enviar cuando el sistema de archivos está lleno»

Ejemplos
adicionales

Solución: Una forma posible de traducir esto es denotar como p a «La respuesta automatizada se puede enviar» y como q a «El sistema de archivos está lleno». Entonces, $\neg p$ representa a «No se cumple que la respuesta automatizada se pueda enviar», lo que se puede expresar como «La respuesta automatizada no se puede enviar». Consecuentemente, nuestra especificación se puede representar mediante la implicación $q \rightarrow \neg p$. ◀

Las especificaciones de sistema no deberían contener requerimientos que puedan entrar en conflicto. Si así fuese, no habría forma de desarrollar un sistema que cumpliera todas las especificaciones. Consecuentemente, las expresiones proposicionales que representan esas especificaciones necesitan ser **consistentes**. Esto es, debe haber una asignación de valores de verdad a las variables de las expresiones que haga a todas las expresiones verdaderas.

EJEMPLO 12 Determina si estas especificaciones de sistemas son consistentes:

- «El mensaje de diagnóstico se almacena en un *buffer* o se vuelve a transmitir».
- «El mensaje de diagnóstico no se almacena en el *buffer*».
- «Si el mensaje de diagnóstico se almacena en el *buffer*, entonces se vuelve a transmitir».

Solución: Para determinar si estas expresiones son consistentes, primero las expresamos usando variables proposicionales. Denotemos a «El mensaje de diagnóstico se almacena en un *buffer*» como p y «El mensaje se vuelve a transmitir» como q . Las especificaciones se pueden escribir entonces como $p \vee q$, $\neg p$ y $p \rightarrow q$. Una asignación de valores de verdad que haga a las tres especificaciones verdaderas debe hacer p falsa para hacer $\neg p$ verdadera. Como queremos que $p \vee q$ sea verdadera, pero p debe ser falsa, q debe ser verdadera. Como $p \rightarrow q$ es verdadera cuando p es falsa y q verdadera, concluimos que estas especificaciones son consistentes, ya que las tres son verdaderas cuando p es falsa y q verdadera. Podríamos haber llegado a la misma conclusión usando una tabla de verdad para examinar las cuatro posibles asignaciones de valores de verdad a p y q . ◀

EJEMPLO 13 ¿Siguen siendo consistentes las especificaciones de sistema del Ejemplo 12 si se añade la especificación «El mensaje de diagnóstico no se vuelve a transmitir»?

Solución: Por los razonamientos del Ejemplo 12, las tres especificaciones de ese ejemplo son verdaderas sólo en el caso de que p sea falsa y q verdadera. Sin embargo, esta nueva especificación es $\neg q$, que es falsa cuando q es verdadera. Consecuentemente, estas cuatro especificaciones son inconsistentes. ◀

BÚSQUEDAS BOOLEANAS

Enlaces

Los conectivos lógicos tienen un amplio campo de aplicación en las búsquedas en grandes colecciones de información como, por ejemplo, los índices de páginas web. Como estas búsquedas emplean técnicas de lógica proposicional, se denominan **búsquedas booleanas**.

En las búsquedas booleanas se usa la conexión *AND* para emparejar datos almacenados que contengan los dos términos de búsqueda, la conexión *OR* se usa para emparejar uno o ambos términos de la búsqueda y la conexión *NOT* (a veces escrita *AND NOT*) se usa para excluir un término