es cuando toma su baño. Así, la *BC* no sería verdadera en el mundo real, sin embargo, mediante procedimientos de aprendizaje buenos no hace falta ser tan pesimistas.

7.4 Lógica proposicional: una lógica muy sencilla

LÓGICA PROPOSICIONAL

Ahora vamos a presentar una lógica muy sencilla llamada **lógica proposicional**[8]. Vamos a cubrir tanto la sintaxis como la semántica (la manera como se define el valor de verdad de las sentencias) de la lógica proposicional. Luego trataremos la **implicación** (la relación entre una sentencia y la que se sigue de ésta) y veremos cómo todo ello nos lleva a un algoritmo de inferencia lógica muy sencillo. Todo ello tratado, por supuesto, en el mundo de *wumpus*.

Sintaxis

SENTENCIAS ATÓMICAS

SÍMBOLO PROPOSICIONAL

La **sintaxis** de la lógica proposicional nos define las sentencias que se pueden construir. Las **sentencias atómicas** (es decir, los elementos sintácticos indivisibles) se componen de un único **símbolo proposicional**. Cada uno de estos símbolos representa una proposición que puede ser verdadera o falsa. Utilizaremos letras mayúsculas para estos símbolos: P, Q, R, y siguientes. Los nombres de los símbolos suelen ser arbitrarios pero a menudo se escogen de manera que tengan algún sentido mnemotécnico para el lector. Por ejemplo, podríamos utilizar $W_{1,3}$ para representar que el *wumpus* se encuentra en la casilla [1, 3]. (Recuerde que los símbolos como $W_{1,3}$ son *atómicos*, esto es, W, 1, y 3 no son partes significantes del símbolo.) Hay dos símbolos proposicionales con significado fijado: *Verdadero*, que es la proposición que siempre es verdadera; y *Falso*, que es la proposición que siempre es falsa.

SENTENCIAS COMPLEJAS

CONECTIVAS LÓGICAS

Las **sentencias complejas** se construyen a partir de sentencias más simples mediante el uso de las **conectivas lógicas**, que son las siguientes cinco:

NEGACIÓN

LITERAL

$\neg$ (no). Una sentencia como $\neg W_{1,3}$ se denomina **negación** de $W_{1,3}$. Un **literal** puede ser una sentencia atómica (un **literal positivo**) o una sentencia atómica negada (un **literal negativo**).

CONJUNCIÓN

$\wedge$ (y). Una sentencia que tenga como conectiva principal $\wedge$, como es $W_{1,3} \wedge H_{3,1}$, se denomina **conjunción**; sus componentes son los **conjuntores**.

DISYUNCIÓN

$\vee$ (o). Una sentencia que utiliza la conectiva $\vee$, como es $(W_{1,3} \wedge H_{3,1}) \vee W_{2,2}$, es una **disyunción** de los **disyuntores** $(W_{1,3} \wedge H_{3,1})$ y $W_{2,2}$. (Históricamente, la conectiva $\vee$ proviene de «vel» en Latín, que significa «o». Para mucha gente, es más fácil recordarla como la conjunción al revés.)

IMPLICACIÓN

PREMISA

CONCLUSIÓN

$\Rightarrow$ (implica). Una sentencia como $(W_{1,3} \wedge H_{3,1}) \Rightarrow \neg W_{2,2}$ se denomina **implicación** (o condicional). Su **premisa** o **antecedente** es $(W_{1,3} \wedge H_{3,1})$, y su **conclusión** o **consecuente** es $\neg W_{2,2}$. Las implicaciones también se conocen como **reglas** o afir-

[8] A la lógica proposicional también se le denomina **Lógica Booleana**, por el matemático George Boole (1815-1864).

maciones **si-entonces**. Algunas veces, en otros libros, el símbolo de la implicación se representa mediante $\supset$ o $\rightarrow$.

BICONDICIONAL

$\Leftrightarrow$ (si y sólo si). La sentencia $W_{1,3} \Leftrightarrow \neg W_{2,2}$ es una **bicondicional.**

En la Figura 7.7 se muestra una gramática formal de la lógica proposicional; mira la página 984 si no estás familiarizado con la notación BNF.

Sentencia	$\rightarrow$	*Sentencia Atómica* | *Sentencia Compleja*
Sentencia Atómica	$\rightarrow$	**Verdadero** | **Falso** | *Símbolo Proposicional*
Símbolo Proposicional	$\rightarrow$	**P** | **Q** | **R** | ...
Sentencia Compleja	$\rightarrow$	$\neg$ *Sentencia*
		| (*Sentencia* $\wedge$ *Sentencia*)
		| (*Sentencia* $\vee$ *Sentencia*)
		| (*Sentencia* $\Rightarrow$ *Sentencia*)
		| (*Sentencia* $\Leftrightarrow$ *Sentencia*)

Figura 7.7 Una gramática BNF (Backus-Naur Form) de sentencias en lógica proposicional.

Fíjese en que la gramática es muy estricta respecto al uso de los paréntesis: cada sentencia construida a partir de conectivas binarias debe estar encerrada en paréntesis. Esto asegura que la gramática no sea ambigua. También significa que tenemos que escribir, por ejemplo, $((A \wedge B) \Rightarrow C)$ en vez de $A \wedge B \Rightarrow C$. Para mejorar la legibilidad, a menudo omitiremos paréntesis, apoyándonos en su lugar en un orden de precedencia de las conectivas. Es una precedencia similar a la utilizada en la aritmética (por ejemplo, $ab + c$ se lee $((ab) + c)$ porque la multiplicación tiene mayor precedencia que la suma). El orden de precedencia en la lógica proposicional (de mayor a menor) es: $\neg$, $\wedge$, $\vee$, $\Rightarrow$ y $\Leftrightarrow$. Así, la sentencia

$$\neg P \vee Q \wedge R \Rightarrow S$$

es equivalente a la sentencia

$$((\neg P) \vee (Q \wedge R)) \Rightarrow S$$

La precedencia entre las conectivas no resuelve la ambigüedad en sentencias como $A \wedge B \wedge C$, que se podría leer como $((A \wedge B) \wedge C)$ o como $(A \wedge (B \wedge C))$. Como estas dos lecturas significan lo mismo según la semántica que mostraremos en la siguiente sección, se permiten este tipo de sentencias. También se permiten sentencias como $A \vee B \vee C$ o $A \Leftrightarrow B \Leftrightarrow C$. Sin embargo, las sentencias como $A \Rightarrow B \Rightarrow C$ no se permiten, ya que su lectura en una dirección y su opuesta tienen significados muy diferentes; en este caso insistimos en la utilización de los paréntesis. Por último, a veces utilizaremos corchetes, en vez de paréntesis, para conseguir una lectura de la sentencia más clara.

Semántica

Una vez especificada la sintaxis de la lógica proposicional, vamos a definir su semántica. La semántica define las reglas para determinar el valor de verdad de una sentencia respecto a un modelo en concreto. En la lógica proposicional un modelo define el valor de verdad (*verdadero* o *falso*). Por ejemplo, si las sentencias de la base de conocimiento utilizan los símbolos proposicionales $H_{1,2}$, $H_{2,2}$, y $H_{3,1}$, entonces un modelo posible sería

$$m_1 = \{H_{1,2} = falso, H_{2,2} = \text{falso}, H_{3,1} = verdadero\}$$

Con tres símbolos proposicionales hay $2^3 = 8$ modelos posibles, exactamente los que aparecen en la Figura 7.5. Sin embargo, fíjese en que gracias a que hemos concretado la sintaxis, los modelos se convierten en objetos puramente matemáticos sin tener necesariamente una conexión al mundo de *wumpus*. $H_{1,2}$ es sólo un símbolo, podría denotar tanto «hay un hoyo en la casilla [1, 2]», como «estaré en París hoy y mañana».

La semántica en lógica proposicional debe especificar cómo obtener el valor de verdad de *cualquier* sentencia, dado un modelo. Este proceso se realiza de forma recursiva. Todas las sentencias se construyen a partir de las sentencias atómicas y las cinco conectivas lógicas; entonces necesitamos establecer cómo definir el valor de verdad de las sentencias atómicas y cómo calcular el valor de verdad de las sentencias construidas con las cinco conectivas lógicas. Para las sentencias atómicas es sencillo:

- *Verdadero* es verdadero en todos los modelos y *Falso* es falso en todos los modelos.
- El valor de verdad de cada símbolo proposicional se debe especificar directamente para cada modelo. Por ejemplo, en el modelo anterior m_1, $H_{1,2}$ es falso.

Para las sentencias complejas, tenemos reglas como la siguiente

- Para toda sentencia s y todo modelo m, la sentencia $\neg s$ es verdadera en m si y sólo si s es falsa en m.

TABLA DE VERDAD

Este tipo de reglas reducen el cálculo del valor de verdad de una sentencia compleja al valor de verdad de las sentencias más simples. Las reglas para las conectivas se pueden resumir en una **tabla de verdad** que especifica el valor de verdad de cada sentencia compleja según la posible asignación de valores de verdad realizada a sus componentes. En la Figura 7.8 se muestra la tabla de verdad de las cinco conectivas lógicas. Utilizando estas tablas de verdad, se puede obtener el valor de verdad de cualquier sentencia s según un modelo m mediante un proceso de evaluación recursiva muy sencillo. Por ejemplo, la sentencia $\neg H_{1,2} \wedge (H_{2,2} \vee H_{3,1})$ evaluada según m_1, da *verda-*

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
falso	*falso*	*verdadero*	*falso*	*falso*	*verdadero*	*verdadero*
falso	*verdadero*	*verdadero*	*falso*	*verdadero*	*verdadero*	*falso*
verdadero	*falso*	*falso*	*falso*	*verdadero*	*falso*	*falso*
verdadero	*verdadero*	*falso*	*verdadero*	*verdadero*	*verdadero*	*verdadero*

Figura 7.8 Tablas de verdad para las cinco conectivas lógicas. Para utilizar la tabla, por ejemplo, para calcular el valor de $P \vee Q$, cuando P es verdadero y Q falso, primero mire a la izquierda en donde P es *verdadera* y Q es *falsa* (la tercera fila). Entonces mire en esa fila justo en la columna de $P \vee Q$ para ver el resultado: *verdadero*. Otra forma de verlo es pensar en cada fila como en un modelo, y que sus entradas en cada fila dicen para cada columna si la sentencia es verdadera en ese modelo.

dero $\wedge$ (*falso* $\vee$ *verdadero*) = *verdadero* $\wedge$ *verdadero* = *verdadero*. El Ejercicio 7.3 pide que escriba el algoritmo ¿V-VERDAD-LP?(s, m) que debe obtener el valor de verdad de una sentencia s en lógica proposicional según el modelo m.

Ya hemos comentado que una base de conocimiento está compuesta por sentencias. Ahora podemos observar que esa base de conocimiento lógica es una conjunción de dichas sentencias. Es decir, si comenzamos con una BC vacía y ejecutamos DECIR(BC, S_1)... DECIR(BC, S_n) entonces tenemos $BC = S_1 \wedge \ldots \wedge S_n$. Esto significa que podemos manejar bases de conocimiento y sentencias de manera intercambiable.

Los valores de verdad de «y», «o» y «no» concuerdan con nuestra intuición, cuando los utilizamos en lenguaje natural. El principal punto de confusión puede presentarse cuando $P \vee Q$ es verdadero porque P lo es, Q lo es, o *ambos* lo son. Hay una conectiva diferente denominada «o exclusiva» («xor» para abreviar) que es falsa cuando los dos disyuntores son verdaderos[9]. No hay consenso respecto al símbolo que representa la o exclusiva, siendo las dos alternativas $\dot{\vee}$ y $\oplus$.

El valor de verdad de la conectiva $\Rightarrow$ puede parecer incomprensible al principio, ya que no encaja en nuestra comprensión intuitiva acerca de «P implica Q» o de «si P entonces Q». Para una cosa, la lógica proposicional no requiere de una relación de causalidad o relevancia entre P y Q. La sentencia «que 5 sea impar implica que Tokio es la capital de Japón» es una sentencia verdadera en lógica proposicional (bajo una interpretación normal), aunque pensándolo es, decididamente, una frase muy rara. Otro punto de confusión es que cualquier implicación es verdadera siempre que su antecedente sea falso. Por ejemplo, «que 5 sea par implica que Sam es astuto» es verdadera, independientemente de que Sam sea o no astuto. Parece algo estrafalario, pero tiene sentido si piensa acerca de «$P \Rightarrow Q$» como si dijera, «si P es verdadero, entonces estoy afirmando que Q es verdadero. De otro modo, no estoy haciendo ninguna afirmación.» La única manera de hacer esta sentencia *falsa* es haciendo que P sea cierta y Q falsa.

La tabla de verdad de la bicondicional $P \Leftrightarrow Q$ muestra que la sentencia es verdadera siempre que $P \Rightarrow Q$ y $Q \Rightarrow P$ lo son. En lenguaje natural a menudo se escribe como «P si y sólo si Q» o «P si Q». Las reglas del mundo de *wumpus* se describen mejor utilizando la conectiva $\Leftrightarrow$. Por ejemplo, una casilla tiene corriente de aire *si* alguna casilla vecina tiene un hoyo, y una casilla tiene corriente de aire *sólo si* una casilla vecina tiene un hoyo. De esta manera necesitamos bicondicionales como

$$B_{1,1} \Leftrightarrow (H_{1,2} \vee H_{2,1}),$$

en donde $B_{1,1}$ significa que hay una brisa en la casilla [1,1]. Fíjese en que la implicación

$$B_{1,1} \Rightarrow (H_{1,2} \vee H_{2,1})$$

es verdadera, aunque incompleta, en el mundo de *wumpus*. Esta implicación no descarta modelos en los que $B_{1,1}$ sea falso y $H_{1,2}$ sea verdadero, hecho que violaría las reglas del mundo de *wumpus*. Otra forma de observar esta incompletitud es que la implicación necesita la presencia de hoyos si hay una corriente de aire, mientras que la bicondicional además necesita la ausencia de hoyos si no hay ninguna corriente de aire.

---

[9] En latín está la palabra específica *aut* para la o exclusiva.