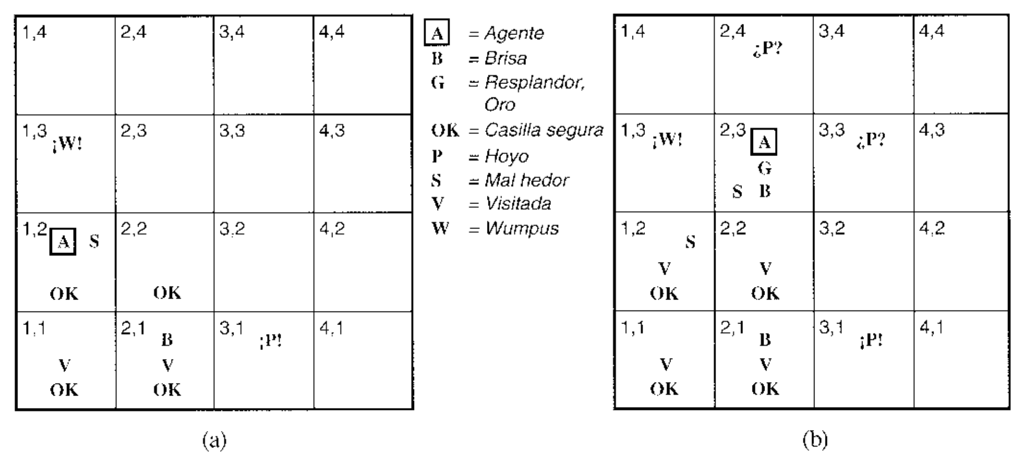

Figura 7.4 Los dos últimos estados en el desarrollo del juego. (a) Después del tercer movimiento, con la percepción [*Hedor, Nada, Nada, Nada, Nada*]. (b) Después del quinto movimiento, con la percepción [*Hedor, Brisa, Resplandor, Nada, Nada*].

El agente ha demostrado en este momento que no hay ni un hoyo ni un *wumpus* en la casilla [2, 2], así que está *OK* para desplazarse a ella. No mostraremos el estado de conocimiento del agente en [2, 2]; asumimos que el agente gira y se desplaza a [2, 3], tal como se muestra en la Figura 7.4(b). En la casilla [2, 3] el agente detecta un resplandor, entonces el agente cogería el oro y acabaría el juego.

En cada caso en que el agente saca una conclusión a partir de la información que tiene disponible, se garantiza que dicha conclusión es correcta si la información disponible también lo es. Esta es una propiedad fundamental del razonamiento lógico. En lo que queda del capítulo vamos a describir cómo construir agentes lógicos que pueden representar la información necesaria para sacar conclusiones similares a las que hemos descrito en los párrafos anteriores.

7.3 Lógica

Esta sección presenta un repaso de todos los conceptos fundamentales de la representación y el razonamiento lógicos. Dejamos los detalles técnicos de cualquier clase concreta de lógica para la siguiente sección. En lugar de ello, utilizaremos ejemplos informales del mundo de *wumpus* y del ámbito familiar de la aritmética. Adoptamos este enfoque poco común, porque los conceptos de la lógica son bastante más generales y bellos de lo que se piensa a priori.

SINTAXIS

En la sección 7.1 dijimos que las bases de conocimiento se componen de sentencias. Estas sentencias se expresan de acuerdo a la **sintaxis** del lenguaje de representación, que especifica todas las sentencias que están bien formadas. El concepto de sintaxis está suficientemente claro en la aritmética: «$x + y = 4$» es una sentencia bien formada, mientras que «$x2y+ =$» no lo es. Por lo general, la sintaxis de los lenguajes lógicos (y la de los aritméticos, en cuanto al mismo tema) está diseñada para escribir libros y artículos. Hay literalmente docenas de diferentes sintaxis, algunas que utilizan muchas letras griegas y símbolos matemáticos complejos, otras basadas en diagramas con flechas y burbujas visualmente muy atractivas. Y sin embargo, en todos estos casos, las sentencias de la base de conocimiento del agente son configuraciones físicas reales (de las partes) del agente. El razonamiento implica generar y manipular estas configuraciones.

SEMÁNTICA

VALOR DE VERDAD

MUNDO POSIBLE

Una lógica también debe definir la **semántica** del lenguaje. Si lo relacionamos con el lenguaje hablado, la semántica trata el «significado» de las sentencias. En lógica, esta definición es bastante más precisa. La semántica del lenguaje define el **valor de verdad** de cada sentencia respecto a cada **mundo posible**. Por ejemplo, la semántica que se utiliza en la aritmética especifica que la sentencia «$x + y = 4$» es verdadera en un mundo en el que x sea 2 e y sea 2, pero falsa en uno en el que x sea 1 e y sea 1[1]. En las lógicas clásicas cada sentencia debe ser o bien verdadera o bien falsa en cada mundo posible, no puede ser lo uno y lo otro[2].

MODELO

Cuando necesitemos ser más precisos, utilizaremos el término **modelo** en lugar del de «mundo posible». (También utilizaremos la frase «m es un modelo de α» para indicar que la sentencia α es verdadera en el modelo m.) Siempre que podamos pensar en los mundos posibles como en (potencialmente) entornos reales en los que el agente pueda o no estar, los modelos son abstracciones matemáticas que simplemente nos permiten definir la verdad o falsedad de cada sentencia que sea relevante. Informalmente podemos pensar, por ejemplo, en que x e y son el número de hombres y mujeres que están sentados en una mesa jugando una partida de *bridge*, y que la sentencia $x + y = 4$ es verdadera cuando los que están jugando son cuatro en total; formalmente, los modelos posibles son justamente todas aquellas posibles asignaciones de números a las variables x e y. Cada una de estas asignaciones indica el valor de verdad de cualquier sentencia aritmética cuyas variables son x e y.

IMPLICACIÓN

Ahora que ya disponemos del concepto de valor de verdad, ya estamos preparados para hablar acerca del razonamiento lógico. Éste requiere de la relación de **implicación** lógica entre las sentencias (la idea de que una sentencia *se sigue lógicamente* de otra sentencia). Su notación matemática es

$$\alpha \models \beta$$

para significar que la sentencia α implica la sentencia β. La definición formal de implicación es esta: $\alpha \models \beta$ si y sólo si en cada modelo en el que α es verdadera, β también lo es. Otra forma de definirla es que si α es verdadera, β también lo debe ser. Informalmente, el valor de verdad de β «está contenido» en el valor de verdad de α. La relación de implicación nos es familiar en la aritmética; no nos disgusta la idea de que la sentencia $x + y = 4$ implica la sentencia $4 = x + y$. Es obvio que en cada modelo en

[1] El lector se habrá dado cuenta de la semejanza entre el concepto de valor de verdad de las sentencias y la satisfacción de restricciones del Capítulo 5. No es casualidad (los lenguajes de restricciones son en efecto lógicas y la resolución de restricciones un tipo de razonamiento lógico).

[2] La **lógica difusa**, que se verá en el Capítulo 14, nos permitirá tratar con grados de valores de verdad.

el que $x + y = 4$ (como lo es el modelo en el que x es 2 e y es 2) también lo es para $4 = x + y$. Pronto veremos que una base de conocimiento puede ser considerada como una afirmación, y a menudo hablaremos de que una base de conocimiento implica una sentencia.

Ahora podemos aplicar el mismo tipo de análisis que utilizamos en la sección anterior al mundo del *wumpus*. Si tomamos la situación de la Figura 7.3(b): el agente no ha detectado nada en la casilla [1, 1], y ha detectado una brisa en la [2, 1]. Estas percepciones, combinadas con el conocimiento del agente sobre las reglas que definen el funcionamiento del mundo de *wumpus* (la descripción REAS de la página 221), constituyen su BC. El agente está interesado (entre otras cosas) en si las casillas adyacentes [1, 2], [2, 2] y [3, 1] tienen hoyos sin fondo. Cada una de las tres casillas pueden o no tener un hoyo, por lo tanto (al menos en este ejemplo) hay $2^3 = 8$ modelos posibles. Tal como se muestran en la Figura 7.5[3].

La BC es falsa en los modelos que contradicen lo que el agente sabe (por ejemplo, la BC es falsa en cualquier modelo en el que la casilla [1, 2] tenga un hoyo), porque no ha detectado ninguna brisa en la casilla [1, 1]. De hecho, hay tres modelos en los que la BC es verdadera, los que se muestran como subconjunto de los modelos de la Figura 7.5. Ahora consideremos las dos conclusiones:

α_1 = «No hay un hoyo en la casilla [1, 2]».
α_2 = «No hay un hoyo en la casilla [2, 2]».

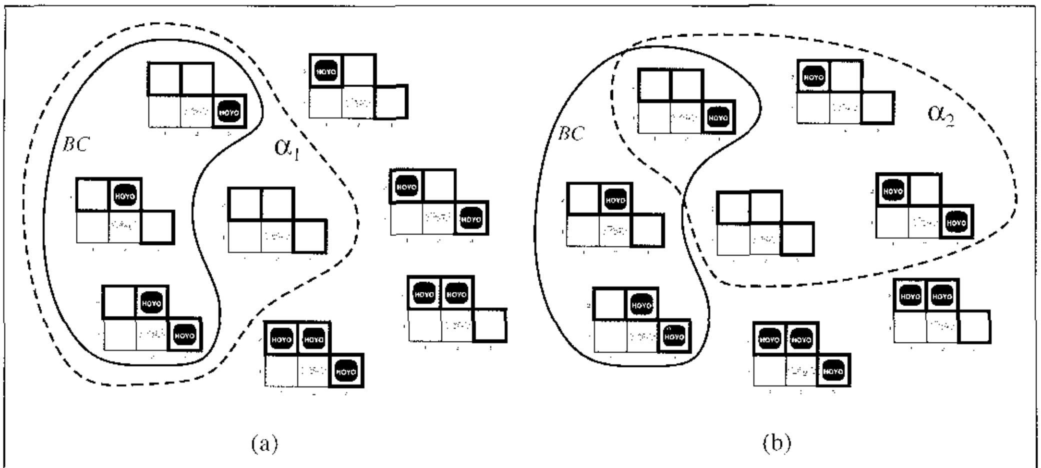

Figura 7.5 Modelos posibles para la presencia de hoyos en las casillas [1, 2], [2, 2] y [3, 1], dadas las observaciones de que no hay nada en la casilla [1, 1] y hay una brisa en la [2, 1]. (a) Modelos de la base de conocimiento y α_1 (no hay un hoyo en [1, 2]). (b) Modelos de la base de conocimiento y α_2 (no hay un hoyo en [2, 2]).

[3] En la Figura 7.5 los modelos se muestran como mundos parciales, porque en realidad tan sólo son asignaciones de *verdadero* y *falso* a sentencias como «hay un hoyo en la casilla [1, 2]», etc. Los modelos, desde el punto de vista matemático, no necesitan tener horribles wumpus etéreos ambulando en ellos.

Hemos rodeado (con línea discontinua) los modelos de α_1 y α_2 en las Figuras 7.5(a) y 7.5(b) respectivamente. Si observamos, podemos ver lo siguiente:

en cada modelo en el que la BC es verdadera, α_1 también lo es.

De aquí que $BC \models \alpha_1$: no hay un hoyo en la casilla [1, 2]. También podemos ver que

en algunos modelos en los que la BC es verdadera, α_2 es falsa.

De aquí que $BC \not\models \alpha_2$: el agente no puede concluir que no haya un hoyo en la casilla [2, 2]. (Ni tampoco puede concluir que lo haya.[4])

El ejemplo anterior no sólo nos muestra el concepto de implicación, sino, también cómo el concepto de implicación se puede aplicar para derivar conclusiones, es decir, llevar a cabo la **inferencia lógica**. El algoritmo de inferencia que se muestra en la Figura 7.5 se denomina **comprobación de modelos** porque enumera todos los modelos posibles y comprueba si α es verdadera en todos los modelos en los que la BC es verdadera.

INFERENCIA LÓGICA

COMPROBACIÓN DE MODELOS

Para entender la implicación y la inferencia nos puede ayudar pensar en el conjunto de todas las consecuencias de la BC como en un pajar, y en α como en una aguja. La implicación es como la aguja que se encuentra en el pajar, y la inferencia consiste en encontrarla. Esta distinción se expresa mediante una notación formal: si el algoritmo de inferencia i puede derivar α de la BC, entonces escribimos

$$BC \vdash_i \alpha,$$

que se pronuncia como «α se deriva de la BC mediante i» o «i deriva α de la BC».

Se dice que un algoritmo de inferencia que deriva sólo sentencias implicadas es **sólido** o que **mantiene la verdad**. La solidez es una propiedad muy deseable. Un procedimiento de inferencia no sólido tan sólo se inventaría cosas poco a poco (anunciaría el descubrimiento de agujas que no existirían). Se puede observar fácilmente que la comprobación de modelos, cuando es aplicable[5], es un procedimiento sólido.

SÓLIDO

MANTENIMIENTO DE LA VERDAD

También es muy deseable la propiedad de **completitud**: un algoritmo de inferencia es completo si puede derivar cualquier sentencia que está implicada. En los pajares reales, que son de tamaño finito, parece obvio que un examen sistemático siempre permite decidir si hay una aguja en el pajar. Sin embargo, en muchas bases de conocimiento, el pajar de las consecuencias es infinito, y la completitud pasa a ser una problemática importante[6]. Por suerte, hay procedimientos de inferencia completos para las lógicas que son suficientemente expresivas para manejar muchas bases de conocimiento.

COMPLETITUD

[4] El agente podría calcular la *probabilidad* de que haya un hoyo en la casilla [2, 2]; en el Capítulo 13 lo veremos.

[5] La comprobación de modelos trabaja bien cuando el espacio de los modelos es finito (por ejemplo, en un mundo del *wumpus* de tamaño de casillas fijo). Por el otro lado, en la aritmética, el espacio de modelos es infinito: aun limitándonos a los enteros, hay infinitos pares de valores para x e y en la sentencia $x + y = 4$.

[6] Compárelo con el caso de la búsqueda en espacios infinitos de estados del Capítulo 3, en donde la búsqueda del primero en profundidad no es completa.