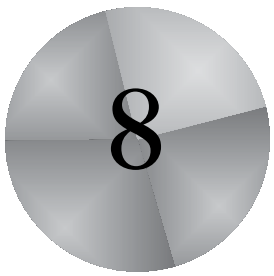

8 Lógica de primer orden

Donde nos daremos cuenta de que el mundo está bendecido con muchos objetos, que algunos de los cuales están relacionados con otros objetos, y nos esforzamos en razonar sobre ellos.

LÓGICA DE PRIMER ORDEN

En el Capítulo 7 demostramos cómo un agente basado en el conocimiento podía representar el mundo en el que actuaba y deducir qué acciones debía llevar a cabo. En el capítulo anterior utilizamos la lógica proposicional como nuestro lenguaje de representación, porque nos bastaba para ilustrar los conceptos fundamentales de la lógica y de los agentes basados en el conocimiento. Desafortunadamente, la lógica proposicional es un lenguaje demasiado endeble para representar de forma precisa el conocimiento de entornos complejos. En este capítulo examinaremos la **lógica de primer orden**[1] que es lo suficientemente expresiva como para representar buena parte de nuestro conocimiento de sentido común. La lógica de predicados también subsume, o forma la base para, muchos otros lenguajes de representación y ha sido estudiada intensamente durante varias décadas. En la Sección 8.1 comenzamos con una breve discusión general acerca de los lenguajes de representación; en la Sección 8.2, se muestra la sintaxis y la semántica de la lógica de primer orden; y en la Sección 8.3 se ilustra el uso de la lógica de primer orden en representaciones sencillas.

8.1 Revisión de la representación

En esta sección, discutiremos acerca de la naturaleza de los lenguajes de representación. Esta discusión nos llevará al desarrollo de la lógica de primer orden, un lenguaje mu-

[1] También denominada **Cálculo de Predicados** (de **Primer Orden**), algunas veces se abrevia mediante LP o CP.

cho más expresivo que la lógica proposicional, que introdujimos en el Capítulo 7. Veremos la lógica proposicional y otros tipos de lenguajes para entender lo que funciona y lo que no. Nuestra discusión será rápida, resumiendo siglos del pensamiento humano, de ensayo y error, todo ello en unos pocos párrafos.

Los lenguajes de programación (como C++, o Java, o Lisp) son, de lejos, la clase más amplia de lenguajes formales de uso común. Los programas representan en sí mismos, y de forma directa, sólo procesos computacionales. Las estructuras de datos de los programas pueden representar hechos; por ejemplo, un programa puede utilizar una matriz de 4×4 para representar el contenido del mundo de *wumpus*. De esta manera, una sentencia de un lenguaje de programación como $Mundo[2, 2] \leftarrow Hoyo$, es una forma bastante natural de expresar que hay un hoyo en la casilla [2, 2]. (A estas representaciones se les puede considerar *ad hoc*; los sistemas de bases de datos fueron desarrollados para proporcionar una manera más genérica, e independiente del dominio, de almacenar y recuperar hechos.) De lo que carecen los lenguajes de programación, es de algún mecanismo general para derivar hechos de otros hechos; cada actualización de la estructura de datos se realiza mediante un procedimiento específico del dominio, cuyos detalles se derivan del conocimiento acerca del dominio del o de la programadora. Este enfoque **procedural** puede contrastar con la naturaleza declarativa de la lógica proposicional, en la que el conocimiento y la inferencia se encuentran separados, y en la que la inferencia se realiza de forma totalmente independiente del dominio.

Otro inconveniente de las estructuras de datos de los programas (y de las bases de datos, respecto a este tema) es la falta de un mecanismo sencillo para expresar, por ejemplo, «Hay un hoyo en la casilla [2, 2] o en la [3, 1]» o «Si hay un *wumpus* en la casilla [1, 1] entonces no hay ninguno en la [2, 2]». Los programas pueden almacenar un valor único para cada variable, y algunos sistemas permiten el valor «desconocido», pero carecen de la expresividad que se necesita para manejar información incompleta.

La lógica proposicional es un lenguaje declarativo porque su semántica se basa en la relación de verdad entre las sentencias y los mundos posibles. Lo que tiene el suficiente poder expresivo para tratar información incompleta, mediante la disyunción y la conjunción. La lógica proposicional presenta una tercera característica que es muy deseable en los lenguajes de representación, a saber, la **composicionalidad**. En un lenguaje composicional, el significado de una sentencia es una función del significado de sus partes. Por ejemplo, «$M_{1,4} \wedge M_{1,2}$» está relacionada con los significados de «$M_{1,4}$» y «$M_{1,2}$». Sería muy extraño que «$M_{1,4}$» significara que hay mal hedor en la casilla [1, 4], que «$M_{1,2}$» significara que hay mal hedor en la casilla [1, 2], y que en cambio, «$M_{1,4} \wedge M_{1,2}$» significara que Francia y Polonia empataran en el partido de jockey de calificación de la última semana. Está claro que la no composicionalidad repercute en que al sistema de razonamiento le sea mucho más difícil subsistir.

COMPOSICIONAL

Tal como vimos en el Capítulo 7, la lógica proposicional carece del poder expresivo para describir de forma *precisa* un entorno con muchos objetos. Por ejemplo, estábamos forzados a escribir reglas separadas para cada casilla al hablar acerca de las brisas y de los hoyos, tal como

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

Por otro lado, en el lenguaje natural parece bastante sencillo decir, de una vez por todas, que «En las casillas adyacentes a hoyos se percibe una pequeña brisa». De alguna manera, la sintaxis y la semántica del lenguaje natural nos hace posible describir el entorno de forma precisa.

De hecho, si lo pensamos por un momento, los lenguajes naturales (como el inglés o el castellano) son muy expresivos. Hemos conseguido escribir casi la totalidad de este libro en lenguaje natural, sólo con lapsos ocasionales en otros lenguajes (incluyendo la lógica, las matemáticas, y los lenguaje de diagramas visuales). En la lingüística y la filosofía del lenguaje hay una larga tradición que ve el lenguaje natural esencialmente como un lenguaje declarativo de representación del conocimiento e intenta definir su semántica formal. Como en un programa de investigación, si tuviera éxito sería de gran valor para la inteligencia artificial, porque esto permitiría utilizar un lenguaje natural (o alguna variación) con los sistemas de representación y razonamiento.

El punto de vista actual sobre el lenguaje natural es que sirve para un propósito algo diferente, a saber, como un medio de **comunicación** más que como una pura representación. Cuando una persona señala y dice, «¡Mira!», el que le oye llega a saber que lo que dice es que Superman finalmente ha aparecido sobre los tejados. Con ello, no queremos decir que la sentencia «¡Mira!» expresa ese hecho. Más bien, que el significado de la sentencia depende tanto de la propia sentencia como del **contexto** al que la sentencia hace referencia. Está claro que uno no podría almacenar una sentencia como «¡Mira!» en una base de conocimiento y esperar recuperar su significado sin haber almacenado también una representación de su contexto, y esto revela la problemática de cómo se puede representar el propio contexto. Los lenguajes naturales también son composicionales (el significado de una sentencia como «Entonces ella lo vio» puede depender de un contexto construido a partir de muchas sentencias precedentes y posteriores a ella). Por último, los lenguajes naturales sufren de la **ambigüedad**, que puede causar ciertos obstáculos en su comprensión. Tal como comenta Pinker (1995): «Cuando la gente piensa acerca de la *primavera*, seguramente no se confunden sobre si piensan acerca de una estación o algo que va ¿boing? (y si una palabra se puede corresponder con dos pensamientos, los pensamientos no pueden ser palabras).»

Nuestro enfoque consistirá en adoptar los fundamentos de la lógica proposicional (una semántica composicional declarativa que es independiente del contexto y no ambigua) y construir una lógica más expresiva basada en dichos fundamentos, tomando prestadas de los lenguajes naturales las ideas acerca de la representación, al mismo tiempo que evitando sus inconvenientes. Cuando observamos la sintaxis del lenguaje natural, los elementos más obvios son los nombres y las sentencias nominales que se refieren a los **objetos** (casillas, hoyos, *wumpus*) y los verbos y las sentencias verbales que se refieren a las **relaciones** entre los objetos (en la casilla se percibe una brisa, la casilla es adyacente a, el agente lanza una flecha). Algunas de estas relaciones son **funciones** (relaciones en las que dada una «entrada» se obtiene un solo «valor»). Es fácil empezar a listar ejemplos de objetos, relaciones y funciones:

OBJETOS

RELACIONES

FUNCIONES

- Objetos: gente, casas, números, teorías, Ronald McDonald, colores, partidos de béisbol, guerras, siglos…

PROPIEDADES

- Relaciones: éstas pueden ser relaciones unitarias, o **propiedades**, como ser de color rojo, ser redondo, ser ficticio, ser un número primo, ser multihistoriado…, o relaciones *n*-arias más generales, como ser hermano de, ser más grande que, estar dentro de, formar parte de, tener color, ocurrir después de, ser dueño de uno mismo, o interponerse entre…
- Funciones: el padre de, el mejor amigo de, el tercer turno, uno mayor que, el comienzo de…

Efectivamente, se puede pensar en casi cualquier aserción como una referencia a objetos y propiedades o relaciones. Como los siguientes ejemplos:

- «Uno sumado a dos es igual a tres.»
  Objetos: uno, dos, tres, uno sumado a dos; Relaciones: es igual a; Funciones: sumado a. («Uno sumado a dos» es el nombre de un objeto que se obtiene aplicando la función «sumado a» a los objetos «uno» y «dos». Hay otro nombre para este objeto.)
- «Las casillas que rodean al *wumpus* son pestilentes.»
  Objetos: *wumpus*, casillas; Propiedad: pestilente; Relación: rodear a.
- «El malvado rey Juan gobernó Inglaterra en 1200.»
  Objetos: Juan, Inglaterra, 1200; Relación: gobernar; Propiedades: malvado, rey.

El lenguaje de la **lógica de primer orden**, cuya sintaxis y semántica definiremos en la siguiente sección, está construido sobre objetos y relaciones. Precisamente por este motivo ha sido tan importante para las Matemáticas, la Filosofía y la inteligencia artificial (y en efecto, en el día a día de la existencia humana) porque se puede pensar en ello de forma utilitaria como en el tratamiento con objetos y de las relaciones entre éstos. La lógica de primer orden también puede expresar hechos acerca de *algunos* o *todos* los objetos de un universo de discurso. Esto nos permite representar leyes generales o reglas, tales como el enunciado «Las casillas que rodean al *wumpus* son pestilentes».

COMPROMISO ONTOLÓGICO

La principal diferencia entre la lógica proposicional y la de primer orden se apoya en el **compromiso ontológico** realizado por cada lenguaje (es decir, lo que asume cada uno acerca de la naturaleza de la *realidad*). Por ejemplo, la lógica proposicional asume que hay hechos que suceden o no suceden en el mundo. Cada hecho puede estar en uno de los dos estados: verdadero o falso[2]. La lógica de primer orden asume mucho más, a saber, que el mundo se compone de objetos con ciertas relaciones entre ellos que suceden o no suceden. Las lógicas de propósito específico aún hacen compromisos ontológicos más allá; por ejemplo, la **lógica temporal** asume que los hechos suceden en *tiempos* concretos y que esos tiempos (que pueden ser instantes o intervalos) están ordenados. De esta manera, las lógicas de propósito específico dan a ciertos tipos de objetos (y a los axiomas acerca de ellos) un estatus de «primera clase» dentro de la lógica, más que simplemente definiéndolos en la base de conocimiento. La **lógica de orden superior** ve las relaciones y funciones que se utilizan en la lógi-

LÓGICA TEMPORAL

LÓGICA DE ORDEN SUPERIOR

[2] A diferencia de los hechos en la **lógica difusa**, que tienen un **grado de verdad** entre 0 y 1. Por ejemplo, la sentencia «Vietnam es una gran ciudad» podría ser verdadera sólo con un grado 0,6 en nuestro mundo.

EL LENGUAJE DEL PENSAMIENTO

Los filósofos y los psicólogos han reflexionado profundamente sobre cómo representan el conocimiento los seres humanos y otros animales. Está claro que la evolución del lenguaje natural ha jugado un papel importante en el desarrollo de esta habilidad en los seres humanos. Por otro lado, muchas evidencias en la Psicología sugieren que los seres humanos no utilizan el lenguaje de forma directa en sus representaciones internas. Por ejemplo ¿cuál de las dos siguientes frases formaba el inicio de la Sección 8.1?

«En esta sección, discutiremos acerca de la naturaleza de los lenguajes de representación…»

«Esta sección cubre el tema de los lenguajes de representación del conocimiento…»

Wanner (1974) encontró que los sujetos hacían la elección acertada en los tests a un nivel casual (cerca del 50 por ciento de las veces) pero que recordaban el contenido de lo que habían leído con un 90 por ciento de precisión. Esto sugiere que la gente procesa las palabras para formar algún tipo de representación no verbal, lo que llamamos **memoria**.

El mecanismo concreto mediante el cual el lenguaje permite la representación y modela las ideas en los seres humanos sigue siendo una incógnita fascinante. La famosa hipótesis de **Sapir-Whorf** sostiene que el lenguaje que hablamos influye profundamente en la manera en que pensamos y tomamos decisiones, en concreto, estableciendo las estructuras de categorías con las que separamos el mundo en diferentes agrupaciones de objetos. Whorf (1956) sostuvo que los esquimales tenían muchas palabras para la nieve, así que tenían una experiencia de la nieve diferente de las personas que hablaban otros idiomas. Algunos lingüistas no están de acuerdo con el fundamento factual de esta afirmación (Pullum (1991) argumenta que el Inuit, el Yupik, y otros lenguajes similares parece que tienen un número parecido de palabras que el inglés para los conceptos relacionados con la nieve) mientras que otros apoyan dicha afirmación (Fortescue, 1984). Parece perfectamente comprensible que las poblaciones que tienen una familiaridad mayor con algunos aspectos del mundo desarrollan un vocabulario mucho más detallado en dichos temas, por ejemplo, los entomólogos dividen lo que muchos de nosotros llamamos simplemente *escarabajos* en cientos de miles de especies y además están personalmente familiarizados con muchas de ellas. (El biólogo evolucionista J. B. S. Haldane una vez se quejó de «Una afición desmesurada a los escarabajos» por parte del Creador.) Más aún, los esquiadores expertos tienen muchos términos para la nieve (en polvo, sopa de pescado, puré de patatas, cruda, maíz, cemento, pasta, azúcar, asfalto, pana, pelusa, etcétera) que representan diferencias que a los profanos no nos son familiares. Lo que no está claro es la dirección de la causalidad (¿los esquiadores se dan cuenta de las diferencias sólo por aprender las palabras, o las diferencias surgen de la experiencia individual y llegan a emparejarse con las etiquetas que se están utilizando en el grupo?) Esta cuestión es especialmente importante en el estudio del desarrollo infantil. Hasta ahora disponemos de poco entendimiento acerca del grado en el cual el aprendizaje del lenguaje y el razonamiento están entrelazados. Por ejemplo, ¿el conocimiento del nombre de un concepto, como *licenciado*, hace que nos sea más fácil construir y razonar acerca de conceptos más complejos que engloban a dicho nombre, como *licenciado idóneo*?

ca de primer orden como objetos en sí mismos. Esto nos permite hacer aserciones acerca de *todas* las relaciones, por ejemplo, uno podría desear definir lo que significa que una relación sea transitiva. A diferencia de las lógicas de propósito específico, la lógica de orden superior es estrictamente más expresiva que la lógica de primer orden, en el sentido de que algunas sentencias de la lógica de orden superior no se pueden expresar mediante un número finito de sentencias de la lógica de primer orden.

COMPROMISOS EPISTEMOLÓGICOS

Una lógica también se puede caracterizar por sus **compromisos epistemológicos** (los posibles estados del conocimiento respecto a cada hecho que la lógica permite). Tanto en la lógica proposicional como en la de primer orden, una sentencia representa un hecho y el agente o bien cree que la sentencia es verdadera, o cree que la sentencia es falsa, o no tiene ninguna opinión. Por lo tanto, estas lógicas tienen tres estados posibles de conocimiento al considerar cualquier sentencia. Por otro lado, los sistemas que utilizan la **teoría de las probabilidades** pueden tener un *grado de creencia*, que va de cero (no se cree en absoluto) a uno (se tiene creencia total)[3]. Por ejemplo, un agente del mundo de *wumpus* que utilice las probabilidades podría creer que el *wumpus* se encuentra en la casilla [1, 3] con una probabilidad de 0,75. En la Figura 8.1 se resumen los compromisos ontológicos y epistemológicos de cinco lógicas distintas.

En la siguiente sección nos meteremos en los detalles de la lógica de primer orden. Al igual que un estudiante de Física necesita familiarizarse con las Matemáticas, un estudiante de IA debe desarrollar sus capacidades para trabajar con la notación lógica. Por otro lado, *no* es tan importante conseguir una preocupación desmesurada sobre las *especificaciones* de la notación lógica (al fin y al cabo, hay docenas de versiones distintas). Los conceptos principales que hay que tener en cuenta son cómo el lenguaje nos facilita una representación precisa y cómo su semántica nos permite realizar procedimientos sólidos de razonamiento.

Lenguaje	Compromiso ontológico (lo que sucede en el mundo)	Compromiso epistemológico (lo que el agente cree acerca de los hechos)
Lógica proposicional	Hechos	Verdadero/falso/desconocido
Lógica de primer orden	Hechos, objetos, relaciones	Verdadero/falso/desconocido
Lógica temporal	Hechos, objetos, relaciones, tiempos	Verdadero/falso/desconocido
Teoría de las probabilidades	Hechos	Grado de creencia $\in$ [0, 1]
Lógica difusa	Hechos con un grado de verdad $\in$ [0, 1]	Valor del intervalo conocido

Figura 8.1 Lenguajes formales y sus compromisos ontológicos y epistemológicos.

[3] Es importante no confundir el grado de creencia de la teoría de probabilidades con el grado de verdad de la lógica difusa. Realmente, algunos sistemas difusos permiten una incertidumbre (un grado de creencia) acerca de los grados de verdad.

8.2 Sintaxis y semántica de la lógica de primer orden

Comenzamos esta sección especificando de forma más precisa la forma en la que los mundos posibles en la lógica de primer orden reflejan el compromiso ontológico respecto a los objetos y las relaciones. Entonces introducimos los diferentes elementos del lenguaje, explicando su semántica a medida que avanzamos.

Modelos en lógica de primer orden

DOMINIO

ELEMENTOS DEL DOMINIO

Recuerde del Capítulo 7 que los modelos en un lenguaje lógico son las estructuras formales que establecen los mundos posibles que se tienen en cuenta. Los modelos de la lógica proposicional son sólo conjuntos de valores de verdad para los símbolos proposicionales. Los modelos de la lógica de primer orden son más interesantes. Primero, ¡éstos contienen a los objetos! El **dominio** de un modelo es el conjunto de objetos que contiene; a estos objetos a veces se les denomina **elementos del dominio**. La Figura 8.2 muestra un modelo con cinco objetos: Ricardo Corazón de León, Rey de Inglaterra de 1189 a 1199; su hermano más joven, el malvado Rey Juan, quien reinó de 1199 a 1215; las piernas izquierda de Ricardo y Juan; y una corona.

Los objetos en el modelo pueden estar relacionados de diversas formas. En la figura, Ricardo y Juan son hermanos. Hablando formalmente, una relación es sólo un con-

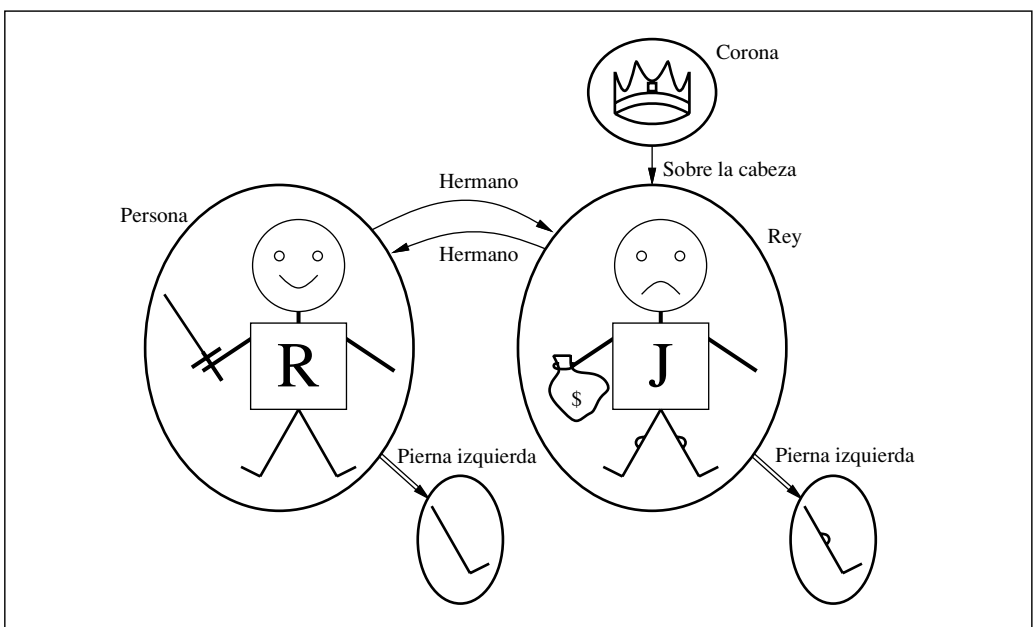

Figura 8.2 Un modelo que contiene cinco objetos, dos relaciones binarias, tres relaciones unitarias (indicadas mediante etiquetas sobre los objetos), y una función unitaria: pierna izquierda.

TUPLAS

junto de **tuplas** de objetos que están relacionados. (Una tupla es una colección de objetos colocados en un orden fijo que se escriben entre paréntesis angulares.) De esta manera, la relación de hermandad en este modelo es el conjunto

$$\{\langle \text{Ricardo Corazón de León, Rey Juan}\rangle, \langle \text{Rey Juan, Ricardo Corazón de León}\rangle\} \quad (8.1)$$

(Aquí hemos nombrado los objetos en español, pero se puede, si uno lo desea, sustituir mentalmente los nombres por las imágenes.) La corona está colocada sobre la cabeza del Rey Juan, así que la relación «sobre la cabeza» contiene sólo una tupla, ⟨Corona, Rey Juan⟩. Las relaciones «hermano» y «sobre la cabeza» son relaciones binarias, es decir, relacionan parejas de objetos. El modelo también contiene relaciones unitarias, o propiedades: la propiedad «ser persona» es verdadera tanto para Ricardo como para Juan; la propiedad «ser rey» es verdadera sólo para Juan (presumiblemente porque Ricardo está muerto en este instante); y la propiedad «ser una corona» sólo es verdadera para la corona.

Hay ciertos tipos de relaciones que es mejor que se consideren como funciones; en estas relaciones un objeto dado debe relacionarse exactamente con otro objeto. Por ejemplo, cada persona tiene una pierna izquierda, entonces el modelo tiene la función unitaria «pierna izquierda» con las siguientes aplicaciones:

$$\begin{aligned}\langle \text{Ricardo Corazón de León}\rangle &\rightarrow \text{pierna izquierda de Ricardo}\\ \langle \text{Rey Juan}\rangle &\rightarrow \text{pierna izquierda de Juan}\end{aligned} \quad (8.2)$$

FUNCIONES TOTALES

Hablando de forma estricta, los modelos en la lógica de primer orden requieren **funciones totales**, es decir, debe haber un valor para cada tupla de entrada. Así, la corona debe tener una pierna izquierda y por lo tanto, también cada una de las piernas izquierdas. Hay una solución técnica para este problema inoportuno, incluyendo un objeto «invisible» adicional, que es la pierna izquierda de cada cosa que no tiene pierna izquierda, inclusive ella misma. Afortunadamente, con tal de que uno no haga aserciones acerca de piernas izquierdas de cosas que no tienen piernas izquierdas, estos tecnicismos dejan de tener importancia.

Símbolos e interpretaciones

Ahora volvemos a la sintaxis del lenguaje. El lector impaciente puede obtener una descripción completa de la gramática formal de la lógica de primer orden en la Figura 8.3.

Los elementos sintácticos básicos de la lógica de primer orden son los símbolos que representan los objetos, las relaciones y las funciones. Por consiguiente, los símbolos se agrupan en tres tipos: **símbolos de constante**, que representan objetos; **símbolos de predicado**, que representan relaciones; y **símbolos de función**, que representan funciones. Adoptamos la convención de que estos símbolos comiencen en letra mayúscula. Por ejemplo, podríamos utilizar los símbolos de constante *Ricardo* y *Juan*; los símbolos de predicado *Hermano*, *SobreCabeza*, *Persona*, *Rey* y *Corona*; y el símbolo de función *PiernaIzquierda*. Al igual que con los símbolos proposicionales, la selección de los nombres depende enteramente del usuario. Cada símbolo de predicado y de función tiene una **aridad** que establece su número de argumentos.

SÍMBOLOS DE CONSTANTE

SÍMBOLOS DE PREDICADO

SÍMBOLOS DE FUNCIÓN

ARIDAD

$$
\begin{array}{rcl}
\textit{Sentencia} & \rightarrow & \textit{SentenciaAtómica} \\
& | & (\textit{Sentencia Conectiva Sentencia}) \\
& | & \textit{Cuantificador Variable}\ldots\ \textit{Sentencia} \\
& | & \neg \textit{Sentencia} \\
\\
\textit{SentenciaAtómica} & \rightarrow & \textit{Predicado}(\textit{Término}\ldots) \mid \textit{Término} = \textit{Término} \\
\\
\textit{Término} & \rightarrow & \textit{Función}(\textit{Término}\ldots) \\
& | & \textit{Constante} \\
& | & \textit{Variable} \\
\\
\textit{Conectiva} & \rightarrow & \Rightarrow \mid \wedge \mid \vee \mid \Leftrightarrow \\
\textit{Cuantificador} & \rightarrow & \forall \mid \exists \\
\textit{Constante} & \rightarrow & A \mid X_1 \mid \textit{Juan} \mid \cdots \\
\textit{Variable} & \rightarrow & a \mid x \mid s \mid \cdots \\
\textit{Predicado} & \rightarrow & \textit{AntesDe} \mid \textit{TieneColor} \mid \textit{EstáLLoviendo} \mid \cdots \\
\textit{Función} & \rightarrow & \textit{Madre} \mid \textit{PiernaIzquierda} \mid \cdots
\end{array}
$$

Figura 8.3 La sintaxis de la lógica de primer orden con igualdad, espicificada en BNF. (Mire la página 984 si no estás familiarizado con esta notación.) La sintaxis es estricta con el tema de los paréntesis; los comentarios acerca de los paréntesis y la precedencia de los operadores de la página 230 se aplica de la misma forma a la lógica de primer orden.

INTERPRETACIÓN

INTERPRETACIÓN DESEADA

La semántica debe relacionar las sentencias con los modelos para determinar su valor de verdad. Para que esto ocurra, necesitamos de una **interpretación** que especifique exactamente qué objetos, relaciones y funciones son referenciados mediante símbolos de constante, de predicados y de función, respectivamente. Una interpretación posible para nuestro ejemplo (a la que llamaremos **interpretación deseada**) podría ser la siguiente:

- *Ricardo* se refiere a Ricardo Corazón de León y *Juan* se refiere al malvado Rey Juan.
- *Hermano* se refiere a la relación de hermandad, es decir, al conjunto de tuplas de objetos que se muestran en la Ecuación (8.1); *SobreCabeza* se refiere a la relación «sobre la cabeza» que sucede entre la corona y el Rey Juan; *Persona*, *Rey* y *Corona* se refieren a los conjuntos de objetos que son personas, reyes y coronas.
- *PiernaIzquierda* se refiere a la función «pierna izquierda», es decir, la aplicación que se muestra en la Ecuación (8.2).

Hay muchas otras interpretaciones posibles que se relacionan con estos símbolos para este modelo en concreto. Por ejemplo, una interpretación podría relacionar *Ricardo* con la corona y *Juan* con la pierna izquierda del Rey Juan. Hay cinco objetos, por lo tanto hay 25 interpretaciones posibles sólo para los símbolos de constante *Ricardo* y *Juan*. Fíjese en que no todos los objetos necesitan un nombre (por ejemplo, la interpretación deseada no nombra la corona o las piernas). También es posible que un objeto tenga varios

nombres; hay una interpretación en la que tanto *Ricardo* como *Juan* se refieren a la corona. Si encuentra que le confunde esta posibilidad recuerde que en la lógica proposicional es totalmente posible tener un modelo en el que *Nublado* y *Soleado* sean ambos verdaderos; la tarea de la base de conocimiento consiste justamente en excluir lo que es inconsistente con nuestro conocimiento.

El valor de verdad de cualquier sentencia se determina por un modelo y por una interpretación de los símbolos de la sentencia. Por lo tanto, la implicación, la validez, etcétera, se determinan con base en *todos los modelos posibles* y *todas las interpretaciones posibles*. Es importante fijarse en que el número de elementos del dominio en cada modelo puede ser infinito, por ejemplo, los elementos del dominio pueden ser números enteros o reales. Por eso, el número de modelos posibles es infinito, igual que el número de interpretaciones. La comprobación de la implicación mediante la enumeración de todos los modelos posibles, que funcionaba en la lógica proposicional, no es una opción acertada para la lógica de primer orden. Aunque el número de objetos esté restringido, el número de combinaciones puede ser enorme. Con los símbolos de nuestro ejemplo, hay aproximadamente 10^{25} combinaciones para un dominio de cinco objetos. (*Véase* Ejercicio 8.5.)

Términos

TÉRMINO

Un **término** es una expresión lógica que se refiere a un objeto. Por lo tanto, los símbolos de constante son términos, pero no siempre es conveniente tener un símbolo distinto para cada objeto. Por ejemplo, en español podríamos utilizar la expresión «la pierna izquierda del Rey Juan», y sería mucho mejor que darle un nombre a su pierna. Para esto sirven los símbolos de función: en vez de utilizar un símbolo de constante utilizamos *PiernaIzquierda*(*Juan*). En el caso general, un término complejo está formado por un símbolo de función seguido de una lista de términos entre paréntesis que son los argumentos del símbolo de función. Es importante recordar que un término complejo tan sólo es un tipo de nombre algo complicado. No es una «llamada a una subrutina» que «devuelve un valor». No hay una subrutina *PiernaIzquierda* que tome una persona como entrada y devuelva una pierna. Podemos razonar acerca de piernas izquierdas (por ejemplo haciendo constar que cada uno tiene una pierna y entonces deducir que Juan debe tener una) sin tener que proporcionar una definición de *PiernaIzquierda*. Esto es algo que no se puede hacer mediante subrutinas en los lenguajes de programación[4].

La semántica formal de los términos es sencilla. Considera un término $f(t_1 \ldots, t_n)$. El símbolo de función f se refiere a alguna función del modelo (llamémosla F); los términos argumento se refieren a objetos del dominio (llamémoslos $d_1 \ldots, d_n$); y el término en su globalidad se refiere al objeto que es el valor obtenido de aplicar la función F a los

[4] Las **expresiones-λ** proporcionan una notación útil mediante la cual se construyen nuevos símbolos de función «al vuelo». Por ejemplo, la función que eleva al cuadrado su argumento se puede escribir como ($\lambda x\ x \times x$) y se puede aplicar a argumentos del mismo modo que cualquier otro símbolo de función. Una expresión-λ también se puede definir y utilizar como un símbolo de predicado. (*Véase* Capítulo 22.) El operador `lambda` del Lisp juega exactamente el mismo papel. Fíjese en que el uso de λ de este modo *no* aumenta el poder expresivo de la lógica de primer orden; porque cualquier sentencia que tenga una expresión-λ se puede rescribir «enchufando» sus argumentos para obtener una sentencia equivalente.

objetos $d_1 \ldots, d_n$. Por ejemplo, supongamos que el símbolo de función *PiernaIzquierda* se refiere a la función que se muestra en la Ecuación (8.2) y que Juan se refiere al Rey Juan, entonces, *PiernaIzquierda*(*Juan*) se refiere a la pierna izquierda del Rey Juan. De esta manera, la interpretación establece el referente de cada término.

Sentencias atómicas

Ahora que ya tenemos tanto los términos para referirnos a los objetos, como los símbolos de predicado para referirnos a las relaciones, podemos juntarlos para construir **sentencias atómicas** que representan hechos. Una sentencia atómica está compuesta por un símbolo de predicado seguido de una lista de términos entre paréntesis:

$$Hermano(Ricardo, Juan)$$

Esto representa, bajo la interpretación deseada que hemos dado antes, que Ricardo Corazón de León es el hermano del Rey Juan[5]. Las sentencias atómicas pueden tener términos complejos. De este modo,

$$CasadoCon(Padre(Ricardo), Madre(Juan))$$

representa que el padre de Ricardo Corazón de León está casado con la madre del Rey Juan (otra vez, bajo la adecuada interpretación).

*Una sentencia atómica es **verdadera** en un modelo dado, y bajo una interpretación dada, si la relación referenciada por el símbolo de predicado sucede entre los objetos referenciados por los argumentos.*

Sentencias compuestas

Podemos utilizar las **conectivas lógicas** para construir sentencias más complejas, igual que en la lógica proposicional. La semántica de las sentencias formadas con las conectivas lógicas es idéntica a la de la lógica proposicional. Aquí hay cuatro sentencias que son verdaderas en el modelo de la Figura 8.2, bajo la interpretación deseada:

$$\neg Hermano(PiernaIzquierda(Ricardo), Juan)$$
$$Hermano(Ricardo, Juan) \wedge Hermano(Juan, Ricardo)$$
$$Rey(Ricardo) \vee Rey(Juan)$$
$$\neg Rey(Ricardo) \Rightarrow Rey(Juan)$$

Cuantificadores

Una vez tenemos una lógica que nos permite representar objetos, es muy natural querer expresar las propiedades de colecciones enteras de objetos en vez de enumerar los objetos por su nombre. Los **cuantificadores** nos permiten hacer esto. La lógica de primer orden contiene dos cuantificadores estándar, denominados *universal* y *existencial*.

CUANTIFICADORES

[5] Por lo general utilizaremos la convención de ordenación de los argumentos $P(x, y)$, que se interpreta como «x es P de y».

Cuantificador universal (∀)

Retomemos la dificultad que teníamos en el Capítulo 7 con la expresión de las reglas generales en la lógica proposicional. Las reglas como «Las casillas vecinas al *wumpus* son apestosas» y «Todos los reyes son personas» son el pan de cada día de la lógica de primer orden. En la Sección 8.3 trataremos con la primera de éstas. Respecto a la segunda regla, «Todos los reyes son personas», se escribe en la lógica de primer orden

$$\forall x \ \ Rey(x) \ \Rightarrow \ Persona(x)$$

generalmente ∀ se pronuncia «Para todo…». (Recuerda que la A boca abajo representa «todo».) Así, la sentencia dice, «Para todo x, si x es un rey, entonces x es una persona». Al símbolo x se le llama **variable**. Por convenio, las variables se escriben en minúsculas. Una variable es un término en sí mismo, y como tal, también puede utilizarse como el argumento de una función, por ejemplo, $PiernaIzquierda(x)$. Un término que no tiene variables se denomina **término base**.

VARIABLE

TÉRMINO BASE

De forma intuitiva, la sentencia $\forall x \ P$, donde P es una expresión lógica, dice que P es verdadera para cada objeto x. Siendo más precisos, $\forall x \ P$ es verdadera en un modelo dado bajo una interpretación dada, si P es verdadera para todas las **interpretaciones ampliadas**, donde cada interpretación ampliada especifica un elemento del dominio al que se refiere x.

INTERPRETACIÓN AMPLIADA

Esto suena algo complicado, pero tan sólo es una manera cautelosa de definir el sentido intuitivo de la cuantificación universal. Considere el modelo que se muestra en la Figura 8.2 y la interpretación deseada que va con él. Podemos ampliar la interpretación de cinco maneras:

$x \rightarrow$ Ricardo Corazón de León,
$x \rightarrow$ Rey Juan,
$x \rightarrow$ pierna izquierda de Ricardo,
$x \rightarrow$ pierna izquierda de Juan,
$x \rightarrow$ la corona.

La sentencia cuantificada universalmente $\forall x \ \ Rey(x) \Rightarrow Persona(x)$ es verdadera bajo la interpretación inicial si la sentencia $Rey(x) \Rightarrow Persona(x)$ es verdadera en cada una de las interpretaciones ampliadas. Es decir, la sentencia cuantificada universalmente es equivalente a afirmar las cinco sentencias siguientes:

Ricardo Corazón de León es un rey ⇒ Ricardo Corazón de León es una persona.
Rey Juan es un rey ⇒ Rey Juan es una persona.
La pierna izquierda de Ricardo es un rey ⇒ La pierna izquierda de Ricardo es una persona.
La pierna izquierda de Juan es un rey ⇒ La pierna izquierda de Juan es una persona.
La corona es un rey ⇒ La corona es una persona.

Vamos a observar cuidadosamente este conjunto de aserciones. Ya que en nuestro modelo el Rey Juan es el único rey, la segunda sentencia aserta que él es una persona, tal como esperamos. Pero, ¿qué ocurre con las otras cuatro sentencias, donde parece que incluso se reivindica acerca de piernas y coronas? ¿Eso forma parte del sentido que tie-

ne «Todos los reyes son personas»? De hecho, las otras cuatro aserciones son verdaderas en el modelo, pero no hacen en absoluto ninguna reivindicación acerca de la naturaleza de persona de las piernas, coronas, o en efecto de Ricardo. Esto es porque ninguno de estos objetos es un rey. Mirando la tabla de verdad de la conectiva $\Rightarrow$ (Figura 7.8) vemos que la implicación es verdadera siempre que su premisa sea falsa (*independientemente* del valor de verdad de la conclusión). Así que, al afirmar una sentencia cuantificada universalmente, que es equivalente a afirmar la lista total de implicaciones individuales, acabamos afirmando la conclusión de la regla sólo para aquellos objetos para los que la premisa es verdadera y no decimos nada acerca de aquellos individuos para los que la premisa es falsa. De este modo, las entradas para la tabla de verdad de la conectiva $\Rightarrow$ son perfectas para escribir reglas generales mediante cuantificadores universales.

Un error común, hecho frecuentemente aún por los lectores más diligentes que han leído este párrafo varias veces, es utilizar la conjunción en vez de la implicación. La sentencia

$$\forall x \;\; Rey(x) \wedge Persona(x)$$

sería equivalente a afirmar

Ricardo Corazón de León es un rey $\wedge$ Ricardo Corazón de León es una persona
Rey Juan es un rey $\wedge$ Rey Juan es una persona
La pierna izquierda de Ricardo es un rey $\wedge$ La pierna izquierda de Ricardo es una persona

etcétera. Obviamente, esto no plasma lo que queremos expresar.

Cuantificación existencial ($\exists$)

La cuantificación universal construye enunciados acerca de todos los objetos. De forma similar, utilizando un cuantificador existencial, podemos construir enunciados acerca de *algún* objeto del universo de discurso sin nombrarlo. Para decir, por ejemplo, que el Rey Juan tiene una corona sobre su cabeza, escribimos

$$\exists x \; Corona(x) \wedge SobreCabeza(x, Juan)$$

$\exists x$ se pronuncia «Existe un x tal que…» o «Para algún x…».

De forma intuitiva, la sentencia $\exists x \; P$ dice que P es verdadera al menos para un objeto x. Siendo más precisos, $\exists x \; P$ es verdadera en un modelo dado bajo una interpretación dada si P es verdadera *al menos en una* interpretación ampliada que asigna a x un elemento del dominio. Para nuestro ejemplo, esto significa que al menos una de las sentencias siguientes debe ser verdadera:

Ricardo Corazón de León es una corona $\wedge$ Ricardo Corazón de León está sobre la cabeza de Juan;
Rey Juan es una corona $\wedge$ Rey Juan está sobre la cabeza de Juan;
La pierna izquierda de Ricardo es una corona $\wedge$ La pierna izquierda de Ricardo está sobre la cabeza de Juan;

La pierna izquierda de Juan es una corona $\wedge$ La pierna izquierda de Juan está sobre la cabeza de Juan;
La corona es una corona $\wedge$ La corona está sobre la cabeza de Juan.

La quinta aserción es verdadera en nuestro modelo, por lo que la sentencia original cuantificada existencialmente es verdadera en el modelo. Fíjese en que, según nuestra definición, la sentencia también sería verdadera en un modelo en el que el Rey Juan llevara dos coronas. Esto es totalmente consistente con la sentencia inicial «El Rey Juan tiene una corona sobre su cabeza»[6].

Igual que el utilizar con el cuantificador $\forall$ la conectiva $\Rightarrow$ parece ser lo natural, $\wedge$ es la conectiva natural para ser utilizada con el cuantificador $\exists$. Utilizar $\wedge$ como la conectiva principal con $\forall$ nos llevó a un enunciado demasiado fuerte en el ejemplo de la sección anterior; y en efecto, utilizar $\Rightarrow$ con $\exists$ nos lleva a un enunciado demasiado débil. Considere la siguiente sentencia:

$$\exists x \; Corona(x) \Rightarrow SobreCabeza(x, Juan)$$

Superficialmente, esto podría parecer una interpretación razonable de nuestra sentencia. Al aplicar la semántica vemos que la sentencia dice que al menos una de las aserciones siguientes es verdadera:

Ricardo Corazón de León es una corona $\Rightarrow$ Ricardo Corazón de León está sobre la cabeza de Juan;
Rey Juan es una corona $\Rightarrow$ Rey Juan está sobre la cabeza de Juan;
La pierna izquierda de Ricardo es una corona $\Rightarrow$ La pierna izquierda de Ricardo está sobre la cabeza de Juan;

etcétera. Ahora una implicación es verdadera si son verdaderas la premisa y la conclusión, *o si su premisa es falsa*. Entonces, si Ricardo Corazón de León no es una corona, entonces la primera aserción es verdadera y se satisface el existencial. Así que, una implicación cuantificada existencialmente es verdadera en cualquier modelo que contenga un objeto para el que la premisa de la implicación sea falsa; de aquí que este tipo de sentencias al fin y al cabo no digan mucho.

Cuantificadores anidados

A menudo queremos expresar sentencias más complejas utilizando múltiples cuantificadores. El caso más sencillo es donde los cuantificadores son del mismo tipo. Por ejemplo, «Los camaradas son hermanos» se puede escribir como

$$\forall x \;\; \forall y \;\; Hermano(x, y) \Rightarrow Camarada(x, y)$$

[6] Hay una variante del cuantificador existencial, escrito por lo general $\exists^1$ o $\exists!$, que significa «Existe exactamente uno.» El mismo significado se puede expresar utilizando sentencias de igualdad, tal como mostraremos en esta misma sección.

Los cuantificadores consecutivos del mismo tipo se pueden escribir como un solo cuantificador con sendas variables. Por ejemplo, para decir que la relación de hermandad es una relación simétrica podemos escribir

$$\forall x, y \;\; Camarada(x, y) \Rightarrow Camarada(y, x)$$

En otros casos tenemos combinaciones. «Todo el mundo ama a alguien» significa que para todas las personas, hay alguien que esa persona ama:

$$\forall x \;\; \exists y \;\; Ama(x, y)$$

Por otro lado, para decir «Hay alguien que es amado por todos», escribimos

$$\exists y \;\; \forall x \;\; Ama(x, y)$$

Por lo tanto, el orden de los cuantificadores es muy importante. Está más claro si introducimos paréntesis. $\forall x \; (\exists y \; Ama(x, y))$ dice que *todo el mundo* tiene una propiedad en particular, en concreto, la propiedad de amar a alguien. Por otro lado, $\exists y \; (\forall y \; Ama(x, y))$ dice que *alguien* en el mundo tiene una propiedad particular, en concreto, la propiedad de ser amado por todos.

Puede aparecer alguna confusión cuando dos cuantificadores se utilizan con el mismo identificador de variable. Considere la sentencia

$$\forall x \; [Corona(x) \vee (\exists x \; Hermano(Ricardo, x))]$$

Aquí la x de $Hermano(Ricardo, x)$ está cuantificada existencialmente. La regla es que la variable pertenece al cuantificador más anidado que la mencione; entonces no será el sujeto de cualquier otro cuantificador[7]. Otra forma de pensar en esto es: $\exists x \; Hermano(Ricardo, x)$ es una sentencia acerca de Ricardo (que él tiene un hermano), no acerca de x; así que poner $\forall x$ fuera no tiene ningún efecto. Se podría perfectamente haber escrito $\exists z \; Hermano(Ricardo, z)$. Y como esto puede ser una fuente de confusión, siempre utilizaremos variables diferentes.

Conexiones entre ∀ y ∃

Los dos cuantificadores realmente están íntimamente conectados el uno al otro, mediante la negación. Afirmar que a todo el mundo no le gustan las pastinacas es lo mismo que afirmar que no existe alguien a quien le gusten, y viceversa:

$$\forall x \;\; \neg Gusta(x, Pastinacas) \quad \text{es equivalente a} \quad \neg\exists x \;\; Gusta(x, Pastinacas).$$

[7] Es el potencial para la inferencia entre cuantificadores que utilizan el mismo identificador de variable lo que motiva el mecanismo barroco de las interpretaciones ampliadas en la semántica de las sentencias cuantificadas. El enfoque intuitivo más obvio de sustituir los objetos de cada ocurrencia de x falla en nuestro ejemplo porque la x de $Hermano(Ricardo, x)$ sería «capturada» por la sustitución. Las interpretaciones ampliadas manejan este tema de forma correcta porque la asignación para x del cuantificador más interiorizado estropea a los cuantificadores externos.

Podemos dar un paso más allá: «A todo el mundo le gusta el helado» significa que no hay nadie a quien no le guste el helado:

$$\forall x \;\; Gusta(x, Helado) \quad \text{es equivalente a} \quad \neg\exists x \;\; \neg Gusta(x, Helado).$$

Como $\forall$ realmente es una conjunción sobre el universo de objetos y $\exists$ es una disyunción, no sería sorprendente que obedezcan a las leyes de Morgan. Las leyes de Morgan para las sentencias cuantificadas y no cuantificadas son las siguientes:

$$\begin{array}{lll} \forall x \;\; \neg P \equiv \neg\exists x \;\; P & \qquad & \neg P \wedge \neg Q \equiv \neg(P \vee Q) \\ \neg\forall x \;\; P \equiv \exists x \;\; \neg P & & \neg(P \wedge Q) \equiv \neg P \vee \neg Q \\ \forall x \;\; P \equiv \neg\exists x \;\; \neg P & & P \wedge Q \equiv \neg(\neg P \vee \neg Q) \\ \exists x \;\; P \equiv \neg\forall x \;\; \neg P & & P \vee Q \equiv \neg(\neg P \wedge \neg Q) \end{array}$$

De este modo, realmente no necesitamos $\forall$ y $\exists$ al mismo tiempo, igual que no necesitamos $\wedge$ y $\vee$ al mismo tiempo. Todavía es más importante la legibilidad que la parquedad, así que seguiremos utilizando ambos cuantificadores.

Igualdad

SÍMBOLO DE IGUALDAD

La lógica de primer orden incluye un mecanismo extra para construir sentencias atómicas, uno que no utiliza un predicado y unos términos como hemos descrito antes. En lugar de ello, podemos utilizar el **símbolo de igualdad** para construir enunciados describiendo que dos términos se refieren al mismo objeto. Por ejemplo,

$$Padre(Juan) = Enrique$$

dice que el objeto referenciado por *Padre*(*Juan*) y el objeto referenciado por *Enrique* son el mismo. Como una interpretación especifica el referente para cualquier término, determinar el valor de verdad de una sentencia de igualdad consiste simplemente en ver que los referentes de los dos términos son el mismo objeto.

El símbolo de igualdad se puede utilizar para representar hechos acerca de una función dada, tal como hicimos con el símbolo *Padre*, también se puede utilizar con la negación para insistir en que dos términos no son el mismo objeto. Para decir que Ricardo tiene al menos dos hermanos escribiríamos

$$\exists x, y \; Hermano(x, Ricardo) \wedge Hermano(y, Ricardo) \wedge \neg(x = y)$$

La sentencia

$$\exists x, y \; Hermano(x, Ricardo) \wedge Hermano(y, Ricardo)$$

no tiene el significado deseado. En concreto, es verdadero en el modelo de la Figura 8.2, donde Ricardo tiene sólo un hermano. Para verlo, considere las interpretaciones ampliadas en las que x e y son asignadas al Rey Juan. La adición de $\neg(x = y)$ excluye dichos modelos. La notación $x \neq y$ se utiliza a veces como abreviación de $\neg(x = y)$.

8.3 Utilizar la lógica de primer orden

DOMINIOS

Ahora que hemos definido un lenguaje lógico expresivo, es hora de aprender a utilizarlo. La mejor forma de hacerlo es a través de ejemplos. Hemos visto algunas sentencias sencillas para mostrar los diversos aspectos de la sintaxis lógica; en esta sección proporcionaremos unas representaciones más sistemáticas de algunos **dominios** sencillos. En la representación del conocimiento un dominio es sólo algún ámbito del mundo acerca del cual deseamos expresar algún conocimiento.

Comenzaremos con una breve descripción de la interfaz DECIR/PREGUNTAR para las bases de conocimiento en primer orden. Entonces veremos los dominios de las relaciones de parentesco, de los números, de los conjuntos, de las listas y del mundo de *wumpus*. La siguiente sección contiene un ejemplo mucho más sustancial (sobre circuitos electrónicos) y en el Capítulo 10 cubriremos cada aspecto del universo de discurso.

Aserciones y peticiones en lógica de primer orden

AFIRMACIONES

Las sentencias se van añadiendo a la base de conocimiento mediante DECIR, igual que en la lógica proposicional. Este tipo de sentencias se denominan **aserciones**. Por ejemplo, podemos afirmar que Juan es un rey y que los reyes son personas mediante las siguientes sentencias:

$$\text{DECIR}(BC,\ Rey(Juan))$$
$$\text{DECIR}(BC,\ \forall x\ \ Rey(x) \Rightarrow Persona(x))$$

Podemos hacer preguntas a la base de conocimiento mediante PREGUNTAR. Por ejemplo,

$$\text{PREGUNTAR}(BC, Rey(Juan))$$

PETICIONES

OBJETIVOS

que devuelve *verdadero*. Las preguntas realizadas con PREGUNTAR se denominan **peticiones** u **objetivos** (no deben confundirse con los objetivos que se utilizan para describir los estados deseados por el agente). En general, cualquier petición que se implica lógicamente de la base de conocimiento sería respondida afirmativamente. Por ejemplo, dadas las dos aserciones en el párrafo precedente, la petición

$$\text{PREGUNTAR}(BC, Persona(Juan))$$

también devolvería *verdadero*. También podemos realizar peticiones cuantificadas, tales como

$$\text{PREGUNTAR}(BC, \exists x\ Persona(x)).$$

SUSTITUCIÓN

LISTA DE LIGADURAS

La respuesta a esta petición podría ser *verdadero*, pero esto no es de ayuda ni es ameno. (Es como responder a «¿Me puedes decir qué hora es?» con un «Sí».) Una petición con variables existenciales es como preguntar «Hay algún x tal qué…» y lo resolvemos proporcionando dicha x. La forma estándar para una respuesta de este tipo es una **sustitución** o **lista de ligaduras**, que es un conjunto de parejas de variable/término. En este

caso en particular, dadas las dos aserciones, la respuesta sería $\{x/Juan\}$. Si hay más de una respuesta posible se puede devolver una lista de sustituciones.

El dominio del parentesco

El primer ejemplo que vamos a tratar es el dominio de las relaciones familiares, o de parentesco. Este dominio incluye hechos como «Isabel es la madre de Carlos» y «Carlos es el padre de Guillermo», y reglas como «La abuela de uno es la madre de su padre».

Está claro que los objetos de nuestro dominio son personas. Tendremos dos predicados unitarios: *Masculino* y *Femenino*. Las relaciones de parentesco (de paternidad, de hermandad, de matrimonio, etcétera) se representarán mediante los predicados binarios: *Padre, Hermano Político, Hermano, Hermana, Niño, Hija, Hijo, Esposo, Mujer, Marido, Abuelo, Nieto, Primo, Tía,* y *Tío*. Utilizaremos funciones para *Madre* y *Padre*, porque todas las personas tienen exactamente uno de cada uno (al menos de acuerdo con las reglas de la naturaleza).

Podemos pasar por cada función y predicado, apuntando lo que sabemos en términos de los otros símbolos. Por ejemplo, la madre de uno es uno de los padres y es femenino:

$$\forall x, y \;\; Madre(y) = x \Leftrightarrow Femenino(x) \wedge Padre(x, y).$$

El marido de uno es un esposo masculino:

$$\forall x, y \;\; Marido(y, x) \Leftrightarrow Masculino(y) \wedge Esposo(y, x).$$

Masculino y Femenino son categorías disjuntas:

$$\forall x \;\; Masculino(x) \Leftrightarrow \neg Femenino(x).$$

Padre e hijo son relaciones inversas:

$$\forall x, y \;\; Padre(x, y) \Leftrightarrow Hijo(y, x).$$

Un abuelo es el padre del padre de uno:

$$\forall x, y \;\; Abuelo(x, y) \Leftrightarrow \exists z \;\; Padre(x, z) \wedge Padre(z, y).$$

Un hermano es otro hijo del padre de uno:

$$\forall x, y \;\; Hermano(x, y) \Leftrightarrow x \neq y \wedge \exists z \;\; Padre(z, x) \wedge Padre(z, y).$$

Podríamos seguir con más páginas como esta, y el Ejercicio 8.11 pide que haga justamente eso.

AXIOMAS

Cada una de estas sentencias se puede ver como un **axioma** del dominio del parentesco. Los axiomas se asocian por lo general con dominios puramente matemáticos (veremos algunos axiomas sobre números en breve) pero la verdad es que se necesitan en todos los dominios. Los axiomas proporcionan la información factual esencial de la cual se pueden derivar conclusiones útiles. Nuestros axiomas de parentesco también son

DEFINICIONES

definiciones; tienen la forma $\forall x, y\ P(x, y) \Leftrightarrow \ldots$ Los axiomas definen la función *Madre* y los predicados *Marido*, *Masculino*, *Padre*, *Abuelo* y *Hermano* en términos de otros predicados. Nuestras definiciones «tocan el fondo» de un conjunto básico de predicados (*Hijo, Esposo,* y *Femenino*) sobre los cuales se definen los demás. Esta es una forma muy natural de desarrollar la representación de un dominio, y es análogo a la forma en que los paquetes de *software* se desarrollan a partir de definiciones sucesivas de subrutinas, partiendo de una biblioteca de funciones primitivas. Fíjese en que no hay necesariamente un único conjunto de predicados primitivos; podríamos perfectamente haber utilizado *Padre*, *Esposo* y *Masculino*. En algunos dominios, tal como veremos, no hay un conjunto básico claramente identificable.

TEOREMAS

No todas las sentencias lógicas acerca de un dominio son axiomas. Algunas son **teoremas**, es decir, son deducidas a partir de los axiomas. Por ejemplo, considere la aserción acerca de que la relación de hermandad es simétrica:

$$\forall x, y\ Hermano(x, y) \Leftrightarrow Hermano(y, x).$$

¿Es un axioma o un teorema? De hecho, es un teorema que lógicamente se sigue de los axiomas definidos para la relación de hermandad. Si PREGUNTAMOS a la base de conocimiento sobre esta sentencia, la base devolvería *verdadero*.

Desde un punto de vista puramente lógico, una base de conocimiento sólo necesita contener axiomas y no necesita contener teoremas, porque los teoremas no aumentan el conjunto de conclusiones que se siguen de la base de conocimiento. Desde un punto de vista práctico, los teoremas son esenciales para reducir el coste computacional para derivar sentencias nuevas. Sin ellos, un sistema de razonamiento tiene que empezar desde el principio cada vez, como si un físico tuviera que volver a deducir las reglas del cálculo con cada problema nuevo.

No todos los axiomas son definiciones. Algunos proporcionan información más general acerca de ciertos predicados sin tener que constituir una definición. Por el contrario, algunos predicados no tienen una definición completa porque no sabemos lo suficiente para caracterizarlos totalmente. Por ejemplo, no hay una manera obvia para completar la sentencia:

$$\forall x\ Persona(x) \Leftrightarrow \ldots$$

Afortunadamente, la lógica de primer orden nos permite hacer uso del predicado *Persona* sin definirlo completamente. En lugar de ello, podemos escribir especificaciones parciales de las propiedades que cada persona tiene y de las propiedades que hacen que algo sea una persona:

$$\forall x\ \ Persona(x) \Leftrightarrow \ldots$$
$$\forall x\ \ \ldots \Leftrightarrow Persona(x)$$

Los axiomas también pueden ser «tan sólo puros hechos», tal como *Masculino*(*Jaime*) y *Esposo*(*Jaime, Laura*). Este tipo de hechos forman las descripciones de las instancias de los problemas concretos, permitiendo así que se responda a preguntas concretas. Entonces, las respuestas a estas preguntas serán los teoremas que se siguen de los axiomas. A menudo, nos encontramos con que las respuestas esperadas no están disponibles, por ejemplo, de *Masculino*(*Jorge*) y *Esposo*(*Jorge, Laura*) esperamos ser capaces de in-

ferir *Femenino*(*Laura*); pero esta sentencia no se sigue de los axiomas dados anteriormente. Y esto es una señal de que nos hemos olvidado de algún axioma.

Números, conjuntos y listas

NÚMEROS NATURALES

AXIOMAS DE PEANO

Los números son quizás el ejemplo más gráfico de cómo se puede construir una gran teoría a partir de un núcleo de axiomas diminuto. Aquí describiremos la teoría de los **números naturales**, o la de los enteros no negativos. Necesitamos un predicado *NumNat* que será verdadero para los números naturales; necesitamos un símbolo de constante, 0; y necesitamos un símbolo de función, S (sucesor). Los **axiomas de Peano** definen los números naturales y la suma[8]. Los números naturales se definen recursivamente:

$$NumNat(0)$$
$$\forall n\ NumNat(n) \Rightarrow NumNat(S(n))$$

Es decir, 0 es un número natural, y para cada objeto n, si n es un número natural entonces el $S(n)$ es un número natural. Así, los números naturales son el 0, el $S(0)$, el $S(S(0))$, etcétera. También necesitamos un axioma para restringir la función sucesor:

$$\forall n\ 0 \neq S(n)$$
$$\forall m, n\ m \neq n \Rightarrow S(m)\ m\ S(n)$$

Ahora podemos definir la adición (suma) en términos de la función sucesor:

$$\forall m\ NumNat(m) \Rightarrow +(m, 0) = m$$
$$\forall m, n\ NumNat(m) \wedge NumNat(n) \Rightarrow +(S(m), n) = S(+(m, n))$$

INFIJA

PREFIJA

El primero de estos axiomas dice que sumar 0 a cualquier número natural m da el mismo m. Fíjese en el uso de la función binaria «+» en el término $+(m, 0)$; en las matemáticas habituales el término estaría escrito $m + 0$, utilizando la notación **infija**. (La notación que hemos utilizado para la lógica de primer orden se denomina **prefija**.) Para hacer que nuestras sentencias acerca de los números sean más fáciles de leer permitiremos el uso de la notación infija. También podemos escribir $S(n)$ como $n + 1$, entonces el segundo axioma se convierte en

$$\forall m, n\ NumNat(m) \wedge NumNat(n) \Rightarrow (m + 1) + n = (m + n) + 1$$

Este axioma reduce la suma a la aplicación repetida de la función sucesor.

SINTAXIS EDULCORADA

El uso de la notación infija es un ejemplo de **sintaxis edulcorada**, es decir, una ampliación o abreviación de una sintaxis estándar que no cambia su semántica. Cualquier sentencia que está edulcorada puede «des-edulcorarse» para producir una sentencia equivalente en la habitual lógica de primer orden.

Una vez tenemos la suma, es fácil definir la multiplicación como una suma repetida, la exponenciación como una multiplicación repetida, la división entera y el resto, los

[8] Los axiomas de Peano también incluyen el principio de inducción, que es una sentencia de lógica segundo orden más que de lógica de primer orden. La importancia de esta diferencia se explica en el Capítulo 9.

números primos, etcétera. De este modo, la totalidad de la teoría de los números (incluyendo la criptografía) se puede desarrollar a partir de una constante, una función, un predicado y cuatro axiomas.

CONJUNTOS

El dominio de los **conjuntos** es tan fundamental para las matemáticas como para el razonamiento del sentido común. (De hecho, es posible desarrollar la teoría de los números con base en la teoría de los conjuntos.) Queremos ser capaces de representar conjuntos individuales, incluyendo el conjunto vacío. Necesitamos un mecanismo para construir conjuntos añadiendo un elemento a un conjunto o tomando la unión o la intersección de dos conjuntos. Querremos saber si un elemento es un miembro de un conjunto, y ser capaces de distinguir conjuntos de objetos que no son conjuntos.

Utilizaremos el vocabulario habitual de la teoría de conjuntos como sintaxis edulcorada. El conjunto vacío es una constante escrita como $\{\}$. Hay un predicado unitario, *Conjunto*, que es verdadero para los conjuntos. Los predicados binarios son $x \in s$ (x es un miembro del conjunto s) y $s_1 \subseteq s_2$ (el conjunto s_1 es un subconjunto, no necesariamente propio, del conjunto s_2). Las funciones binarias son $s_1 \cap s_2$ (la intersección de dos conjuntos), $s_1 \cup s_2$ (la unión de dos conjuntos), y $\{x|s\}$ (el conjunto resultante de añadir el elemento x al conjunto s). Un conjunto posible de axiomas es el siguiente:

1. Los únicos conjuntos son el conjunto vacío y aquellos construidos añadiendo algo a un conjunto:
$$\forall s \;\; Conjunto(s) \Leftrightarrow (s = \{\}) \vee (\exists x, s_2 \;\; Conjunto(s_2) \wedge s = \{x|s_2\})$$
2. El conjunto vacío no tiene elementos añadidos a él, en otras palabras, no hay forma de descomponer un *ConjuntoVacío* en un conjunto más pequeño y un elemento:
$$\neg\exists x, s \;\; \{x|s\} = \{\}$$
3. Añadir un elemento que ya pertenece a un conjunto no tiene ningún efecto:
$$\forall x, s \;\; x \in s \Leftrightarrow s = \{x|s\}$$
4. Los únicos elementos de un conjunto son los que fueron añadidos a él. Expresamos esto recursivamente, diciendo que x es un miembro de s si y sólo si s es igual a algún conjunto s_2 al que se le ha añadido un elemento y, y que y era el mismo elemento que x o que x es un miembro de s_2:
$$\forall x, s \;\; x \in s \Leftrightarrow [\exists y, s_2 \;\; (s = \{y|s_2\} \wedge (x = y \vee x \in s_2))]$$
5. Un conjunto es un subconjunto de otro conjunto si y sólo si todos los miembros del primer conjunto son miembros del segundo conjunto:
$$\forall s_1, s_2 \;\; s_1 \subseteq s_2 \Leftrightarrow (\forall x \;\; x \in s_1 \Rightarrow x \in s_2)$$
6. Dos conjuntos son iguales si y sólo si cada uno es subconjunto del otro:
$$\forall s_1, s_2 \;\; (s_1 = s_2) \Leftrightarrow (s_1 \subseteq s_2 \wedge s_2 \subseteq s_1)$$
7. Un objeto pertenece a la intersección de dos conjuntos si y sólo si es miembro de ambos conjuntos:
$$\forall x, s_1, s_2 \;\; x \in (s_1 \cap s_2) \Leftrightarrow (x \in s_1 \wedge x \in s_2)$$

8. Un objeto pertenece a la unión de dos conjuntos si y sólo si es miembro de alguno de los dos:

$$\forall x, s_1, s_2 \; x \in (s_1 \cup s_2) \Leftrightarrow (x \in s_1 \vee x \in s_2)$$

LISTAS

Las **listas** son muy parecidas a los conjuntos. Las diferencias son que las listas están ordenadas y que el mismo elemento puede aparecer más de una vez en una lista. Podemos utilizar el vocabulario del Lisp para las listas: *Nil* es la constante para las listas sin elementos; *Cons, Unir, Primero* y *Resto* son funciones; y *Encontrar* es el predicado que hace en listas lo que *Miembro* hace en conjuntos. *¿Lista?* es un predicado que es verdadero sólo para las listas. Como en los conjuntos, es común el uso de sintaxis edulcoradas en las sentencias lógicas que tratan sobre listas. La lista vacía es [ ]. El término $Cons(x, y)$, donde y es un conjunto no vacío, se escribe $[x|y]$. El término $Cons(x, Nil)$, (por ejemplo, la lista conteniendo el elemento x), se escribe $[x]$. Una lista con varios elementos, como $[A, B, C]$, se corresponde al término anidado $Cons(A, Cons(B, Cons(C, Nil)))$. El Ejercicio 8.14 pide que escriba los axiomas para las listas.

El mundo de *wumpus*

En el Capítulo 7 se dieron algunos axiomas para el mundo de *wumpus* en lógica proposicional. Los axiomas en lógica de primer orden de esta sección son mucho más precisos, capturando de forma natural exactamente lo que queremos expresar.

Recuerde que el agente *wumpus* recibe un vector de percepciones con cinco elementos. La sentencia en primer orden correspondiente almacenada en la base de conocimiento debe incluir tanto la percepción como el instante de tiempo en el que ocurrió ésta, de otra manera el agente se confundiría acerca de cuándo vio qué cosa. Utilizaremos enteros para los instantes de tiempo. Una típica sentencia de percepción sería

$$Percepción([MalHedor, Brisa, Resplandor, Nada, Nada], 5)$$

Aquí, *Percepción* es un predicado binario y *MalHedor* y otros son constantes colocadas en la lista. Las acciones en el mundo de *wumpus* se pueden representar mediante términos lógicos:

$$Girar(Derecha), Girar(Izquierda), Avanzar, Disparar, Agarrar, Libertar, Escalar$$

Para hallar qué acción es la mejor, el programa del agente construye una petición como esta

$$\exists a \; MejorAcción(a, 5)$$

PREGUNTAR resolvería esta petición y devolvería una lista de ligaduras como $\{a/Agarrar\}$. El programa del agente entonces puede devolver *Agarrar* como la acción que debe llevar a cabo, pero primero debe DECIR a la propia base de conocimiento que está ejecutando la acción *Agarrar*.

Los datos acerca de las crudas percepciones implican ciertos hechos acerca del estado actual. Por ejemplo:

$$\forall t, s, g, m, c \; Percepción([s, Brisa, g, m, c], t) \Rightarrow Brisa(t),$$
$$\forall t, s, b, m, c \; Percepción([s, b, Resplandor, m, c], t) \Rightarrow Resplandor(t),$$

etcétera. Estas reglas muestran una forma trivial del proceso de razonamiento denominado **percepción**, que estudiaremos en profundidad en el Capítulo 24. Fíjese en la cuantificación sobre t. En la lógica proposicional, habríamos necesitado copias de cada sentencia para cada instante de tiempo.

El comportamiento simple de tipo «reflexivo» también puede ser implementado mediante sentencias de implicación cuantificada. Por ejemplo, tenemos

$$\forall t \; Resplandor(t) \Rightarrow MejorAcción(Agarrar, t)$$

Dadas las percepciones y las reglas de los párrafos precedentes, esto nos daría la conclusión deseada $MejorAcción(Agarrar, 5)$ (es decir, Agarrar es lo más correcto a hacer). Fíjese en la correspondencia entre esta regla y la conexión directa percepción-acción de los agentes basados en circuitos de la Figura 7.20; la conexión en el circuito se cuantifica *implícitamente* sobre el tiempo.

SINCRÓNICA

DIACRÓNICA

Hasta ahora en esta sección, las sentencias que tratan con el tiempo han sido sentencias **sincrónicas** («al mismo tiempo»), es decir, las sentencias relacionan propiedades del estado del mundo con otras propiedades del mismo estado del mundo. Las sentencias que permiten razonar «a través del tiempo» se denominan **diacrónicas**; por ejemplo, el agente necesita saber combinar la información acerca de sus localizaciones anteriores con la información acerca de la acción que acaba de realizar, para establecer su localización actual. Aplazaremos la discusión acerca de las sentencias diacrónicas hasta el Capítulo 10; por ahora, sólo asuma que las inferencias necesarias se han realizado para los predicados de localización, y otros, dependientes del tiempo.

Hemos representado las percepciones y las acciones; ahora es el momento de representar el propio entorno. Vamos a empezar con los objetos. Los candidatos obvios son las casillas, los hoyos y el *wumpus*. Podríamos nombrar cada casilla ($Casilla_{1,2}$, etcétera) pero entonces el hecho de que la $Casilla_{1,2}$ y la $Casilla_{1,3}$ estén adyacentes tendría que ser un hecho «extra», y necesitaríamos un hecho de este tipo para cada par de casillas. Es mejor utilizar un término complejo en el que la fila y la columna aparezcan como enteros; por ejemplo, simplemente podemos usar la lista de términos [1, 2]. La adyacencia entre dos casillas se puede definir mediante

$$\forall x, y, a, b, \; Adyacente([x, y], [a, b]) \Leftrightarrow$$
$$[a, b] \in \{[x + 1, y], [x - 1, y], [x, y + 1], [x, y - 1]\}$$

También podríamos nombrar cada hoyo, pero sería inapropiado por otro motivo: no hay ninguna razón para distinguir a unos hoyos de otros[9]. Es mucho más sencillo utilizar un predicado unario *Hoyo* que es verdadero en las casillas que contengan hoyos. Por último, como sólo hay exactamente un *wumpus*, una constante *Wumpus* es tan buena como un predicado unitario (y quizá más digno para el punto de vista del *wumpus*). El *wumpus* vive exactamente en una casilla, por tanto es una buena idea utilizar una función como $Casa(Wumpus)$ para nombrar la casilla. Esto evita por completo el enorme conjunto de

[9] De forma similar, muchos de nosotros no nombramos cada pájaro que vuela sobre nuestras cabezas en sus migraciones a regiones más cálidas en invierno. Un ornitólogo que desea estudiar los patrones de migración, los ratios de supervivencia, etcétera, nombraría cada pájaro por medio de una alarma en su pata, porque cada pájaro debe ser observado.

sentencias que se necesitaban en la lógica proposicional para decir que una casilla en concreto contenía al *wumpus*. (Aún sería mucho peor para la lógica proposicional si hubieran dos *wumpus*.)

La localización del agente cambia con el tiempo, entonces escribiremos $En(Agente, s, t)$ para indicar que el agente se encuentra en la casilla s en el instante t. Dada su localización actual, el agente puede inferir las propiedades de la casilla a partir de las propiedades de su percepción actual. Por ejemplo, si el agente se encuentra en una casilla y percibe una brisa, entonces la casilla tiene una corriente de aire:

$$\forall s, t \;\; En(Agente, s, t) \wedge Brisa(t) \Rightarrow CorrienteAire(s)$$

Es útil saber si una *casilla* tiene una corriente de aire porque sabemos que los hoyos no pueden desplazarse. Fíjese en que *CorrienteAire* no tiene el argumento del tiempo.

Habiendo descubierto qué casillas tienen brisa (o son apestosas) y, muy importante, las que *no* tienen brisa (o *no* son apestosas), el agente puede deducir dónde están los hoyos (y dónde está el *wumpus*). Hay dos tipos de reglas sincrónicas que podrían permitir sacar este tipo de deducciones:

REGLAS DE DIAGNÓSTICO

- **Reglas de diagnóstico:**
  Las reglas de diagnóstico nos llevan de los efectos observados a sus causas ocultas. Para encontrar hoyos, las reglas obvias de diagnóstico dicen que si una casilla tiene una brisa, alguna casilla adyacente debe contener un hoyo, o

  $$\forall s \; CorrienteAire(s) \Rightarrow \exists r \; Adyacente(r, s) \wedge Hoyo(r)$$

  y si una casilla no tiene una brisa, ninguna casilla adyacente contiene un hoyo[10]:

  $$\forall s \; \neg CorrienteAire(s) \Rightarrow \neg \exists r \; Adyacente(r, s) \wedge Hoyo(r)$$

  Combinando estas dos reglas, obtenemos la siguiente sentencia bicondicional

  $$\forall s \; CorrienteAire(s) \Leftrightarrow \exists r \; Adyacente(r, s) \wedge Hoyo(r) \qquad (8.3)$$

REGLAS CAUSALES

- **Reglas causales:**
  Las reglas causales reflejan la dirección que se asume de causalidad en el mundo: algunas propiedades ocultas del mundo causan que se generen ciertas percepciones. Por ejemplo, un hoyo causa que todas sus casillas adyacentes tengan una brisa:

  $$\forall r \; Hoyo(r) \Rightarrow [\forall s \; Adyacente(s, r) \Rightarrow CorrienteAire(s)]$$

  y si todas las casillas adyacentes a una casilla dada no tienen hoyos, la casilla no tiene brisa:

  $$\forall s \; [\forall r \; Adyacente(r, s) \Rightarrow \neg Hoyo(r)] \Rightarrow \neg CorrienteAire(s)$$

[10] Hay una tendencia humana natural en olvidar anotar la información negativa de este tipo. En una conversación esta tendencia es totalmente normal (sería muy extraño decir «Hay dos copas en la mesa *y no hay tres o más*,» aunque pensar que «Hay dos copas en la mesa» sigue siendo verdadero, estrictamente hablando, cuando hay tres o más). Retomaremos este tema en el Capítulo 10.

Con algo de esfuerzo, es posible demostrar que estas dos sentencias juntas son equivalentes lógicamente a la sentencia bicondicional de la Ecuación (8.3). También se puede pensar en la propia bicondicional como en una regla causal, porque describe cómo se genera el valor de verdad de *CorrienteAire* a partir del estado del mundo.

RAZONAMIENTO BASADO EN MODELOS

Los sistemas que razonan con reglas causales se denominan sistemas de **razonamiento basado en modelos**, porque las reglas causales forman un modelo de cómo se comporta el entorno. La diferencia entre el razonamiento basado en modelos y el de diagnóstico es muy importante en muchas áreas de la IA. El diagnóstico médico en concreto ha sido un área de investigación muy activa, en la que los enfoques basados en asociaciones directas entre los síntomas y las enfermedades (un enfoque de diagnóstico) han sido reemplazados gradualmente por enfoques que utilizan un modelo explícito del proceso de la enfermedad y de cómo se manifiesta en los síntomas. Estos temas se presentarán también en el Capítulo 13.

Cualquier tipo de representación que el agente utilice, *si los axiomas describen correcta y completamente la forma en que el mundo se comporta y la forma en que las percepciones se producen, entonces cualquier procedimiento de inferencia lógica completo inferirá la posible descripción del estado del mundo más robusta, dadas las percepciones disponibles*. Así que el diseñador del agente puede concentrarse en obtener el conocimiento acorde, sin preocuparse demasiado acerca del proceso de deducción. Además, hemos visto que la lógica de primer orden puede representar el mundo de *wumpus*, y no de forma menos precisa que la descripción en lenguaje natural dada en el Capítulo 7.

8.4 Ingeniería del conocimiento con lógica de primer orden

La sección anterior ilustraba el uso de la lógica de primer orden para representar el conocimiento de tres dominios sencillos. Esta sección describe el proceso general de construcción de una base de conocimiento (un proceso denominado **ingeniería del conocimiento**). Un ingeniero del conocimiento es alguien que investiga un dominio concreto, aprende qué conceptos son los importantes en ese dominio, y crea una representación formal de los objetos y relaciones del dominio. Ilustraremos el proceso de ingeniería del conocimiento en un dominio de circuitos electrónicos, que imaginamos ya es bastante familiar, para que nos podamos concentrar en los temas representacionales involucrados. El enfoque que tomaremos es adecuado para desarrollar bases de conocimiento de *propósito-específico* cuyo dominio se circunscribe cuidadosamente y cuyo rango de peticiones se conoce de antemano. Las bases de conocimiento de *propósito-general*, cuya intención es que apoyen las peticiones de todo el abanico del conocimiento humano, se discutirán en el Capítulo 10.

INGENIERÍA DEL CONOCIMIENTO