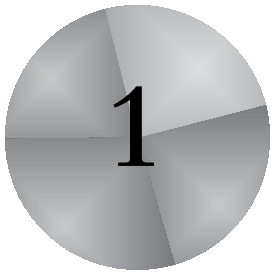

1 Introducción

Donde se intentará explicar por qué se considera a la inteligencia artificial un tema digno de estudio y donde se intentará definirla con exactitud; es esta tarea muy recomendable antes de emprender de lleno su estudio.

INTELIGENCIA ARTIFICIAL

Los hombres se han denominado a sí mismos como *Homo sapiens* (hombre sabio) porque nuestras capacidades mentales son muy importantes para nosotros. Durante miles de años, hemos tratado de entender *cómo pensamos*; es decir, entender cómo un simple puñado de materia puede percibir, entender, predecir y manipular un mundo mucho más grande y complicado que ella misma. El campo de la **inteligencia artificial**, o IA, va más allá: no sólo intenta comprender, sino que también se esfuerza en construir entidades inteligentes.

La IA es una de las ciencias más recientes. El trabajo comenzó poco después de la Segunda Guerra Mundial, y el nombre se acuñó en 1956. La IA se cita, junto a la biología molecular, como un campo en el que a la mayoría de científicos de otras disciplinas «les gustaría trabajar». Un estudiante de ciencias físicas puede pensar razonablemente que todas las buenas ideas han sido ya propuestas por Galileo, Newton, Einstein y otros. Por el contrario, la IA aún tiene flecos sin cerrar en los que podrían trabajar varios Einsteins a tiempo completo.

La IA abarca en la actualidad una gran variedad de subcampos, que van desde áreas de propósito general, como el aprendizaje y la percepción, a otras más específicas como el ajedrez, la demostración de teoremas matemáticos, la escritura de poesía y el diagnóstico de enfermedades. La IA sintetiza y automatiza tareas intelectuales y es, por lo tanto, potencialmente relevante para cualquier ámbito de la actividad intelectual humana. En este sentido, es un campo genuinamente universal.

1.1 ¿Qué es la IA?

Hemos proclamado que la IA es excitante, pero no hemos dicho qué *es*. La Figura 1.1 presenta definiciones de inteligencia artificial extraídas de ocho libros de texto. Las que aparecen en la parte superior se refieren a *procesos mentales* y al *razonamiento*, mientras que las de la parte inferior aluden a la *conducta*. Las definiciones de la izquierda miden el éxito en términos de la fidelidad en la forma de actuar de los *humanos*, mientras que las de la derecha toman como referencia un concepto ideal de inteligencia, que llamaremos **racionalidad**. Un sistema es racional si hace «lo correcto», en función de su conocimiento.

RACIONALIDAD

A lo largo de la historia se han seguido los cuatro enfoques mencionados. Como es de esperar, existe un enfrentamiento entre los enfoques centrados en los humanos y los centrados en torno a la racionalidad[1]. El enfoque centrado en el comportamiento humano debe ser una ciencia empírica, que incluya hipótesis y confirmaciones mediante experimentos. El enfoque racional implica una combinación de matemáticas e ingeniería. Cada grupo al mismo tiempo ha ignorado y ha ayudado al otro. A continuación revisaremos cada uno de los cuatro enfoques con más detalle.

Sistemas que piensan como humanos	**Sistemas que piensan racionalmente**
«El nuevo y excitante esfuerzo de hacer que los computadores piensen… máquinas con mentes, en el más amplio sentido literal». (Haugeland, 1985)	«El estudio de las facultades mentales mediante el uso de modelos computacionales». (Charniak y McDermott, 1985)
«[La automatización de] actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje…» (Bellman, 1978)	«El estudio de los cálculos que hacen posible percibir, razonar y actuar». (Winston, 1992)
Sistemas que actúan como humanos	**Sistemas que actúan racionalmente**
«El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia». (Kurzweil, 1990)	«La Inteligencia Computacional es el estudio del diseño de agentes inteligentes». (Poole *et al*., 1998)
«El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor». (Rich y Knight, 1991)	«IA… está relacionada con conductas inteligentes en artefactos». (Nilsson, 1998)

Figura 1.1 Algunas definiciones de inteligencia artificial, organizadas en cuatro categorías.

[1] Conviene aclarar, que al distinguir entre comportamiento *humano* y *racional* no se está sugiriendo que los humanos son necesariamente «irracionales» en el sentido de «inestabilidad emocional» o «desequilibrio mental». Basta con darnos cuenta de que no somos perfectos: no todos somos maestros de ajedrez, incluso aquellos que conocemos todas las reglas del ajedrez; y desafortunadamente, no todos obtenemos un sobresaliente en un examen. Kahneman *et al*. (1982) ha elaborado un catálogo con algunos de los errores que sistemáticamente cometen los humanos cuando razonan.

Comportamiento humano: el enfoque de la Prueba de Turing

PRUEBA DE TURING

La **Prueba de Turing**, propuesta por Alan Turing (1950), se diseñó para proporcionar una definición operacional y satisfactoria de inteligencia. En vez de proporcionar una lista larga y quizá controvertida de cualidades necesarias para obtener inteligencia artificialmente, él sugirió una prueba basada en la incapacidad de diferenciar entre entidades inteligentes indiscutibles y seres humanos. El computador supera la prueba si un evaluador humano no es capaz de distinguir si las respuestas, a una serie de preguntas planteadas, son de una persona o no. En el Capítulo 26 se comentan detalles de esta prueba y se discute si un computador que supera la prueba es realmente inteligente. Hoy por hoy, podemos decir que programar un computador para que supere la prueba requiere un trabajo considerable. El computador debería poseer las siguientes capacidades:

PROCESAMIENTO DE LENGUAJE NATURAL

REPRESENTACIÓN DEL CONOCIMIENTO

RAZONAMIENTO AUTOMÁTICO

APRENDIZAJE MÁQUINA

- **Procesamiento de lenguaje natural** que le permita comunicarse satisfactoriamente en inglés.
- **Representación del conocimiento** para almacenar lo que se conoce o siente.
- **Razonamiento automático** para utilizar la información almacenada para responder a preguntas y extraer nuevas conclusiones.
- **Aprendizaje automático** para adaptarse a nuevas circunstancias y para detectar y extrapolar patrones.

La Prueba de Turing evitó deliberadamente la interacción *física* directa entre el evaluador y el computador, dado que para medir la inteligencia es innecesario simular físicamente a una persona. Sin embargo, la llamada Prueba Global de Turing incluye una señal de vídeo que permite al evaluador valorar la capacidad de percepción del evaluado, y también le da la oportunidad al evaluador de pasar objetos físicos «a través de una ventanita». Para superar la Prueba Global de Turing el computador debe estar dotado de

PRUEBA DE TURING GLOBAL

VISTA COMPUTACIONAL

ROBÓTICA

- **Visión computacional** para percibir objetos.
- **Robótica** para manipular y mover objetos.

Estas seis disciplinas abarcan la mayor parte de la IA, y Turing merece ser reconocido por diseñar una prueba que se conserva vigente después de 50 años. Los investigadores del campo de la IA han dedicado poco esfuerzo a la evaluación de sus sistemas con la Prueba de Turing, por creer que es más importante el estudio de los principios en los que se basa la inteligencia que duplicar un ejemplar. La búsqueda de un ingenio que «volara artificialmente» tuvo éxito cuando los hermanos Wright, entre otros, dejaron de imitar a los pájaros y comprendieron los principios de la aerodinámica. Los textos de ingeniería aerodinámica no definen el objetivo de su campo como la construcción de «máquinas que vuelen como palomas de forma que puedan incluso confundir a otras palomas».

Pensar como un humano: el enfoque del modelo cognitivo

Para poder decir que un programa dado piensa como un humano, es necesario contar con un mecanismo para determinar cómo piensan los humanos. Es necesario *penetrar* en el

funcionamiento de las mentes humanas. Hay dos formas de hacerlo: mediante introspección (intentando atrapar nuestros propios pensamientos conforme éstos van apareciendo) y mediante experimentos psicológicos. Una vez se cuente con una teoría lo suficientemente precisa sobre cómo trabaja la mente, se podrá expresar esa teoría en la forma de un programa de computador. Si los datos de entrada/salida del programa y los tiempos de reacción son similares a los de un humano, existe la evidencia de que algunos de los mecanismos del programa se pueden comparar con los que utilizan los seres humanos. Por ejemplo, a Allen Newell y Herbert Simon, que desarrollaron el «Sistema de Resolución General de Problemas» (SRGP) (Newell y Simon, 1961), no les bastó con que su programa resolviera correctamente los problemas propuestos. Lo que les interesaba era seguir la pista de las etapas del proceso de razonamiento y compararlas con las seguidas por humanos a los que se les enfrentó a los mismos problemas. En el campo interdisciplinario de la **ciencia cognitiva** convergen modelos computacionales de IA y técnicas experimentales de psicología intentando elaborar teorías precisas y verificables sobre el funcionamiento de la mente humana.

CIENCIA COGNITIVA

La ciencia cognitiva es un campo fascinante, merecedora de una enciclopedia dedicada a ella (Wilson y Keil, 1999). En este libro no se intenta describir qué se conoce de la cognición humana. Ocasionalmente se hacen comentarios acerca de similitudes o diferencias entre técnicas de IA y cognición humana. La auténtica ciencia cognitiva se fundamenta necesariamente en la investigación experimental en humanos y animales, y en esta obra se asume que el lector sólo tiene acceso a un computador para experimentar.

En los comienzos de la IA había confusión entre las distintas aproximaciones: un autor podría argumentar que un algoritmo resolvía adecuadamente una tarea y que *por tanto* era un buen modelo de representación humana, o viceversa. Los autores actuales hacen diferencia entre las dos reivindicaciones; esta distinción ha permitido que ambas disciplinas, IA y ciencia cognitiva, se desarrollen más rápidamente. Los dos campos continúan alimentándose entre sí, especialmente en las áreas de la visión y el lenguaje natural. En particular, el campo de la visión ha avanzado recientemente con la ayuda de una propuesta integrada que tiene en cuenta la evidencia neurofisiológica y los modelos computacionales.

Pensamiento racional: el enfoque de las «leyes del pensamiento»

El filósofo griego Aristóteles fue uno de los primeros en intentar codificar la «manera correcta de pensar», es decir, un proceso de razonamiento irrefutable. Sus **silogismos** son esquemas de estructuras de argumentación mediante las que siempre se llega a conclusiones correctas si se parte de premisas correctas (por ejemplo: «Sócrates es un hombre; todos los hombres son mortales; por lo tanto Sócrates es mortal»). Estas leyes de pensamiento supuestamente gobiernan la manera de operar de la mente; su estudio fue el inicio de un campo llamado **lógica**.

SILOGISMOS

LÓGICA

Estudiosos de la lógica desarrollaron, en el siglo XIX, una notación precisa para definir sentencias sobre todo tipo de elementos del mundo y especificar relaciones entre

ellos (compárese esto con la notación aritmética común, que prácticamente sólo sirve para representar afirmaciones acerca de la igualdad y desigualdad entre números). Ya en 1965 existían programas que, en principio, resolvían *cualquier* problema resoluble descrito en notación lógica[2]. La llamada tradición **logista** dentro del campo de la inteligencia artificial trata de construir sistemas inteligentes a partir de estos programas.

LOGISTA

Este enfoque presenta dos obstáculos. No es fácil transformar conocimiento informal y expresarlo en los términos formales que requieren de notación lógica, particularmente cuando el conocimiento que se tiene es inferior al 100 por 100. En segundo lugar, hay una gran diferencia entre poder resolver un problema «en principio» y hacerlo en la práctica. Incluso problemas con apenas una docena de datos pueden agotar los recursos computacionales de cualquier computador a menos que cuente con alguna directiva sobre los pasos de razonamiento que hay que llevar a cabo primero. Aunque los dos obstáculos anteriores están presentes en *todo* intento de construir sistemas de razonamiento computacional, surgieron por primera vez en la tradición lógica.

Actuar de forma racional: el enfoque del agente racional

AGENTE

Un **agente** es algo que razona (*agente* viene del latín *agere*, hacer). Pero de los agentes informáticos se espera que tengan otros atributos que los distingan de los «programas» convencionales, como que estén dotados de controles autónomos, que perciban su entorno, que persistan durante un período de tiempo prolongado, que se adapten a los cambios, y que sean capaces de alcanzar objetivos diferentes. Un **agente racional** es aquel que actúa con la intención de alcanzar el mejor resultado o, cuando hay incertidumbre, el mejor resultado esperado.

AGENTE RACIONAL

En el caso del enfoque de la IA según las «leyes del pensamiento», todo el énfasis se pone en hacer inferencias correctas. La obtención de estas inferencias correctas puede, a veces, formar *parte* de lo que se considera un agente racional, ya que una manera racional de actuar es llegar a la conclusión lógica de que si una acción dada permite alcanzar un objetivo, hay que llevar a cabo dicha acción. Sin embargo, el efectuar una inferencia correcta no depende siempre de la *racionalidad*, ya que existen situaciones para las que no hay nada correcto que hacer y en las que hay que tomar una decisión. Existen también formas de actuar racionalmente que no implican realizar inferencias. Por ejemplo, el retirar la mano de una estufa caliente es un acto reflejo mucho más eficiente que una respuesta lenta llevada a cabo tras una deliberación cuidadosa.

Todas la habilidades que se necesitan en la Prueba de Turing deben permitir emprender acciones racionales. Por lo tanto, es necesario contar con la capacidad para representar el conocimiento y razonar basándonos en él, porque ello permitirá alcanzar decisiones correctas en una amplia gama de situaciones. Es necesario ser capaz de generar sentencias comprensibles en lenguaje natural, ya que el enunciado de tales oraciones permite a los agentes desenvolverse en una sociedad compleja. El aprendizaje no se lleva a cabo por erudición exclusivamente, sino que profundizar en el conocimiento de cómo funciona el mundo facilita la concepción de estrategias mejores para manejarse en él.

[2] Si no se encuentra una solución, el programa nunca debe parar de buscarla.

La percepción visual es necesaria no sólo porque ver es divertido, sino porque es necesaria para poder tener una idea mejor de lo que una acción puede llegar a representar, por ejemplo, el ver un delicioso bocadillo contribuirá a que nos acerquemos a él.

Por esta razón, el estudiar la IA desde el enfoque del diseño de un agente racional ofrece al menos dos ventajas. La primera es más general que el enfoque que proporcionan las «leyes del pensamiento», dado que el efectuar inferencias correctas es sólo uno de los mecanismos existentes para garantizar la racionalidad. La segunda es más afín a la forma en la que se ha producido el avance científico que los enfoques basados en la conducta o pensamiento humano, porque la norma de la racionalidad está claramente definida y es de aplicación general. Por el contrario, la conducta humana se adapta bien a un entorno específico, y en parte, es producto de un proceso evolutivo complejo, en gran medida desconocido, que aún está lejos de llevarnos a la perfección. *Por tanto, esta obra se centrará en los principios generales que rigen a los agentes racionales y en los elementos necesarios para construirlos*. Más adelante quedará patente que a pesar de la aparente facilidad con la que se puede describir un problema, cuando se intenta resolver surgen una enorme variedad de cuestiones. El Capítulo 2 revisa algunos de estos aspectos con más detalle.

Un elemento importante a tener en cuenta es el siguiente: más bien pronto que tarde se observará cómo obtener una racionalidad perfecta (hacer siempre lo correcto) no es posible en entornos complejos. La demanda computacional que esto implica es demasiado grande. En la mayor parte de esta obra se adoptará la hipótesis de trabajo de que la racionalidad perfecta es un buen punto de partida para el análisis. Lo cual simplifica el problema y proporciona el escenario base adecuado sobre el que se asientan los cimientos de este campo. Los Capítulos 6 y 17 se centran explícitamente en el tema de la **racionalidad limitada** (actuar adecuadamente cuando no se cuenta con el tiempo suficiente para efectuar todos los cálculos que serían deseables).

RACIONALIDAD LIMITADA

1.2 Los fundamentos de la inteligencia artificial

Esta sección presenta una breve historia de las disciplinas que han contribuido con ideas, puntos de vista y técnicas al desarrollo de la IA. Como toda revisión histórica, en este caso se centra en un pequeño número de personas, eventos e ideas e ignora otras que también fueron importantes. La historia se organiza en torno a una serie de cuestiones, dejando claro que no se quiere dar la impresión de que estas cuestiones son las únicas por las que las disciplinas se han preocupado y que el objetivo último de todas estas disciplinas era hacer avanzar la IA.

Filosofía (desde el año 428 a.C. hasta el presente)

- ¿Se pueden utilizar reglas formales para extraer conclusiones válidas?
- ¿Cómo se genera la inteligencia mental a partir de un cerebro físico?
- ¿De dónde viene el conocimiento?
- ¿Cómo se pasa del conocimiento a la acción?

Aristóteles (384-322 a.C.) fue el primero en formular un conjunto preciso de leyes que gobernaban la parte racional de la inteligencia. Él desarrolló un sistema informal para razonar adecuadamente con silogismos, que en principio permitía extraer conclusiones mecánicamente, a partir de premisas iniciales. Mucho después, Ramón Lull (d. 1315) tuvo la idea de que el razonamiento útil se podría obtener por medios artificiales. Sus «ideas» aparecen representadas en la portada de este manuscrito. Thomas Hobbes (1588-1679) propuso que el razonamiento era como la computación numérica, de forma que «nosotros sumamos y restamos silenciosamente en nuestros pensamientos». La automatización de la computación en sí misma estaba en marcha; alrededor de 1500, Leonardo da Vinci (1452-1519) diseñó, aunque no construyó, una calculadora mecánica; construcciones recientes han mostrado que su diseño era funcional. La primera máquina calculadora conocida se construyó alrededor de 1623 por el científico alemán Wilhelm Schickard (1592-1635), aunque la Pascalina, construida en 1642 por Blaise Pascal (1623-1662), sea más famosa. Pascal escribió que «la máquina aritmética produce efectos que parecen más similares a los pensamientos que a las acciones animales». Gottfried Wilhelm Leibniz (1646-1716) construyó un dispositivo mecánico con el objetivo de llevar a cabo operaciones sobre conceptos en lugar de sobre números, pero su campo de acción era muy limitado.

Ahora que sabemos que un conjunto de reglas pueden describir la parte racional y formal de la mente, el siguiente paso es considerar la mente como un sistema físico. René Descartes (1596-1650) proporciona la primera discusión clara sobre la distinción entre la mente y la materia y los problemas que surgen. Uno de los problemas de una concepción puramente física de la mente es que parece dejar poco margen de maniobra al libre albedrío: si el pensamiento está totalmente gobernado por leyes físicas, entonces una piedra podría «decidir» caer en dirección al centro de la Tierra gracias a su libre albedrío. A pesar de ser denodado defensor de la capacidad de razonamiento, Descartes fue un defensor del **dualismo**. Sostenía que existe una parte de la mente (o del alma o del espíritu) que está al margen de la naturaleza, exenta de la influencia de las leyes físicas. Los animales, por el contrario, no poseen esta cualidad dual; a ellos se le podría concebir como si se tratasen de máquinas. Una alternativa al dualismo es el **materialismo**, que considera que las operaciones del cerebro realizadas de acuerdo a las leyes de la física *constituyen* la mente. El libre albedrío es simplemente la forma en la que la percepción de las opciones disponibles aparecen en el proceso de selección.

DUALISMO

MATERIALISMO

Dada una mente física que gestiona conocimiento, el siguiente problema es establecer las fuentes de este conocimiento. El movimiento **empírico**, iniciado con el *Novum Organum*[3], de Francis Bacon (1561-1626), se caracteriza por el aforismo de John Locke (1632-1704): «Nada existe en la mente que no haya pasado antes por los sentidos». David Hume (1711-1776) propuso en *A Treatise of Human Nature* (Hume, 1739) lo que actualmente se conoce como principio de **inducción**: las reglas generales se obtienen mediante la exposición a asociaciones repetidas entres sus elementos. Sobre la base de las propuestas de Ludwig Wittgenstein (1889-1951) y Bertrand Russell (1872-1970), el famoso Círculo de Viena, liderado por Rudolf Carnap (1891-1970), desarrolló la doctrina del **positivismo lógico**. Esa doctrina sostiene que todo el conocimiento se puede

EMPÍRICO

INDUCCIÓN

[3] Una actualización del *Organon*, o instrumento de pensamiento, de Aristóteles.

caracterizar mediante teorías lógicas relacionadas, en última instancia, con **sentencias de observación** que corresponden a estímulos sensoriales[4]. La **teoría de la confirmación** de Carnap y Carl Hempel (1905-1997) intenta explicar cómo el conocimiento se obtiene a partir de la experiencia. El libro de Carnap, *The Logical Structure of the World* (1928), define un procedimiento computacional explícito para la extracción de conocimiento a partir de experiencias primarias. Fue posiblemente la primera teoría en mostrar la mente como un proceso computacional.

POSITIVISMO LÓGICO

SENTENCIA DE OBSERVACIÓN

TEORÍA DE LA CONFIRMACIÓN

El último elemento en esta discusión filosófica sobre la mente es la relación que existe entre conocimiento y acción. Este asunto es vital para la IA, ya que la inteligencia requiere tanto acción como razonamiento. Más aún, simplemente con comprender cómo se justifican determinadas acciones se puede llegar a saber cómo construir un agente cuyas acciones sean justificables (o racionales). Aristóteles argumenta que las acciones se pueden justificar por la conexión lógica entre los objetivos y el conocimiento de los efectos de las acciones (la última parte de este extracto también aparece en la portada de este libro):

> ¿Cómo es que el pensamiento viene acompañado en algunos casos de acciones y en otros no?, ¿en algunos casos por movimiento y en otros no? Parece como si la misma cosa sucediera tanto si razonáramos o hiciéramos inferencias sobre objetos que no cambian; pero en este caso el fin es una proposición especulativa... mientras la conclusión resultante de las dos premisas es una acción... Yo necesito abrigarme; una manta abriga. Yo necesito una manta. Qué necesito, qué debo hacer; necesito una manta. Necesito hacer una manta. Y la conclusión, «Yo tengo que hacer una manta», es una acción. (Nussbaum, 1978, p. 40)

En *Nicomachean Ethics* (Libro III. 3, 1112b), Aristóteles continúa trabajando en este tema, sugiriendo un algoritmo:

> Nosotros no reflexionamos sobre los fines, sino sobre los medios. Un médico no reflexiona sobre si debe curar, ni un orador sobre si debe persuadir... Ellos asumen el fin y consideran cómo y con qué medios se obtienen, y si resulta fácil y es por tanto productivo; mientras que si sólo se puede alcanzar por un medio se tiene en consideración *cómo* se alcanzará por este y por qué medios se obtendrá *éste*, hasta que se llegue a la causa primera..., y lo último en el orden del análisis parece ser lo primero en el orden de los acontecimientos. Y si se llega a un estado imposible, se abandona la búsqueda, como por ejemplo si se necesita dinero y no se puede conseguir; pero si hay una posibilidad se intentará.

El algoritmo de Aristóteles se implementó 2.300 años más tarde por Newell y Simon con la ayuda de su programa SRGP. El cual se conoce como sistema de planificación regresivo (*véase* el Capítulo 11).

El análisis basado en objetivos es útil, pero no indica qué hacer cuando varias acciones nos llevan a la consecución del objetivo, o cuando ninguna acción facilita su completa consecución. Antoine Arnauld (1612-1694) describió correctamente una forma cuantitativa para decidir qué acción llevar a cabo en un caso como este (*véase* el Capítulo 16). El libro *Utilitarianism* (Mill, 1863) de John Stuart Mill (1806-1873) propone

[4] En este contexto, es posible comprobar o rechazar toda aseveración significativa mediante el análisis del significado de las palabras o mediante la experimentación. Dado que esto no es aplicable en la mayor parte del ámbito de la metafísica, como era intención, el positivismo lógico se hizo impopular en algunos círculos.

la idea de un criterio de decisión racional en todos los ámbitos de la actividad humana. En la siguiente sección se explica una teoría de la decisión más formalmente.

Matemáticas (aproximadamente desde el año 800 al presente)

- ¿Qué reglas formales son las adecuadas para obtener conclusiones válidas?
- ¿Qué se puede computar?
- ¿Cómo razonamos con información incierta?

Los filósofos delimitaron las ideas más importantes de la IA, pero para pasar de ahí a una ciencia formal es necesario contar con una formulación matemática en tres áreas fundamentales: lógica, computación y probabilidad.

El concepto de lógica formal se remonta a los filósofos de la antigua Grecia (*véase* el Capítulo 7), pero su desarrollo matemático comenzó realmente con el trabajo de George Boole (1815-1864) que definió la lógica proposicional o Booleana (Boole, 1847). En 1879, Gottlob Frege (1848-1925) extendió la lógica de Boole para incluir objetos y relaciones, y creó la lógica de primer orden que se utiliza hoy como el sistema más básico de representación de conocimiento[5]. Alfred Tarski (1902-1983) introdujo una teoría de referencia que enseña cómo relacionar objetos de una lógica con objetos del mundo real. El paso siguiente consistió en definir los límites de lo que se podía hacer con la lógica y la informática.

ALGORITMO

Se piensa que el primer **algoritmo** no trivial es el algoritmo Euclídeo para el cálculo del máximo común divisor. El considerar los algoritmos como objetos en sí mismos se remonta a la época de al-Khowarazmi, un matemático persa del siglo IX, con cuyos escritos también se introdujeron los números arábigos y el álgebra en Europa. Boole, entre otros, presentó algoritmos para llevar a cabo deducciones lógicas y hacia el final del siglo XIX se llevaron a cabo numerosos esfuerzos para formalizar el razonamiento matemático general con la lógica deductiva. En 1900, David Hilbert (1862-1943) presentó una lista de 23 problemas que acertadamente predijo ocuparían a los matemáticos durante todo ese siglo. En el último de ellos se preguntaba si existe un algoritmo que permita determinar la validez de cualquier proposición lógica en la que aparezcan números naturales (el famoso *Entscheidungsproblem*, o problema de decisión). Básicamente, lo que Hilbert se preguntaba es si hay límites fundamentales en la capacidad de los procedimientos efectivos de demostración. En 1930, Kurt Gödel (1906-1978) demostró que existe un procedimiento eficiente para demostrar cualquier aseveración verdadera en la lógica de primer orden de Frege y Russell, sin embargo con la lógica de primer orden no era posible capturar el principio de inducción matemática necesario para la caracterización de los números naturales. En 1931, demostró que, en efecto, existen límites reales. Mediante su **teorema de incompletitud** demostró que en cualquier lenguaje que tuviera la capacidad suficiente para expresar las propiedades de los números naturales, existen aseveraciones verdaderas no decidible en el sentido de que no es posible decidir su validez mediante ningún algoritmo.

TEOREMA DE INCOMPLETITUD

[5] La notación para la lógica de primer orden propuesta por Frege no se ha aceptado universalmente, por razones que son aparentemente obvias cuando se observa el ejemplo que aparece en la cubierta de este libro.

El resultado fundamental anterior se puede interpretar también como la indicación de que existen algunas funciones de los números enteros que no se pueden representar mediante un algoritmo, es decir no se pueden calcular. Lo anterior llevó a Alan Turing (1912-1954) a tratar de caracterizar exactamente aquellas funciones que sí *eran* susceptibles de ser caracterizadas. La noción anterior es de hecho problemática hasta cierto punto, porque no es posible dar una definición formal a la noción de cálculo o procedimiento efectivo. No obstante, la tesis de Church-Turing, que afirma que la máquina de Turing (Turing, 1936) es capaz de calcular cualquier función computable, goza de aceptación generalizada ya que proporciona una definición suficiente. Turing también demostró que existen algunas funciones que no se pueden calcular mediante la máquina de Turing. Por ejemplo, ninguna máquina puede decidir *en general* si un programa dado producirá una respuesta a partir de unas entradas, o si seguirá calculando indefinidamente.

INTRATABILIDAD

Si bien ser no decidible ni computable son importantes para comprender el proceso del cálculo, la noción de **intratabilidad** tuvo repercusiones más importantes. En términos generales se dice que un problema es intratable si el tiempo necesario para la resolución de casos particulares de dicho problema crece exponencialmente con el tamaño de dichos casos. La diferencia entre crecimiento polinomial y exponencial de la complejidad se destacó por primera vez a mediados de los años 60 (Cobham, 1964; Edmonds, 1965). Es importante porque un crecimiento exponencial implica la imposibilidad de resolver casos moderadamente grandes en un tiempo razonable. Por tanto, se debe optar por dividir el problema de la generación de una conducta inteligente en subproblemas que sean tratables en vez de manejar problemas intratables.

NP-COMPLETITUD

¿Cómo se puede reconocer un problema intratable? La teoría de la **NP-completitud**, propuesta por primera vez por Steven Cook (1971) y Richard Karp (1972) propone un método. Cook y Karp demostraron la existencia de grandes clases de problemas de razonamiento y búsqueda combinatoria canónica que son NP completos. Toda clase de problema a la que la clase de problemas NP completos se pueda reducir será seguramente intratable (aunque no se ha demostrado que los problemas NP completos son necesariamente intratables, la mayor parte de los teóricos así lo creen). Estos resultados contrastan con el optimismo con el que la prensa popular recibió a los primeros computadores, «Supercerebros Electrónicos» que eran «¡Más rápidos que Einstein!». A pesar del rápido incremento en la velocidad de los computadores, los sistemas inteligentes se caracterizarán por el uso cuidadoso que hacen de los recursos. De manera sucinta, ¡el mundo es un ejemplo de problema *extremadamente* grande! Recientemente la IA ha ayudado a explicar por qué algunos ejemplos de problemas NP completos son difíciles de resolver y otros son fáciles (Cheeseman *et al*., 1991).

PROBABILIDAD

Además de la lógica y el cálculo, la tercera gran contribución de las matemáticas a la IA es la teoría de la **probabilidad**. El italiano Gerolamo Cardano (1501-1576) fue el primero en proponer la idea de probabilidad, presentándola en términos de los resultados de juegos de apuesta. La probabilidad se convirtió pronto en parte imprescindible de las ciencias cuantitativas, ayudando en el tratamiento de mediciones con incertidumbre y de teorías incompletas. Pierre Fermat (1601-1665), Blaise Pascal (1623-1662), James Bernoulli (1654-1705), Pierre Laplace (1749-1827), entre otros, hicieron avanzar esta teoría e introdujeron nuevos métodos estadísticos. Thomas Bayes (1702-1761) propuso una

regla para la actualización de probabilidades subjetivas a la luz de nuevas evidencias. La regla de Bayes y el área resultante llamado análisis Bayesiano conforman la base de las propuestas más modernas que abordan el razonamiento incierto en sistemas de IA.

Economía (desde el año 1776 hasta el presente)

- ¿Cómo se debe llevar a cabo el proceso de toma de decisiones para maximizar el rendimiento?
- ¿Cómo se deben llevar a cabo acciones cuando otros no colaboren?
- ¿Cómo se deben llevar a cabo acciones cuando los resultados se obtienen en un futuro lejano?

La ciencia de la economía comenzó en 1776, cuando el filósofo escocés Adam Smith (1723-1790) publicó *An Inquiri into the Nature and Causes of the Wealth of Nations*. Aunque los antiguos griegos, entre otros, habían hecho contribuciones al pensamiento económico, Smith fue el primero en tratarlo como una ciencia, utilizando la idea de que las economías pueden concebirse como un conjunto de agentes individuales que intentan maximizar su propio estado de bienestar económico. La mayor parte de la gente cree que la economía sólo se trata de dinero, pero los economistas dicen que ellos realmente estudian cómo la gente toma decisiones que les llevan a obtener los beneficios esperados. Léon Walras (1834-1910) formalizó el tratamiento matemático del «beneficio deseado» o **utilidad,** y fue posteriormente mejorado por Frank Ramsey (1931) y después por John von Neumann y Oskar Morgenstern en su libro *The Theory of Games and Economic Behavior* (1944).

TEORÍA DE LA DECISIÓN

La **teoría de la decisión**, que combina la teoría de la probabilidad con la teoría de la utilidad, proporciona un marco completo y formal para la toma de decisiones (económicas o de otra índole) realizadas bajo incertidumbre, esto es, en casos en los que las descripciones probabilísticas capturan adecuadamente la forma en la que se toman las decisiones en el entorno; lo cual es adecuado para «grandes» economías en las que cada agente no necesita prestar atención a las acciones que lleven a cabo el resto de los agentes individualmente. Cuando se trata de «pequeñas» economías, la situación se asemeja más a la de un **juego**: las acciones de un jugador pueden afectar significativamente a la utilidad de otro (tanto positiva como negativamente). Los desarrollos de von Neumann y Morgenstern a partir de la **teoría de juegos** (*véase* también Luce y Raiffa, 1957) mostraban el hecho sorprendente de que, en algunos juegos, un agente racional debía actuar de forma aleatoria o, al menos, aleatoria en apariencia con respecto a sus contrincantes.

TEORÍA DE JUEGOS

La gran mayoría de los economistas no se preocuparon de la tercera cuestión mencionada anteriormente, es decir, cómo tomar decisiones racionales cuando los resultados de las acciones no son inmediatos y por el contrario se obtienen los resultados de las acciones de forma *secuencial*. El campo de la **investigación operativa** persigue este objetivo; dicho campo emergió en la Segunda Guerra Mundial con los esfuerzos llevados a cabo en el Reino Unido en la optimización de instalaciones de radar, y posteriormente en aplicaciones civiles relacionadas con la toma de decisiones de dirección complejas. El trabajo de Richard Bellman (1957) formaliza una clase de problemas de decisión secuencial llamados **procesos de decisión de Markov**, que se estudiarán en los Capítulos 17 y 21.

INVESTIGACIÓN OPERATIVA

El trabajo en la economía y la investigación operativa ha contribuido en gran medida a la noción de agente racional que aquí se presenta, aunque durante muchos años la investigación en el campo de la IA se ha desarrollado por sendas separadas. Una razón fue la **complejidad** aparente que trae consigo el tomar decisiones racionales. Herbert Simon (1916-2001), uno de los primeros en investigar en el campo de la IA, ganó el premio Nobel en Economía en 1978 por su temprano trabajo, en el que mostró que los modelos basados en **satisfacción** (que toman decisiones que son «suficientemente buenas», en vez de realizar cálculos laboriosos para alcanzar decisiones óptimas) proporcionaban una descripción mejor del comportamiento humano real (Simon, 1947). En los años 90, hubo un resurgimiento del interés en las técnicas de decisión teórica para sistemas basados en agentes (Wellman, 1995).

SATISFACCIÓN

Neurociencia (desde el año 1861 hasta el presente)

- ¿Cómo procesa información el cerebro?

NEUROCIENCIA

La **Neurociencia** es el estudio del sistema neurológico, y en especial del cerebro. La forma exacta en la que en un cerebro se genera el pensamiento es uno de los grandes misterios de la ciencia. Se ha observado durante miles de años que el cerebro está de alguna manera involucrado en los procesos de pensamiento, ya que fuertes golpes en la cabeza pueden ocasionar minusvalía mental. También es ampliamente conocido que los cerebros humanos son de alguna manera diferentes; aproximadamente en el 335 a.C. Aristóteles escribió, «de entre todos los animales el hombre tiene el cerebro más grande en proporción a su tamaño»[6]. Aunque, no fue hasta mediados del siglo XVIII cuando se aceptó mayoritariamente que el cerebro es la base de la conciencia. Hasta este momento, se pensaba que estaba localizado en el corazón, el bazo y la glándula pineal.

El estudio de Paul Broca (1824-1880) sobre la afasia (dificultad para hablar) en pacientes con el cerebro dañado, en 1861, le dio fuerza a este campo y convenció a la sociedad médica de la existencia de áreas localizadas en el cerebro responsables de funciones cognitivas específicas. En particular, mostró que la producción del habla se localizaba en una parte del hemisferio izquierdo; hoy en día conocida como el área de Broca[7]. En esta época ya se sabía que el cerebro estaba formado por células nerviosas o **neuronas**, pero no fue hasta 1873 cuando Camillo Golgi (1843-1926) desarrolló una técnica de coloración que permitió la observación de neuronas individuales en el cerebro (*véase* la Figura 1.2). Santiago Ramón y Cajal (1852-1934) utilizó esta técnica es sus estudios pioneros sobre la estructura neuronal del cerebro[8].

NEURONAS

En la actualidad se dispone de información sobre la relación existente entre las áreas del cerebro y las partes del cuerpo humano que controlan o de las que reciben impulsos

[6] Desde entonces, se ha descubierto que algunas especies de delfines y ballenas tienen cerebros relativamente grandes. Ahora se piensa que el gran tamaño de los cerebros humanos se debe en parte a la mejora reciente en su sistema de refrigeración.

[7] Muchos citan a Alexander Hood (1824) como una fuente posiblemente anterior.

[8] Golgi insistió en la creencia de que las funciones cerebrales se desarrollaron inicialmente en el medio continuo en el que las neuronas estaban inmersas, mientras que Cajal propuso la «doctrina neuronal». Ambos compartieron el premio Nobel en 1906 pronunciando un discurso de aceptación antagónico.

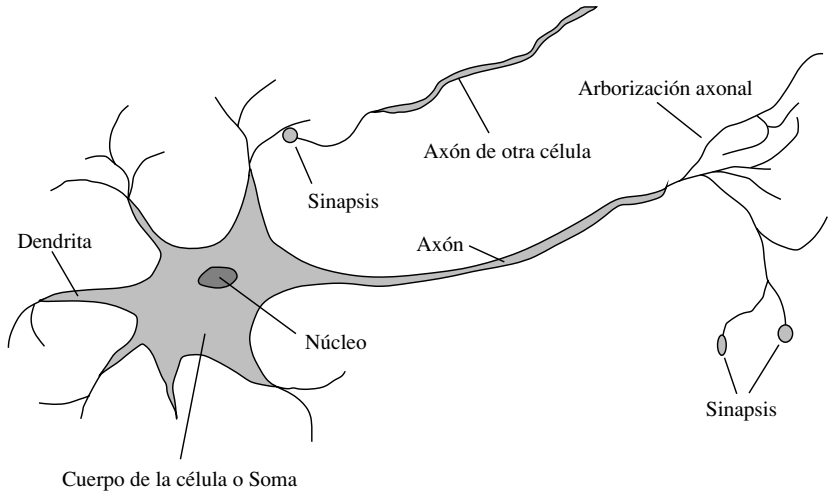

Figura 1.2 Partes de una célula nerviosa o neurona. Cada neurona contiene un cuerpo celular, o soma, que tiene un núcleo celular. Un número de fibras llamadas dendritas se ramifican a partir del cuerpo de la célula junto con una única fibra larga llamada axón. El axón se alarga considerablemente, mucho más que lo que se representa en esta imagen. Normalmente miden un centímetro (100 veces más que el diámetro del cuerpo de la célula), pero pueden alcanzar hasta un metro de longitud. Una neurona se conecta con entre 10 y 100.000 neuronas formando una maraña llamada sinapsis. Las señales se propagan de neurona a neurona mediante una reacción electroquímica complicada. Las señales controlan la actividad del cerebro a corto plazo, y permiten establecer cambios en la posición y conectividad de las neuronas a largo plazo. Se piensa que estos mecanismos forman la base del aprendizaje del cerebro. La mayoría del procesamiento de información se lleva a cabo en la corteza del cerebro, la capa más externa de éste. La unidad de organización básica es una columna de tejido de aproximadamente 0,5 mm de diámetro, con lo cual se amplía la profundidad total de la corteza cerebral, que en el caso de los humanos es de cuatro mm. Una columna contiene aproximadamente 20.000 neuronas.

sensoriales. Tales relaciones pueden cambiar de forma radical incluso en pocas semanas, y algunos animales parecen disponer de múltiples posibilidades. Más aún, no se tiene totalmente claro cómo algunas áreas se pueden encargar de ciertas funciones que eran responsabilidad de áreas dañadas. No hay prácticamente ninguna teoría que explique cómo se almacenan recuerdos individuales.

Los estudios sobre la actividad de los cerebros intactos comenzó en 1929 con el descubrimiento del electroencefalograma (EEG) desarrollado por Hans Berger. El reciente descubrimiento de las imágenes de resonancia magnética funcional (IRMF) (Ogawa *et al*., 1990) está proporcionando a los neurólogos imágenes detalladas de la actividad cerebral sin precedentes, permitiéndoles obtener medidas que se corresponden con procesos cognitivos en desarrollo de manera muy interesante. Este campo está evolucionando gracias a los avances en los estudios en celdas individuales y su actividad neuronal. A pesar de estos avances, nos queda un largo camino para llegar a comprender cómo funcionan todos estos procesos cognitivos.

La conclusión verdaderamente increíble es que *una colección de simples células puede llegar a generar razonamiento, acción, y conciencia* o, dicho en otras palabras, *los cerebros generan las inteligencias* (Searle, 1992). La única teoría alternativa es el misticismo: que nos dice que existe alguna esfera mística en la que las mentes operan fuera del control de la ciencia física.

Cerebros y computadores digitales realizan tareas bastante diferentes y tienen propiedades distintas. La Figura 1.3 muestra cómo hay 1.000 veces más neuronas en un cerebro humano medio que puertas lógicas en la UCP de un computador estándar. La ley de Moore[9] predice que el número de puertas lógicas de la UCP se igualará con el de neuronas del cerebro alrededor del año 2020. Por supuesto, poco se puede inferir de esta predicción; más aún, la diferencia en la capacidad de almacenamiento es insignificante comparada con las diferencias en la velocidad de intercambio y en paralelismo. Los circuitos de los computadores pueden ejecutar una instrucción en un nanosegundo, mientras que las neuronas son millones de veces más lentas. Las neuronas y las sinapsis del cerebro están activas simultáneamente, mientras que los computadores actuales tienen una o como mucho varias UCP. Por tanto, incluso sabiendo que un computador es un millón de veces más rápido en cuanto a su velocidad de intercambio, el cerebro acaba siendo 100.000 veces más rápido en lo que hace.

Psicología (desde el año 1879 hasta el presente)

- ¿Cómo piensan y actúan los humanos y los animales?

La psicología científica se inició con los trabajos del físico alemán Hermann von Helmholtz (1821-1894), según se referencia habitualmente, y su discípulo Wilhelm Wundt (1832-1920). Helmholtz aplicó el método científico al estudio de la vista humana, y su obra *Handbook of Physiological Optics*, todavía en nuestros días, se considera como «el tratado actual más importante sobre la física y la fisiología de la vista humana» (Nalwa, 1993, p. 15). En 1879, Wundt abrió el primer laboratorio de psicología experimental en

	Computador	Cerebro Humano
Unidades computacionales	1 UCP, 10^8 puertas	10^{11} neuronas
Unidades de Almacenamiento	10^{10} bits RAM	10^{11} neuronas
	10^{11} bits disco	10^{14} sinapsis
Duración de un ciclo	10^{-9} sec	10^{-3} sec
Ancho de banda	10^{10} bits/sec	10^{14} bits/sec
Memoria actualización/sec	10^9	10^{14}

Figura 1.3 Comparación básica entre los recursos de cómputo generales de que disponen los computadores (*circa 2003*) y el cerebro. Las cifras correspondientes a los computadores se han incrementado en al menos un factor 10 desde la primera edición de este libro, y se espera que suceda la mismo en esta década. Las cifras correspondientes al cerebro no han cambiado en los últimos 10.000 años.

[9] La ley de Moore dice que el número de transistores por pulgada cuadrada se duplica cada año o año y medio. La capacidad del cerebro humano se dobla aproximadamente cada dos o cuatro millones de años.

la Universidad de Leipzig. Wundt puso mucho énfasis en la realización de experimentos controlados cuidadosamente en la que sus operarios realizaban tareas de percepción o asociación al tiempo que sometían a introspección sus procesos mentales. Los meticulosos controles evolucionaron durante un largo período de tiempo hasta convertir la psicología en una ciencia, pero la naturaleza subjetiva de los datos hizo poco probable que un investigador pudiera contradecir sus propias teorías. Biólogos, estudiando el comportamiento humano, por el contrario, carecían de datos introspectivos y desarrollaron una metodología objetiva, tal y como describe H. S. Jennings (1906) en su influyente trabajo *Behavior of the Lower Organisms*. El movimiento **conductista**, liderado por John Watson (1878-1958) aplicó este punto de vista a los humanos, rechazando *cualquier* teoría en la que intervinieran procesos mentales, argumentando que la introspección no aportaba una evidencia fiable. Los conductistas insistieron en el estudio exclusivo de mediciones objetivas de percepciones (o *estímulos*) sobre animales y de las acciones resultantes (o *respuestas*). Construcciones mentales como conocimientos, creencias, objetivos y pasos en un razonamiento quedaron descartadas por ser consideradas «psicología popular» no científica. El conductismo hizo muchos descubrimientos utilizando ratas y palomas, pero tuvo menos éxito en la comprensión de los seres humanos. Aún así, su influencia en la psicología fue notable (especialmente en Estados Unidos) desde aproximadamente 1920 hasta 1960.

CONDUCTISMO

La conceptualización del cerebro como un dispositivo de procesamiento de información, característica principal de la **psicología cognitiva**, se remonta por lo menos a las obras de William James[10] (1842-1910). Helmholtz también pone énfasis en que la percepción entraña cierto tipo de inferencia lógica inconsciente. Este punto de vista cognitivo se vio eclipsado por el conductismo en Estados Unidos, pero en la Unidad de Psicología Aplicada de Cambridge, dirigida por Frederic Bartlett (1886-1969), los modelos cognitivos emergieron con fuerza. La obra *The Nature of Explanation*, de Kenneth Craik (1943), discípulo y sucesor de Bartlett, reestablece enérgicamente la legitimidad de términos «mentales» como creencias y objetivos, argumentando que son tan científicos como lo pueden ser la presión y la temperatura cuando se habla acerca de los gases, a pesar de que éstos estén formados por moléculas que no tienen ni presión ni temperatura. Craik establece tres elementos clave que hay que tener en cuenta para diseñar un agente basado en conocimiento: (1) el estímulo deberá ser traducido a una representación interna, (2) esta representación se debe manipular mediante procesos cognitivos para así generar nuevas representaciones internas, y (3) éstas, a su vez, se traducirán de nuevo en acciones. Dejó muy claro por qué consideraba que estos eran los requisitos idóneos para diseñar un agente:

PSICOLOGÍA COGNITIVA

> Si el organismo tiene en su cabeza «un modelo a pequeña escala» de la realidad externa y de todas sus posibles acciones, será capaz de probar diversas opciones, decidir cuál es la mejor, planificar su reacción ante posibles situaciones futuras antes de que éstas surjan, emplear lo aprendido de experiencias pasadas en situaciones presentes y futuras, y en todo momento, reaccionar ante los imprevistos que acontezcan de manera satisfactoria, segura y más competente (Craik, 1943).

[10] William James era hermano del novelista Henry James. Se comenta que Henry escribió novelas narrativas como si se tratara de psicología y William escribió sobre psicología como si se tratara de novelas narrativas.

Después de la muerte de Craik en un accidente de bicicleta en 1945, Donald Boadbent continuó su trabajo, y su libro *Perception and Communication* (1958) incluyó algunos de los primeros modelos de procesamiento de información del fenómeno psicológico. Mientras tanto, en Estados Unidos el desarrollo del modelo computacional llevo a la creación del campo de la **ciencia cognitiva**. Se puede decir que este campo comenzó en un simposio celebrado en el MIT, en septiembre de 1956 (como se verá a continuación este evento tuvo lugar sólo dos meses después de la conferencia en la que «nació» la IA). En este simposio, George Miller presentó *The Magic Number Seven*, Noam Chomsky presentó *Three Models of Language*, y Allen Newell y Herbert Simon presentaron *The Logic Theory Machine*. Estos tres artículos influyentes mostraron cómo se podían utilizar los modelos informáticos para modelar la psicología de la memoria, el lenguaje y el pensamiento lógico, respectivamente. Los psicólogos comparten en la actualidad el punto de vista común de que «la teoría cognitiva debe ser como un programa de computador» (Anderson, 1980), o dicho de otra forma, debe describir un mecanismo de procesamiento de información detallado, lo cual lleva consigo la implementación de algunas funciones cognitivas.

CIENCIA COGNITIVA

Ingeniería computacional (desde el año 1940 hasta el presente)

- ¿Cómo se puede construir un computador eficiente?

Para que la inteligencia artificial pueda llegar a ser una realidad se necesitan dos cosas: inteligencia y un artefacto. El computador ha sido el artefacto elegido. El computador electrónico digital moderno se inventó de manera independiente y casi simultánea por científicos en tres países involucrados en la Segunda Guerra Mundial. El equipo de Alan Turing construyó, en 1940, el primer computador *operacional* de carácter electromecánico, llamado Heath Robinson[11], con un único propósito: descifrar mensajes alemanes. En 1943 el mismo grupo desarrolló el Colossus, una máquina potente de propósito general basada en válvulas de vacío[12]. El primer computador operacional *programable* fue el Z-3, inventado por Konrad Zuse en Alemania, en 1941. Zuse también inventó los números de coma flotante y el primer lenguaje de programación de alto nivel, Plankalkül. El primer computador *electrónico*, el ABC, fue creado por John Atanasoff junto a su discípulo Clifford Berry entre 1940 y 1942 en la Universidad Estatal de Iowa. Las investigaciones de Atanasoff recibieron poco apoyo y reconocimiento; el ENIAC, desarrollado en el marco de un proyecto militar secreto, en la Universidad de Pensilvania, por un equipo en el que trabajaban entre otros John Mauchly y John Eckert, puede considerarse como el precursor de los computadores modernos.

Desde mediados del siglo pasado, cada generación de dispositivos *hardware* ha conllevado un aumento en la velocidad de proceso y en la capacidad de almacenamiento,

[11] Heath Robinson fue un caricaturista famoso por sus dibujos, que representaban artefactos de uso diario, caprichosos y absurdamente complicados, por ejemplo, uno para untar mantequilla en el pan tostado.

[12] En la postguerra, Turing quiso utilizar estos computadores para investigar en el campo de la IA, por ejemplo, desarrollando uno de los primeros programas para jugar a la ajedrez (Turing *et al*., 1953). El gobierno británico bloqueó sus esfuerzos.

así como una reducción de precios. La potencia de los computadores se dobla cada 18 meses aproximadamente y seguirá a este ritmo durante una o dos décadas más. Después, se necesitará ingeniería molecular y otras tecnologías novedosas.

Por supuesto que antes de la aparición de los computadores ya había dispositivos de cálculo. Las primeras máquinas automáticas, que datan del siglo XVII, ya se mencionaron en la página seis. La primera máquina programable fue un telar, desarrollado en 1805 por Joseph Marie Jacquard (1752-1834) que utilizaba tarjetas perforadas para almacenar información sobre los patrones de los bordados. A mediados del siglo XIX, Charles Babbage (1792-1871) diseñó dos máquinas, que no llegó a construir. La «Máquina de Diferencias», que aparece en la portada de este manuscrito, se concibió con la intención de facilitar los cálculos de tablas matemáticas para proyectos científicos y de ingeniería. Finalmente se construyó y se presentó en 1991 en el Museo de la Ciencia de Londres (Swade, 1993). La «Máquina Analítica» de Babbage era mucho más ambiciosa: incluía memoria direccionable, programas almacenados y saltos condicionales; fue el primer artefacto dotado de los elementos necesarios para realizar una computación universal. Ada Lovelace, colega de Babbage e hija del poeta Lord Byron, fue seguramente la primera programadora (el lenguaje de programación Ada se llama así en honor a esta programadora). Ella escribió programas para la inacabada Máquina Analítica e incluso especuló acerca de la posibilidad de que la máquina jugara al ajedrez y compusiese música.

La IA también tiene una deuda con la parte *software* de la informática que ha proporcionado los sistemas operativos, los lenguajes de programación, y las herramientas necesarias para escribir programas modernos (y artículos sobre ellos). Sin embargo, en este área la deuda se ha saldado: la investigación en IA ha generado numerosas ideas novedosas de las que se ha beneficiado la informática en general, como por ejemplo el tiempo compartido, los intérpretes imperativos, los computadores personales con interfaces gráficas y ratones, entornos de desarrollo rápido, listas enlazadas, administración automática de memoria, y conceptos claves de la programación simbólica, funcional, dinámica y orientada a objetos.

Teoría de control y cibernética (desde el año 1948 hasta el presente)

- ¿Cómo pueden los artefactos operar bajo su propio control?

Ktesibios de Alejandría (250 a.C.) construyó la primera máquina auto controlada: un reloj de agua con un regulador que mantenía el flujo de agua circulando por él, con un ritmo constante y predecible. Esta invención cambió la definición de lo que un artefacto podía hacer. Anteriormente, solamente seres vivos podían modificar su comportamiento como respuesta a cambios en su entorno. Otros ejemplos de sistemas de control auto regulables y retroalimentados son el motor de vapor, creado por James Watt (1736-1819), y el termostato, inventado por Cornelis Drebbel (1572-1633), que también inventó el submarino. La teoría matemática de los sistemas con retroalimentación estables se desarrolló en el siglo XIX.

TEORÍA DE CONTROL

La figura central del desarrollo de lo que ahora se llama la **teoría de control** fue Norbert Wiener (1894-1964). Wiener fue un matemático brillante que trabajó en sistemas de control biológicos y mecánicos y en sus vínculos con la cognición. De la misma forma que Craik (quien también utilizó sistemas de control como modelos psicológicos), Wiener y sus colegas Arturo Rosenblueth y Julian Bigelow desafiaron la ortodoxia conductista (Rosenblueth *et al*., 1943). Ellos veían el comportamiento determinista como algo emergente de un mecanismo regulador que intenta minimizar el «error» (la diferencia entre el estado presente y el estado objetivo). A finales de los años 40, Wiener, junto a Warren McCulloch, Walter Pitts y John von Neumann, organizaron una serie de conferencias en las que se exploraban los nuevos modelos cognitivos matemáticos y computacionales, e influyeron en muchos otros investigadores en el campo de las ciencias del comportamiento. El libro de Wiener, *Cybernetics* (1948), fue un *bestseller* y desveló al público las posibilidades de las máquinas con inteligencia artificial.

CIBERNÉTICA

FUNCIÓN OBJETIVO

La teoría de control moderna, especialmente la rama conocida como control óptimo estocástico, tiene como objetivo el diseño de sistemas que maximizan una **función objetivo** en el tiempo. Lo cual se asemeja ligeramente a nuestra visión de lo que es la IA: diseño de sistemas que se comportan de forma óptima. ¿Por qué, entonces, IA y teoría de control son dos campos diferentes, especialmente teniendo en cuenta la cercana relación entre sus creadores? La respuesta está en el gran acoplamiento existente entre las técnicas matemáticas con las que estaban familiarizados los investigadores y entre los conjuntos de problemas que se abordaban desde cada uno de los puntos de vista. El cálculo y el álgebra matricial, herramientas de la teoría de control, se utilizaron en la definición de sistemas que se podían describir mediante conjuntos fijos de variables continuas; más aún, el análisis exacto es sólo posible en sistemas *lineales*. La IA se fundó en parte para escapar de las limitaciones matemáticas de la teoría de control en los años 50. Las herramientas de inferencia lógica y computación permitieron a los investigadores de IA afrontar problemas relacionados con el lenguaje, visión y planificación, que estaban completamente fuera del punto de mira de la teoría de control.

Lingüística (desde el año 1957 hasta el presente)

- ¿Cómo está relacionado el lenguaje con el pensamiento?

En 1957, B. F. Skinner publicó *Verbal Behavior*. La obra presentaba una visión extensa y detallada desde el enfoque conductista al aprendizaje del lenguaje, y estaba escrita por los expertos más destacados de este campo. Curiosamente, una revisión de este libro llegó a ser tan famosa como la obra misma, y provocó el casi total desinterés por el conductismo. El autor de la revisión fue Noam Chomsky, quien acababa de publicar un libro sobre su propia teoría, *Syntactic Structures*. Chomsky mostró cómo la teoría conductista no abordaba el tema de la creatividad en el lenguaje: no explicaba cómo es posible que un niño sea capaz de entender y construir oraciones que nunca antes ha escuchado. La teoría de Chomsky (basada en modelos sintácticos que se remontaban al lingüista indio Panini, aproximadamente 350 a.C.) sí podía explicar lo anterior y, a diferencia de teorías anteriores, poseía el formalismo suficiente como para permitir su programación.

LINGÜÍSTICA COMPUTACIONAL

La lingüística moderna y la IA «nacieron», al mismo tiempo y maduraron juntas, solapándose en un campo híbrido llamado **lingüística computacional** o **procesamiento del lenguaje natural**. El problema del entendimiento del lenguaje se mostró pronto mucho más complejo de lo que se había pensado en 1957. El entendimiento del lenguaje requiere la comprensión de la materia bajo estudio y de su contexto, y no solamente el entendimiento de la estructura de las sentencias. Lo cual puede parecer obvio, pero no lo fue para la mayoría de la comunidad investigadora hasta los años 60. Gran parte de los primeros trabajos de investigación en el área de la **representación del conocimiento** (el estudio de cómo representar el conocimiento de forma que el computador pueda razonar a partir de dicha representación) estaban vinculados al lenguaje y a la búsqueda de información en el campo del lenguaje, y su base eran las investigaciones realizadas durante décadas en el análisis filosófico del lenguaje.

1.3 Historia de la inteligencia artificial

Una vez revisado el material básico estamos ya en condiciones de cubrir el desarrollo de la IA propiamente dicha.

Génesis de la inteligencia artificial (1943–1955)

Warren McCulloch y Walter Pitts (1943) han sido reconocidos como los autores del primer trabajo de IA. Partieron de tres fuentes: conocimientos sobre la fisiología básica y funcionamiento de las neuronas en el cerebro, el análisis formal de la lógica proposicional de Russell y Whitehead y la teoría de la computación de Turing. Propusieron un modelo constituido por neuronas artificiales, en el que cada una de ellas se caracterizaba por estar «activada» o «desactivada»; la «activación» se daba como respuesta a la estimulación producida por una cantidad suficiente de neuronas vecinas. El estado de una neurona se veía como «equivalente, de hecho, a una proposición con unos estímulos adecuados». Mostraron, por ejemplo, que cualquier función de cómputo podría calcularse mediante alguna red de neuronas interconectadas, y que todos los conectores lógicos (*and*, *or*, *not*, etc.) se podrían implementar utilizando estructuras de red sencillas. McCulloch y Pitts también sugirieron que redes adecuadamente definidas podrían aprender. Donald Hebb (1949) propuso y demostró una sencilla regla de actualización para modificar las intensidades de las conexiones entre neuronas. Su regla, ahora llamada **de aprendizaje Hebbiano o de Hebb**, sigue vigente en la actualidad.

Dos estudiantes graduados en el Departamento de Matemáticas de Princeton, Marvin Minsky y Dean Edmonds, construyeron el primer computador a partir de una red neuronal en 1951. El SNARC, como se llamó, utilizaba 3.000 válvulas de vacío y un mecanismo de piloto automático obtenido de los desechos de un avión bombardero B-24 para simular una red con 40 neuronas. El comité encargado de evaluar el doctorado de Minsky veía con escepticismo el que este tipo de trabajo pudiera considerarse como matemático, pero se dice que von Newmann dijo, «Si no lo es actualmente, algún día lo será».

Minsky posteriormente probó teoremas influyentes que mostraron las limitaciones de la investigación con redes neuronales.

Hay un número de trabajos iniciales que se pueden caracterizar como de IA, pero fue Alan Turing quien articuló primero una visión de la IA en su artículo *Computing Machinery and Intelligence*, en 1950. Ahí, introdujo la prueba de Turing, el aprendizaje automático, los algoritmos genéricos y el aprendizaje por refuerzo.

Nacimiento de la inteligencia artificial (1956)

Princeton acogió a otras de la figuras señeras de la IA, John McCarthy. Posteriormente a su graduación, McCarthy se transladó al Dartmouth College, que se erigiría en el lugar del nacimiento oficial de este campo. McCarthy convenció a Minsky, Claude Shannon y Nathaniel Rochester para que le ayudaran a aumentar el interés de los investigadores americanos en la teoría de autómatas, las redes neuronales y el estudio de la inteligencia. Organizaron un taller con una duración de dos meses en Darmouth en el verano de 1956. Hubo diez asistentes en total, entre los que se incluían Trenchard More de Princeton, Arthur Samuel de IBM, y Ray Solomonoff y Oliver Selfridge del MIT.

Dos investigadores del Carnegie Tech[13], Allen Newell y Herbert Simon, acapararon la atención. Si bien los demás también tenían algunas ideas y, en algunos casos, programas para aplicaciones determinadas como el juego de damas, Newell y Simon contaban ya con un programa de razonamiento, el Teórico Lógico (TL), del que Simon afirmaba: «Hemos inventado un programa de computación capaz de pensar de manera no numérica, con lo que ha quedado resuelto el venerable problema de la dualidad mente-cuerpo»[14]. Poco después del término del taller, el programa ya era capaz de demostrar gran parte de los teoremas del Capítulo 2 de *Principia Matemática* de Russell y Whitehead. Se dice que Russell se manifestó complacido cuando Simon le mostró que la demostración de un teorema que el programa había generado era más corta que la que aparecía en *Principia*. Los editores de la revista *Journal of Symbolic Logic* resultaron menos impresionados y rechazaron un artículo cuyos autores eran Newell, Simon y el Teórico Lógico (TL).

El taller de Dartmouth no produjo ningún avance notable, pero puso en contacto a las figuras importantes de este campo. Durante los siguientes 20 años, el campo estuvo dominado por estos personajes, así como por sus estudiantes y colegas del MIT, CMU, Stanford e IBM. Quizá lo último que surgió del taller fue el consenso en adoptar el nuevo nombre propuesto por McCarthy para este campo: **Inteligencia Artificial**. Quizá «racionalidad computacional» hubiese sido más adecuado, pero «IA» se ha mantenido.

Revisando la propuesta del taller de Dartmouth (McCarthy *et al*., 1955), se puede apreciar por qué fue necesario para la IA convertirse en un campo separado. ¿Por qué

[13] Actualmente Universidad Carnegie Mellon (UCM).

[14] Newell y Simon también desarrollaron un lenguaje de procesamiento de listas, IPL, para poder escribir el TL. No disponían de un compilador y lo tradujeron a código máquina a mano. Para evitar errores, trabajaron en paralelo, diciendo en voz alta números binarios, conforme escribían cada instrucción para asegurarse de que ambos coincidían.

no todo el trabajo hecho en el campo de la IA se ha realizado bajo el nombre de teoría de control, o investigación operativa, o teoría de la decisión, que, después de todo, persiguen objetivos similares a los de la IA? O, ¿por qué no es la IA una rama de las matemáticas? La primera respuesta es que la IA desde el primer momento abarcó la idea de duplicar facultades humanas como la creatividad, la auto-mejora y el uso del lenguaje. Ninguno de los otros campos tenían en cuenta esos temas. La segunda respuesta está relacionada con la metodología. La IA es el único de estos campos que es claramente una rama de la informática (aunque la investigación operativa comparte el énfasis en la simulación por computador), además la IA es el único campo que persigue la construcción de máquinas que funcionen automáticamente en medios complejos y cambiantes.

Entusiasmo inicial, grandes esperanzas (1952–1969)

Los primeros años de la IA estuvieron llenos de éxitos (aunque con ciertas limitaciones). Teniendo en cuenta lo primitivo de los computadores y las herramientas de programación de aquella época, y el hecho de que sólo unos pocos años antes, a los computadores se les consideraba como artefactos que podían realizar trabajos aritméticos y nada más, resultó sorprendente que un computador hiciese algo remotamente inteligente. La comunidad científica, en su mayoría, prefirió creer que «una máquina nunca podría hacer *tareas*» (*véase* el Capítulo 26 donde aparece una extensa lista de *tareas* recopilada por Turing). Naturalmente, los investigadores de IA responderían demostrando la realización de una *tarea* tras otra. John McCarthy se refiere a esta época como la era de «¡Mira, mamá, ahora sin manos!».

Al temprano éxito de Newell y Simon siguió el del sistema de resolución general de problemas, o SRGP. A diferencia del Teórico Lógico, desde un principio este programa se diseñó para que imitara protocolos de resolución de problemas de los seres humanos. Dentro del limitado número de puzles que podía manejar, resultó que la secuencia en la que el programa consideraba que los subobjetivos y las posibles acciones eran semejantes a la manera en que los seres humanos abordaban los mismos problemas. Es decir, el SRGP posiblemente fue el primer programa que incorporó el enfoque de «pensar como un ser humano». El éxito del SRGP y de los programas que le siguieron, como los modelos de cognición, llevaron a Newell y Simon (1976) a formular la famosa hipótesis del **sistema de símbolos físicos**, que afirma que «un sistema de símbolos físicos tiene los medios suficientes y necesarios para generar una acción inteligente». Lo que ellos querían decir es que cualquier sistema (humano o máquina) que exhibiese inteligencia debería operar manipulando estructuras de datos compuestas por símbolos. Posteriormente se verá que esta hipótesis se ha modificado atendiendo a distintos puntos de vista.

SISTEMA DE SÍMBOLOS FÍSICOS

En IBM, Nathaniel Rochester y sus colegas desarrollaron algunos de los primeros programas de IA. Herbert Gelernter (1959) construyó el demostrador de teoremas de geometría (DTG), el cual era capaz de probar teoremas que muchos estudiantes de matemáticas podían encontrar muy complejos de resolver. A comienzos 1952, Arthur Samuel escribió una serie de programas para el juego de las damas que eventualmente aprendieron a jugar hasta alcanzar un nivel equivalente al de un *amateur*. De paso, echó por

tierra la idea de que los computadores sólo pueden hacer lo que se les dice: su programa pronto aprendió a jugar mejor que su creador. El programa se presentó en la televisión en febrero de 1956 y causó una gran impresión. Como Turing, Samuel tenía dificultades para obtener el tiempo de cómputo. Trabajaba por las noches y utilizaba máquinas que aún estaban en período de prueba en la planta de fabricación de IBM. El Capítulo 6 trata el tema de los juegos, y en el Capítulo 21 se describe con detalle las técnicas de aprendizaje utilizadas por Samuel.

John McCarthy se trasladó de Darmouth al MIT, donde realizó tres contribuciones cruciales en un año histórico: 1958. En el Laboratorio de IA del MIT Memo Número 1, McCarthy definió el lenguaje de alto nivel **Lisp**, que se convertiría en el lenguaje de programación dominante en la IA. Lisp es el segundo lenguaje de programación más antiguo que se utiliza en la actualidad, ya que apareció un año después de FORTRAN. Con Lisp, McCarthy tenía la herramienta que necesitaba, pero el acceso a los escasos y costosos recursos de cómputo aún era un problema serio. Para solucionarlo, él, junto a otros miembros del MIT, inventaron el tiempo compartido. También, en 1958, McCarthy publicó un artículo titulado *Programs with Common Sense*, en el que describía el Generador de Consejos, un programa hipotético que podría considerarse como el primer sistema de IA completo. Al igual que el Teórico Lógico y el Demostrador de Teoremas de Geometría, McCarthy diseñó su programa para buscar la solución a problemas utilizando el conocimiento. Pero, a diferencia de los otros, manejaba el conocimiento general del mundo. Por ejemplo, mostró cómo algunos axiomas sencillos permitían a un programa generar un plan para conducirnos hasta el aeropuerto y tomar un avión. El programa se diseñó para que aceptase nuevos axiomas durante el curso normal de operación, permitiéndole así ser competente en áreas nuevas, sin *necesidad de reprogramación*. El Generador de Consejos incorporaba así los principios centrales de la representación del conocimiento y el razonamiento: es útil contar con una representación formal y explícita del mundo y de la forma en que la acción de un agente afecta al mundo, así como, ser capaces de manipular estas representaciones con procesos deductivos. Es sorprendente constatar cómo mucho de lo propuesto en el artículo escrito en 1958 permanece vigente incluso en la actualidad.

1958 fue el año en el que Marvin Minsky se trasladó al MIT. Sin embargo, su colaboración inicial no duró demasiado. McCarthy se centró en la representación y el razonamiento con lógica formal, mientras que Minsky estaba más interesado en lograr que los programas funcionaran y eventualmente desarrolló un punto de vista anti-lógico. En 1963 McCarthy creó el Laboratorio de IA en Stanford. Su plan para construir la versión más reciente del Generador de Consejos con ayuda de la lógica sufrió un considerable impulso gracias al descubrimiento de J. A. Robinson del método de resolución (un algoritmo completo para la demostración de teoremas para la lógica de primer orden; *véase* el Capítulo 9). El trabajo realizado en Stanford hacía énfasis en los métodos de propósito general para el razonamiento lógico. Algunas aplicaciones de la lógica incluían los sistemas de planificación y respuesta a preguntas de Cordell Green (1969b), así como el proyecto de robótica de Shakey en el nuevo Instituto de Investigación de Stanford (Stanford Research Institute, SRI). Este último proyecto, comentado en detalle en el Capítulo 25, fue el primero que demostró la total integración del razonamiento lógico y la actividad física.

Minsky supervisó el trabajo de una serie de estudiantes que eligieron un número de problemas limitados cuya solución pareció requerir inteligencia. Estos dominios limi-

MICROMUNDOS

tados se conocen como **micromundos**. El programa SAINT de James Slagle (1963a) fue capaz de resolver problemas de integración de cálculo en forma cerrada, habituales en los primeros cursos de licenciatura. El programa ANALOGY de Tom Evans (1968) resolvía problemas de analogía geométrica que se aplicaban en las pruebas de medición de inteligencia, semejante al de la Figura 1.4. El programa STUDENT de Daniel Bobrow (1967) podía resolver problemas de álgebra del tipo:

> Si el número de clientes de Tom es dos veces el cuadrado del 20 por ciento de la cantidad de anuncios que realiza, y éstos ascienden a 45, ¿cuántos clientes tiene Tom?

El micromundo más famoso fue el mundo de los bloques, que consiste en un conjunto de bloques sólidos colocados sobre una mesa (más frecuentemente, en la simulación de ésta), como se muestra en la Figura 1.5. Una tarea típica de este mundo es la reordenación de los bloques de cierta manera, con la ayuda de la mano de un robot que es capaz de tomar un bloque cada vez. El mundo de los bloques fue el punto de partida para el proyecto de visión de David Huffman (1971), la visión y el trabajo de propagación con restricciones de David Waltz (1975), la teoría del aprendizaje de Patrick Winston (1970), del programa para la comprensión de lenguaje natural de Terry Winograd (1972) y del planificador de Scott Fahlman (1974).

El trabajo realizado por McCulloch y Pitts con redes neuronales hizo florecer esta área. El trabajo de Winograd y Cowan (1963) mostró cómo un gran número de elementos podría representar un concepto individual de forma colectiva, lo cual llevaba consigo un aumento proporcional en robustez y paralelismo. Los métodos de aprendizaje de Hebb se reforzaron con las aportaciones de Bernie Widrow (Widrow y Hoff, 1960; Widrow, 1962), quien llamó **adalines** a sus redes, y por Frank Rosenblatt (1962) con sus **perceptrones**. Rosenblatt demostró el famoso teorema del perceptrón, con lo que mostró que su algoritmo de aprendizaje podría ajustar las intensidades de las conexiones de un perceptrón para que se adaptaran a los datos de entrada, siempre y cuando existiera una correspondencia. Estos temas se explicarán en el Capítulo 20.

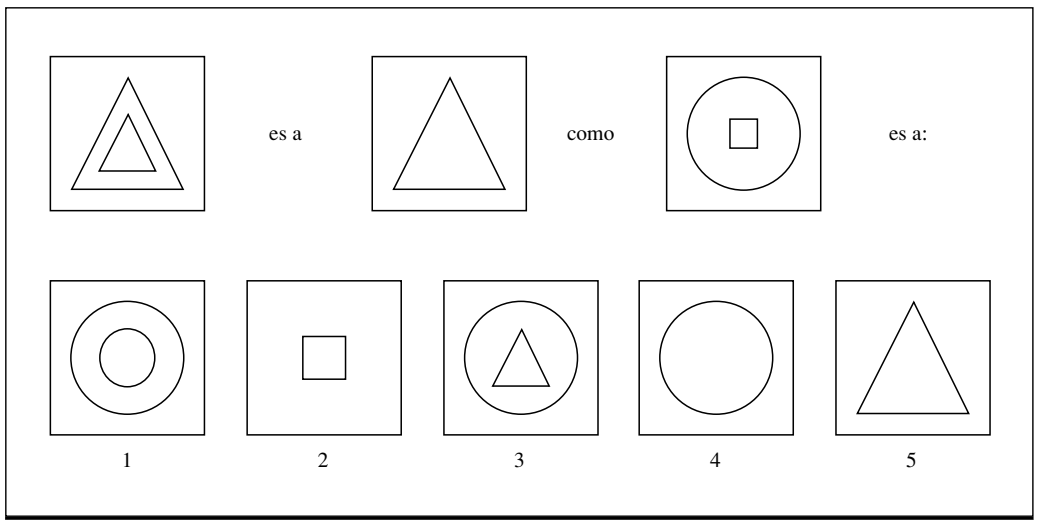

Figura 1.4 Ejemplo de un problema resuelto por el programa ANALOGY de Evans.

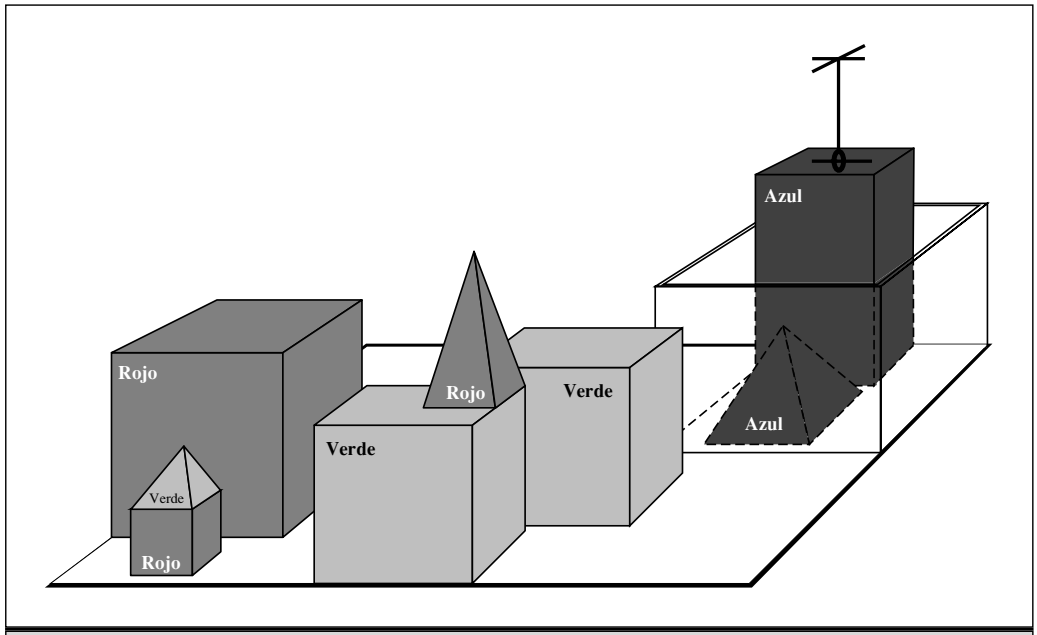

Figura 1.5 Esquema del mundo de los bloques. SHRDLU (Winograd, 1972) ha completado su tarea, «Busca un bloque que sea más alto que el que ahora tienes y ponlo en la caja».

Una dosis de realidad (1966–1973)

Desde el principio, los investigadores de IA hicieron públicas, sin timidez, predicciones sobre el éxito que les esperaba. Con frecuencia, se cita el siguiente comentario realizado por Herbert Simon en 1957:

> Sin afán de sorprenderlos y dejarlos atónitos, pero la forma más sencilla que tengo de resumirlo es diciéndoles que actualmente en el mundo existen máquinas capaces de pensar, aprender y crear. Además, su aptitud para hacer lo anterior aumentará rápidamente hasta que (en un futuro previsible) la magnitud de problemas que serán capaces de resolver irá a la par que la capacidad de la mente humana para hacer lo mismo.

Términos como «futuro previsible» pueden interpretarse de formas distintas, pero Simon también hizo predicciones más concretas: como que en diez años un computador llegaría a ser campeón de ajedrez, y que se podría demostrar un importante teorema matemático con una máquina. Estas predicciones se cumplirían (al menos en parte) dentro de los siguientes 40 años y no en diez. El exceso de confianza de Simon se debió a la prometedora actuación de los primeros sistemas de IA en problemas simples. En la mayor parte de los casos resultó que estos primeros sistemas fallaron estrepitosamente cuando se utilizaron en problemas más variados o de mayor dificultad.

El primer tipo de problemas surgió porque la mayoría de los primeros programas contaban con poco o ningún conocimiento de las materia objeto de estudio; obtenían resultados gracias a sencillas manipulaciones sintácticas. Una anécdota típica tuvo lugar cuando se comenzaba a trabajar en la traducción automática, actividad que recibía un

generoso patrocinio del Consejo Nacional para la Investigación de Estados Unidos en un intento de agilizar la traducción de artículos científicos rusos en vísperas del lanzamiento del Sputnik en 1957. Al principio se consideró que todo se reduciría a sencillas transformaciones sintácticas apoyadas en las gramáticas rusas e inglesa y al emplazamiento de palabras mediante un diccionario electrónico, lo que bastaría para obtener el significado exacto de las oraciones. La realidad es que para traducir es necesario contar con un conocimiento general sobre el tema, que permita resolver ambigüedades y así, precisar el contenido de una oración. La famosa retraducción del ruso al inglés de la frase «el espíritu es fuerte pero la carne es débil», cuyo resultado fue «el vodka es bueno pero la carne está podrida» es un buen ejemplo del tipo de dificultades que surgieron. En un informe presentado en 1966, el comité consultivo declaró que «no se ha logrado obtener ninguna traducción de textos científicos generales ni se prevé obtener ninguna en un futuro inmediato». Se canceló todo el patrocinio del gobierno estadounidense que se había asignado a los proyectos académicos sobre traducción. Hoy día, la traducción automática es una herramienta imperfecta pero de uso extendido en documentos técnicos, comerciales, gubernamentales y de Internet.

El segundo problema fue que muchos de los problemas que se estaban intentando resolver mediante la IA eran intratables. La mayoría de los primeros programas de IA resolvían problemas experimentando con diversos pasos hasta que se llegara a encontrar una solución. Esto funcionó en los primeros programas debido a que los micromundos con los que se trabajaba contenían muy pocos objetos, y por lo tanto muy pocas acciones posibles y secuencias de soluciones muy cortas. Antes de que se desarrollara la teoría de la complejidad computacional, se creía que para «aumentar» el tamaño de los programas de forma que estos pudiesen solucionar grandes problemas sería necesario incrementar la velocidad del *hardware* y aumentar las memorias. El optimismo que acompañó al logro de la demostración de problemas, por ejemplo, pronto se vio eclipsado cuando los investigadores fracasaron en la demostración de teoremas que implicaban más de unas pocas decenas de condiciones. *El hecho de que, en principio, un programa sea capaz de encontrar una solución no implica que tal programa encierre todos los mecanismos necesarios para encontrar la solución en la práctica.*

EVOLUCIÓN AUTOMÁTICA

La ilusoria noción de una ilimitada capacidad de cómputo no sólo existió en los programas para la resolución de problemas. Los primeros experimentos en el campo de la **evolución automática** (ahora llamados **algoritmos genéticos**) (Friedberg, 1958; Friedberg *et al*., 1959) estaban basados en la, sin duda correcta, premisa de que efectuando una adecuada serie de pequeñas mutaciones a un programa de código máquina se podría generar un programa con buen rendimiento aplicable en cualquier tarea sencilla. Después surgió la idea de probar con mutaciones aleatorias aplicando un proceso de selección con el fin de conservar aquellas mutaciones que hubiesen demostrado ser más útiles. No obstante, las miles de horas de CPU dedicadas, no dieron lugar a ningún avance tangible. Los algoritmos genéticos actuales utilizan representaciones mejores y han tenido más éxito.

La incapacidad para manejar la «explosión combinatoria» fue una de las principales críticas que se hicieron a la IA en el informe de Lighthill (Lighthill, 1973), informe en el que se basó la decisión del gobierno británico para retirar la ayuda a las investigaciones sobre IA, excepto en dos universidades. (La tradición oral presenta un cuadro un

poco distinto y más animado, en el que se vislumbran ambiciones políticas y animadversiones personales, cuya descripción está fuera del ámbito de esta obra.)

El tercer obstáculo se derivó de las limitaciones inherentes a las estructuras básicas que se utilizaban en la generación de la conducta inteligente. Por ejemplo, en 1969, en el libro de Minsky y Papert, *Perceptrons*, se demostró que si bien era posible lograr que los perceptrones (una red neuronal simple) aprendieran cualquier cosa que pudiesen representar, su capacidad de representación era muy limitada. En particular, un perceptrón con dos entradas no se podía entrenar para que aprendiese a reconocer cuándo sus dos entradas eran diferentes. Si bien los resultados que obtuvieron no eran aplicables a redes más complejas multicapa, los fondos para la investigación de las redes neuronales se redujeron a prácticamente nada. Es irónico que los nuevos algoritmos de aprendizaje de retroalimentación utilizados en las redes multicapa y que fueron la causa del gran resurgimiento de la investigación en redes neuronales de finales de los años 80, en realidad, se hayan descubierto por primera vez en 1969 (Bryson y Ho, 1969).

Sistemas basados en el conocimiento: ¿clave del poder? (1969–1979)

El cuadro que dibujaba la resolución de problemas durante la primera década de la investigación en la IA estaba centrado en el desarrollo de mecanismos de búsqueda de propósito general, en los que se entrelazaban elementos de razonamiento básicos para encontrar así soluciones completas. A estos procedimientos se les ha denominado **métodos débiles**, debido a que no tratan problemas más amplios o más complejos. La alternativa a los métodos débiles es el uso de conocimiento específico del dominio que facilita el desarrollo de etapas de razonamiento más largas, pudiéndose así resolver casos recurrentes en dominios de conocimiento restringido. Podría afirmarse que para resolver un problema en la práctica, es necesario saber de antemano la correspondiente respuesta.

MÉTODOS DÉBILES

El programa DENDRAL (Buchanan *et al*., 1969) constituye uno de los primeros ejemplos de este enfoque. Fue diseñado en Stanford, donde Ed Feigenbaum (discípulo de Herbert Simon), Bruce Buchanan (filósofo convertido en informático) y Joshua Lederberg (genetista ganador del Premio Nobel) colaboraron en la solución del problema de inferir una estructura molecular a partir de la información proporcionada por un espectrómetro de masas. El programa se alimentaba con la fórmula elemental de la molécula (por ejemplo, $C_6H_{13}NO_2$) y el espectro de masas, proporcionando las masas de los distintos fragmentos de la molécula generada después de ser bombardeada con un haz de electrones. Por ejemplo, un espectro de masas con un pico en $m = 15$, correspondería a la masa de un fragmento de metilo (CH_3).

La versión más simple del programa generaba todas las posibles estructuras que correspondieran a la fórmula, luego predecía el espectro de masas que se observaría en cada caso, y comparaba éstos con el espectro real. Como era de esperar, el método anterior resultó pronto inviable para el caso de moléculas con un tamaño considerable. Los creadores de DENDRAL consultaron con químicos analíticos y se dieron cuenta de que éstos trabajaban buscando patrones conocidos de picos en el espectro que sugerían estructuras comunes en la molécula. Por ejemplo, par identificar el subgrupo (con un peso de 28) de las cetonas (C=0) se empleó la siguiente regla:

si hay dos picos en x_1 y x_2 tales que
a) $x_1 + x_2 = M + 28$ (siendo M la masa de toda la molécula);
b) $x_1 - 28$ es un pico alto;
c) $x_2 - 28$ es un pico alto;
d) al menos una de x_1 y x_2 es alta.
entonces existe un subgrupo de cetonas

Al reconocer que la molécula contiene una subestructura concreta se reduce el número de posibles candidatos de forma considerable. La potencia de DENDRAL se basaba en que:

> Toda la información teórica necesaria para resolver estos problemas se ha proyectado desde su forma general [componente predicho por el espectro] («primeros principios») a formas eficientes especiales («recetas de cocina»). (Feigenbaum *et al*., 1971)

La trascendencia de DENDRAL se debió a ser el primer sistema de *conocimiento intenso* que tuvo éxito: su base de conocimiento estaba formada por grandes cantidades de reglas de propósito particular. En sistemas diseñados posteriormente se incorporaron también los elementos fundamentales de la propuesta de McCarthy para el Generador de Consejos, la nítida separación del conocimiento (en forma de reglas) de la parte correspondiente al razonamiento.

Teniendo en cuenta esta lección, Feigenbaum junto con otros investigadores de Stanford dieron comienzo al Proyecto de Programación Heurística, PPH, dedicado a determinar el grado con el que la nueva metodología de los **sistemas expertos** podía aplicarse a otras áreas de la actividad humana. El siguiente gran esfuerzo se realizó en el área del diagnóstico médico. Feigenbaum, Buchanan y el doctor Edward Shortliffe diseñaron el programa MYCIN, para el diagnóstico de infecciones sanguíneas. Con 450 reglas aproximadamente, MYCIN era capaz de hacer diagnósticos tan buenos como los de un experto y, desde luego, mejores que los de un médico recién graduado. Se distinguía de DENDRAL en dos aspectos principalmente. En primer lugar, a diferencia de las reglas de DENDRAL, no se contaba con un modelo teórico desde el cual se pudiesen deducir las reglas de MYCIN. Fue necesario obtenerlas a partir de extensas entrevistas con los expertos, quienes las habían obtenido de libros de texto, de otros expertos o de su experiencia directa en casos prácticos. En segundo lugar, las reglas deberían reflejar la incertidumbre inherente al conocimiento médico. MYCIN contaba con un elemento que facilitaba el cálculo de incertidumbre denominado **factores de certeza** (*véase* el Capítulo 13), que al parecer (en aquella época) correspondía muy bien a la manera como los médicos ponderaban las evidencias al hacer un diagnóstico.

SISTEMAS EXPERTOS

La importancia del conocimiento del dominio se demostró también en el área de la comprensión del lenguaje natural. Aunque el sistema SHRDLU de Winograd para la comprensión del lenguaje natural había suscitado mucho entusiasmo, su dependencia del análisis sintáctico provocó algunos de los mismos problemas que habían aparecido en los trabajos realizados en la traducción automática. Era capaz de resolver los problemas de ambigüedad e identificar los pronombres utilizados, gracias a que se había diseñado especialmente para un área (el mundo de los bloques). Fueron varios los investigadores que, como Eugene Charniak, estudiante de Winograd en el MIT, opinaron que para una sólida comprensión del lenguaje era necesario contar con un conocimiento general sobre el mundo y un método general para usar ese conocimiento.

En Yale, el lingüista transformado en informático Roger Schank reforzó lo anterior al afirmar: «No existe eso que llaman sintaxis», lo que irritó a muchos lingüistas, pero sirvió para iniciar un útil debate. Schank y sus estudiantes diseñaron una serie de programas (Schank y Abelson, 1977; Wilensky, 1978; Schank y Riesbeck, 1981; Dyer, 1983) cuyo objetivo era la comprensión del lenguaje natural. El foco de atención estaba menos en el lenguaje *per se* y más en los problemas vinculados a la representación y razonamiento del conocimiento necesario para la comprensión del lenguaje. Entre los problemas estaba el de la representación de situaciones estereotipo (Cullingford, 1981), la descripción de la organización de la memoria humana (Rieger, 1976; Kolodner, 1983) y la comprensión de planes y objetivos (Wilensky, 1983).

El crecimiento generalizado de aplicaciones para solucionar problemas del mundo real provocó el respectivo aumento en la demanda de esquemas de representación del conocimiento que funcionaran. Se desarrolló una considerable cantidad de lenguajes de representación y razonamiento diferentes. Algunos basados en la lógica, por ejemplo el lenguaje Prolog gozó de mucha aceptación en Europa, aceptación que en Estados Unidos fue para la familia del PLANNER. Otros, siguiendo la noción de **marcos** de Minsky (1975), se decidieron por un enfoque más estructurado, al recopilar información sobre objetos concretos y tipos de eventos, organizando estos tipos en grandes jerarquías taxonómicas, similares a las biológicas.

MARCOS

La IA se convierte en una industria (desde 1980 hasta el presente)

El primer sistema experto comercial que tuvo éxito, R1, inició su actividad en Digital Equipment Corporation (McDermott, 1982). El programa se utilizaba en la elaboración de pedidos de nuevos sistemas informáticos. En 1986 representaba para la compañía un ahorro estimado de 40 millones de dólares al año. En 1988, el grupo de Inteligencia Artificial de DEC había distribuido ya 40 sistemas expertos, y había más en camino. Du Pont utilizaba ya 100 y estaban en etapa de desarrollo 500 más, lo que le generaba ahorro de diez millones de dólares anuales aproximadamente. Casi todas las compañías importantes de Estados Unidos contaban con su propio grupo de IA, en el que se utilizaban o investigaban sistemas expertos.

En 1981 los japoneses anunciaron el proyecto «Quinta Generación», un plan de diez años para construir computadores inteligentes en los que pudiese ejecutarse Prolog. Como respuesta Estados Unidos constituyó la Microelectronics and Computer Technology Corporation (MCC), consorcio encargado de mantener la competitividad nacional en estas áreas. En ambos casos, la IA formaba parte de un gran proyecto que incluía el diseño de chips y la investigación de la relación hombre máquina. Sin embargo, los componentes de IA generados en el marco de MCC y del proyecto Quinta Generación nunca alcanzaron sus objetivos. En el Reino Unido, el informe Alvey restauró el patrocinio suspendido por el informe Lighthill[15].

[15] Para evitar confusiones, se creó un nuevo campo denominado Sistemas Inteligentes Basados en Conocimiento (IKBS, *Intelligent Knowledge-Based Systems*) ya que la Inteligencia Artificial había sido oficialmente cancelada.

En su conjunto, la industria de la IA creció rápidamente, pasando de unos pocos millones de dólares en 1980 a billones de dólares en 1988. Poco después de este período llegó la época llamada «El Invierno de la IA», que afectó a muchas empresas que no fueron capaces de desarrollar los extravagantes productos prometidos.

Regreso de las redes neuronales (desde 1986 hasta el presente)

Aunque la informática había abandonado de manera general el campo de las redes neuronales a finales de los años 70, el trabajo continuó en otros campos. Físicos como John Hopfield (1982) utilizaron técnicas de la mecánica estadística para analizar las propiedades de almacenamiento y optimización de las redes, tratando colecciones de nodos como colecciones de átomos. Psicólogos como David Rumelhart y Geoff Hinton continuaron con el estudio de modelos de memoria basados en redes neuronales. Como se verá en el Capítulo 20, el impulso más fuerte se produjo a mediados de la década de los 80, cuando por lo menos cuatro grupos distintos reinventaron el algoritmo de aprendizaje de retroalimentación, mencionado por vez primera en 1969 por Bryson y Ho. El algoritmo se aplicó a diversos problemas de aprendizaje en los campos de la informática y la psicología, y la gran difusión que conocieron los resultados obtenidos, publicados en la colección *Parallel Distributed Processing* (Rumelhart y McClelland, 1986), suscitó gran entusiasmo.

CONEXIONISTAS

Aquellos modelos de inteligencia artificial llamados **conexionistas**[16] fueron vistos por algunos como competidores tanto de los modelos simbólicos propuestos por Newell y Simon como de la aproximación lógica de McCarthy entre otros (Smolensky, 1988). Puede parecer obvio que los humanos manipulan símbolos hasta cierto nivel, de hecho, el libro *The Symbolic Species* (1997) de Terrence Deacon sugiere que esta es la *característica que define* a los humanos, pero los conexionistas más ardientes se preguntan si la manipulación de los símbolos desempeña algún papel justificable en determinados modelos de cognición. Este interrogante no ha sido aún clarificado, pero la tendencia actual es que las aproximaciones conexionistas y simbólicas son complementarias y no competidoras.

IA se convierte en una ciencia (desde 1987 hasta el presente)

En los últimos años se ha producido una revolución tanto en el contenido como en la metodología de trabajo en el campo de la inteligencia artificial.[17] Actualmente es más usual el desarrollo sobre teorías ya existentes que proponer teorías totalmente novedo-

[16] Se usa la traducción literal del término *connectionist* por no existir un término equivalente en español (*N. del RT*).

[17] Hay quien ha caracterizado este cambio como la victoria de los pulcros (aquellos que consideran que las teorías de IA deben basarse rigurosamente en las matemáticas) sobre los desaliñados (aquellos que después de intentar muchas ideas, escriben algunos programas y después evalúan las que aparentemente funcionan). Ambos enfoques son útiles. Esta tendencia en favor de una mayor pulcritud es señal de que el campo ha alcanzado cierto nivel de estabilidad y madurez. Lo cual no implica que tal estabilidad se puede ver alterada con el surgimiento de otras ideas poco estructuradas.

sas, tomar como base rigurosos teoremas o sólidas evidencias experimentales más que intuición, y demostrar la utilidad de las aplicaciones en el mundo real más que crear ejemplos de juguete.

La IA se fundó en parte en el marco de una rebelión en contra de las limitaciones de los campos existentes como la teoría de control o la estadística, y ahora abarca estos campos. Tal y como indica David McAllester (1998),

> En los primeros años de la IA parecía perfectamente posible que las nuevas formas de la computación simbólica, por ejemplo, los marcos y las redes semánticas, hicieran que la mayor parte de la teoría clásica pasara a ser obsoleta. Esto llevó a la IA a una especie de aislamiento, que la separó del resto de las ciencias de la computación. En la actualidad se está abandonando este aislamiento. Existe la creencia de que el aprendizaje automático no se debe separar de la teoría de la información, que el razonamiento incierto no se debe separar de los modelos estocásticos, de que la búsqueda no se debe aislar de la optimización clásica y el control, y de que el razonamiento automático no se debe separar de los métodos formales y del análisis estático.

En términos metodológicos, se puede decir, con rotundidad, que la IA ya forma parte del ámbito de los métodos científicos. Para que se acepten, las hipótesis se deben someter a rigurosos experimentos empíricos, y los resultados deben analizarse estadísticamente para identificar su relevancia (Cohen, 1995). El uso de Internet y el compartir repositorios de datos de prueba y código, ha hecho posible que ahora se puedan contrastar experimentos.

Un buen modelo de la tendencia actual es el campo del reconocimiento del habla. En la década de los 70 se sometió a prueba una gran variedad de arquitecturas y enfoques. Muchos de ellos fueron un tanto *ad hoc* y resultaban frágiles, y fueron probados sólo en unos pocos ejemplos elegidos especialmente. En años recientes, las aproximaciones basadas en los **modelos de Markov ocultos**, MMO, han pasado a dominar el área. Dos son las características de los MMO que tienen relevancia. Primero, se basan en una rigurosa teoría matemática, lo cual ha permitido a los investigadores del lenguaje basarse en los resultados de investigaciones matemáticas hechas en otros campos a lo largo de varias décadas. En segundo lugar, los modelos se han generado mediante un proceso de aprendizaje en grandes *corpus* de datos de lenguaje reales. Lo cual garantiza una funcionalidad robusta, y en sucesivas pruebas ciegas, los MMO han mejorado sus resultados a un ritmo constante. La tecnología del habla y el campo relacionado del reconocimiento de caracteres manuscritos están actualmente en transición hacia una generalizada utilización en aplicaciones industriales y de consumo.

Las redes neuronales también siguen esta tendencia. La mayor parte del trabajo realizado con redes neuronales en la década de los 80 se realizó con la idea de dejar a un lado lo que se podía hacer y de descubrir en qué se diferenciaban las redes neuronales de otras técnicas «tradicionales». La utilización de metodologías mejoradas y marcos teóricos, ha autorizado que este campo alcance un grado de conocimiento que ha permitido que ahora las redes neuronales se puedan comparar con otras técnicas similares de campos como la estadística, el reconocimiento de patrones y el aprendizaje automático, de forma que las técnicas más prometedoras pueden aplicarse a cualquier problema. Como resultado de estos desarrollos, la tecnología denominada **minería de datos** ha generado una nueva y vigorosa industria.

MINERÍA DE DATOS

La aparición de *Probabilistic Reasoning in Intelligent Systems* de Judea Pearl (1988) hizo que se aceptara de nuevo la probabilidad y la teoría de la decisión como parte de la IA, como consecuencia del resurgimiento del interés despertado y gracias especialmente al artículo *In Defense of Probability* de Peter Cheeseman (1985). El formalismo de las **redes de Bayes** apareció para facilitar la representación eficiente y el razonamiento riguroso en situaciones en las que se disponía de conocimiento incierto. Este enfoque supera con creces muchos de los problemas de los sistemas de razonamiento probabilístico de las décadas de los 60 y 70; y ahora domina la investigación de la IA en el razonamiento incierto y los sistemas expertos. Esta aproximación facilita el aprendizaje a partir de la experiencia, y combina lo mejor de la IA clásica y las redes neuronales. El trabajo de Judea Pearl (1982a) y de Eric Horvitz y David Heckerman (Horvitz y Heckerman, 1986; Horvitz *et al*., 1986) sirvió para promover la noción de sistemas expertos *normativos*: es decir, los que actúan racionalmente de acuerdo con las leyes de la teoría de la decisión, sin que intenten imitar las etapas de razonamiento de los expertos humanos. El sistema operativo Windows™ incluye varios sistemas expertos de diagnóstico normativos para la corrección de problemas. En los Capítulos 13 y 16 se aborda este tema.

Revoluciones similares y suaves se han dado en robótica, visión por computador, y aprendizaje automático. La comprensión mejor de los problemas y de su complejidad, junto a un incremento en la sofisticación de las matemáticas ha facilitado el desarrollo de una agenda de investigación y de métodos más robustos. En cualquier caso, la formalización y especialización ha llevado también a la fragmentación: áreas como la visión y la robótica están cada vez más aislados de la «rama central» de la IA. La concepción unificadora de IA como diseño de agentes racionales puede facilitar la unificación de estos campos diferentes.

Emergencia de los sistemas inteligentes (desde 1995 hasta el presente)

Quizás animados por el progreso en la resolución de subproblemas de IA, los investigadores han comenzado a trabajar de nuevo en el problema del «agente total». El trabajo de Allen Newell, John Laird, y Paul Rosenbloom en SOAR (Newell, 1990; Laird *et al*., 1987) es el ejemplo mejor conocido de una arquitectura de agente completa. El llamado «movimiento situado» intenta entender la forma de actuar de los agentes inmersos en entornos reales, que disponen de sensores de entradas continuas. Uno de los medios más importantes para los agentes inteligentes es Internet. Los sistemas de IA han llegado a ser tan comunes en aplicaciones desarrolladas para la Web que el sufijo «-bot» se ha introducido en el lenguaje común. Más aún, tecnologías de IA son la base de muchas herramientas para Internet, como por ejemplo motores de búsqueda, sistemas de recomendación, y los sistemas para la construcción de portales Web.

Además de la primera edición de este libro de texto (Russell y Norvig, 1995), otros libros de texto han adoptado recientemente la perspectiva de agentes (Poole *et al*., 1998; Nilsson, 1998). Una de las conclusiones que se han extraído al tratar de construir agentes completos ha sido que se deberían reorganizar los subcampos aislados de la IA para

que sus resultados se puedan interrelacionar. En particular, ahora se cree mayoritariamente que los sistemas sensoriales (visión, sónar, reconocimiento del habla, etc.) no pueden generar información totalmente fidedigna del medio en el que habitan. Otra segunda consecuencia importante, desde la perspectiva del agente, es que la IA se ha ido acercando a otros campos, como la teoría de control y la economía, que también tratan con agentes.

1.4 El estado del arte

¿Qué es capaz de hacer la IA hoy en día? Responder de manera concisa es difícil porque hay muchas actividades divididas en muchos subcampos. Aquí se presentan unas cuantas aplicaciones; otras aparecerán a lo largo del texto.

Planificación autónoma: a un centenar de millones de millas de la Tierra, el programa de la NASA Agente Remoto se convirtió en el primer programa de planificación autónoma a bordo que controlaba la planificación de las operaciones de una nave espacial desde abordo (Jonsson *et al*., 2000). El Agente Remoto generaba planes a partir de objetivos generales especificados desde tierra, y monitorizaba las operaciones de la nave espacial según se ejecutaban los planes (detección, diagnóstico y recuperación de problemas según ocurrían).

Juegos: Deep Blue de IBM fue el primer sistema que derrotó a un campeón mundial en una partida de ajedrez cuando superó a Garry Kasparov por un resultado de 3.5 a 2.5 en una partida de exhibición (Goodman y Keene, 1997). Kasparov dijo que había percibido un «nuevo tipo de inteligencia» al otro lado del tablero. La revista *Newsweek* describió la partida como «La partida final». El valor de las acciones de IBM se incrementó en 18 billones de dólares.

Control autónomo: el sistema de visión por computador ALVINN fue entrenado para dirigir un coche de forma que siguiese una línea. Se instaló en una furgoneta controlada por computador en el NAVLAB de UCM y se utilizó para dirigir al vehículo por Estados Unidos. Durante 2.850 millas controló la dirección del vehículo en el 98 por ciento del trayecto. Una persona lo sustituyó en el dos por ciento restante, principalmente en vías de salida. El NAVLAB posee videocámaras que transmiten imágenes de la carretera a ALVINN, que posteriormente calcula la mejor dirección a seguir, basándose en las experiencias acumuladas en los viajes de entrenamiento.

Diagnosis: los programas de diagnóstico médico basados en el análisis probabilista han llegado a alcanzar niveles similares a los de médicos expertos en algunas áreas de la medicina. Heckerman (1991) describe un caso en el que un destacado experto en la patología de los nodos linfáticos se mofó del diagnóstico generado por un programa en un caso especialmente difícil. El creador del programa le sugirió que le preguntase al computador cómo había generado el diagnóstico. La máquina indicó los factores más importantes en los que había basado su decisión y explicó la ligera interacción existente entre varios de los síntomas en este caso. Eventualmente, el experto aceptó el diagnóstico del programa.

Planificación logística: durante la crisis del Golfo Pérsico de 1991, las fuerzas de Estados Unidos desarrollaron la herramienta *Dynamic Analysis and Replanning Tool*

(DART) (Cross y Walker, 1994), para automatizar la planificación y organización logística del transporte. Lo que incluía hasta 50.000 vehículos, carga y personal a la vez, teniendo en cuenta puntos de partida, destinos, rutas y la resolución de conflictos entre otros parámetros. Las técnicas de planificación de IA permitieron que se generara un plan en cuestión de horas que podría haber llevado semanas con otros métodos. La agencia DARPA (*Defense Advanced Research Project Agency*) afirmó que esta aplicación por sí sola había más que amortizado los 30 años de inversión de DARPA en IA.

Robótica: muchos cirujanos utilizan hoy en día asistentes robot en operaciones de microcirugía. HipNav (DiGioia *et al*., 1996) es un sistema que utiliza técnicas de visión por computador para crear un modelo tridimensional de la anatomía interna del paciente y después utiliza un control robotizado para guiar el implante de prótesis de cadera.

Procesamiento de lenguaje y resolución de problemas: PROVER B (Littman *et al*., 1999) es un programa informático que resuelve crucigramas mejor que la mayoría de los humanos, utilizando restricciones en programas de relleno de palabras, una gran base de datos de crucigramas, y varias fuentes de información como diccionarios y bases de datos *online*, que incluyen la lista de películas y los actores que intervienen en ellas, entre otras cosas. Por ejemplo, determina que la pista «Historia de Niza» se puede resolver con «ETAGE» ya que su base de datos incluye el par pista/solución «Historia en Francia/ETAGE» y porque reconoce que los patrones «Niza X» y «X en Francia» a menudo tienen la misma solución. El programa no sabe que Niza es una ciudad de Francia, pero es capaz de resolver el puzle.

Estos son algunos de los ejemplos de sistemas de inteligencia artificial que existen hoy en día. No se trata de magia o ciencia ficción, son más bien ciencia, ingeniería y matemáticas, para los que este libro proporciona una introducción.

1.5 Resumen

En este capítulo se define la IA y se establecen los antecedentes culturales que han servido de base. Algunos de los aspectos más destacables son:

- Cada uno tiene una visión distinta de lo que es la IA. Es importante responder a las dos preguntas siguientes: ¿Está interesado en el razonamiento y el comportamiento? ¿Desea modelar seres humanos o trabajar a partir de un ideal estándar?
- En este libro se adopta el criterio de que la inteligencia tiene que ver principalmente con las **acciones racionales**. Desde un punto de vista ideal, un **agente inteligente** es aquel que emprende la mejor acción posible ante una situación dada. Se estudiará el problema de la construcción de agentes que sean inteligentes en este sentido.
- Los filósofos (desde el año 400 a.C.) facilitaron el poder imaginar la IA, al concebir la idea de que la mente es de alguna manera como una máquina que funciona a partir del conocimiento codificado en un lenguaje interno, y al considerar que el pensamiento servía para seleccionar la acción a llevar a cabo.
- Las matemáticas proporcionaron las herramientas para manipular tanto las aseveraciones de certeza lógicas, como las inciertas de tipo probabilista. Asimismo,

prepararon el terreno para un entendimiento de lo que es el cálculo y el razonamiento con algoritmos.

- Los economistas formalizaron el problema de la toma de decisiones para maximizar los resultados esperados.
- Los psicólogos adoptaron la idea de que los humanos y los animales podían considerarse máquinas de procesamiento de información. Los lingüistas demostraron que el uso del lenguaje se ajusta a ese modelo.
- Los informáticos proporcionaron los artefactos que hicieron posible la aplicación de la IA. Los programas de IA tienden a ser extensos y no podrían funcionar sin los grandes avances en velocidad y memoria aportados por la industria informática.
- La teoría de control se centra en el diseño de dispositivos que actúan de forma óptima con base en la retroalimentación que reciben del entorno en el que están inmersos. Inicialmente, las herramientas matemáticas de la teoría de control eran bastante diferentes a las técnicas que utilizaba la IA, pero ambos campos se están acercando.
- Las historia de la IA ha pasado por ciclos de éxito, injustificado optimismo y consecuente desaparición de entusiasmo y apoyos financieros. También ha habido ciclos caracterizados por la introducción de enfoques nuevos y creativos y de un perfeccionamiento sistemático de los mejores.
- La IA ha avanzado más rápidamente en la década pasada debido al mayor uso del método científico en la experimentación y comparación de propuestas.
- Los avances recientes logrados en el entendimiento de las bases teóricas de la inteligencia han ido aparejados con las mejoras realizadas en la optimización de los sistemas reales. Los subcampos de la IA se han integrado más y la IA ha encontrado elementos comunes con otras disciplinas.

NOTAS BIBLIOGRÁFICAS E HISTÓRICAS

El estatus metodológico de la inteligencia artificial se ha investigado en *The Sciences of the Artificial*, escrito por Herb Simon (1981), en el cual se analizan áreas de investigación interesadas en el desarrollo de artefactos complejos. Explica cómo la IA se puede ver como ciencia y matemática. Cohen (1995) proporciona una visión de la metodología experimental en el marco de la IA. Ford y Hayes (1995) presentan una revisión crítica de la utilidad de la Prueba de Turing.

Artificial Intelligence: The Very Idea, de John Haugeland (1985), muestra una versión amena de los problemas prácticos y filosóficos de la IA. La ciencia cognitiva está bien descrita en varios textos recientes (Johnson-Laird, 1988; Stillings *et al.*, 1995; Thagard, 1996) y en la *Encyclopedia of the Cognitive Sciences* (Wilson y Keil, 1999). Baker (1989) cubre la parte sintáctica de la lingüística moderna, y Chierchia y McConnell-Ginet (1990) la semántica. Jurafsky y Martin (2000) revisan la lingüística computacional.

Los primeros trabajos en el campo de la IA se citan en *Computers and Thought* (1963), de Feigenbaum y Feldman, en *Semantic Information Processing* de Minsky, y en la serie *Machine Intelligence*, editada por Donald Michie. Webber y Nilsson (1981)

y Luger (1995) han recogido una nutrida cantidad de artículos influyentes. Los primeros artículos sobre redes neuronales están reunidos en *Neurocomputing* (Anderson y Rosenfeld, 1988). La *Encyclopedia of AI* (Shapiro, 1992) contiene artículos de investigación sobre prácticamente todos los temas de IA. Estos artículos son muy útiles para iniciarse en las diversas áreas presentes en la literatura científica.

El trabajo más reciente se encuentra en las actas de las mayores conferencias de IA: la *International Joint Conference on AI* (IJCAI), de carácter bianual, la *European Conference on AI* (ECAI), de carácter bianual, y la *National Conference on AI*, conocida normalmente como AAAI por la organización que la patrocina. Las revistas científicas que presentan aspectos generales de la IA más importantes son *Artificial Intelligence, Computational Intelligence, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Intelligent Systems*, y la revista electrónica *Journal of Artificial Intelligence Research*. Hay también numerosas revistas y conferencias especializadas en áreas concretas, que se mencionarán en los capítulos apropiados. La asociaciones profesionales de IA más importantes son la American Association for Artificial Intelligence (AAAI), la ACM Special Interest Group in Artificial Intelligence (SIGART), y la Society for Artificial Intelligence and Simulation of Behaviour (AISB). La revista *AI Magazine* de AAAI contiene muchos artículos de interés general y manuales, y su página Web, aaai.org contiene noticias e información de referencia.

EJERCICIOS

El propósito de los siguientes ejercicios es estimular la discusión, y algunos de ellos se podrían utilizar como proyectos. Alternativamente, se podría hacer un esfuerzo inicial para solucionarlos ahora, de forma que una vez se haya leído todo el libro se puedan revisar estos primeros intentos.

1.1 Defina con sus propias palabras: (a) inteligencia, (b) inteligencia artificial, (c) agente.

1.2 Lea el artículo original de Turing sobre IA (Turing, 1950). En él se comentan algunas objeciones potenciales a su propuesta y a su prueba de inteligencia. ¿Cuáles de estas objeciones tiene todavía validez? ¿Son válidas sus refutaciones? ¿Se le ocurren nuevas objeciones a esta propuesta teniendo en cuenta los desarrollos realizados desde que se escribió el artículo? En el artículo, Turing predijo que para el año 2000 sería probable que un computador tuviera un 30 por ciento de posibilidades de superar una Prueba de Turing dirigida por un evaluador inexperto con una duración de cinco minutos. ¿Considera razonable lo anterior en el mundo actual? ¿Y en los próximos 50 años?

1.3 Todos los años se otorga el premio Loebner al programa que lo hace mejor en una Prueba de Turing concreta. Investigue y haga un informe sobre el último ganador del premio Loebner. ¿Qué técnica utiliza? ¿Cómo ha hecho que progrese la investigación en el campo de la IA?

1.4 Hay clases de problemas bien conocidos que son intratables para los computadores, y otras clases sobre los cuales un computador no pueda tomar una decisión. ¿Quiere esto decir que es imposible lograr la IA?

1.5 Supóngase que se extiende ANALOGY, el programa de Evans, como para alcanzar una puntuación de 200 en una prueba normal de cociente de inteligencia. ¿Quiere decir lo anterior que se ha creado un programa más inteligente que un ser humano? Explíquese.

1.6 ¿Cómo puede la introspección (revisión de los pensamientos íntimos) ser inexacta? ¿Se puede estar equivocado sobre lo que se cree? Discútase.

1.7 Consulte en la literatura existente sobre la IA si alguna de las siguientes tareas se puede efectuar con computadores:

a) Jugar una partida de tenis de mesa (ping-pong) decentemente.
b) Conducir un coche en el centro del Cairo.
c) Comprar comestibles para una semana en el mercado.
d) Comprar comestibles para una semana en la web.
e) Jugar una partida de *bridge* decentemente a nivel de competición.
f) Descubrir y demostrar nuevos teoremas matemáticos.
g) Escribir intencionadamente una historia divertida.
h) Ofrecer asesoría legal competente en un área determinada.
i) Traducir inglés hablado al sueco hablado en tiempo real.
j) Realizar una operación de cirugía compleja.

En el caso de las tareas que no sean factibles de realizar en la actualidad, trate de describir cuáles son las dificultades y calcule para cuándo se podrán superar.

1.8 Algunos autores afirman que la percepción y las habilidades motoras son la parte más importante de la inteligencia y que las capacidades de «alto nivel» son más bien parásitas (simples añadidos a las capacidades básicas). Es un hecho que la mayor parte de la evolución y que la mayor parte del cerebro se han concentrado en la percepción y las habilidades motoras, en tanto la IA ha descubierto que tareas como juegos e inferencia lógica resultan más sencillas, en muchos sentidos, que percibir y actuar en el mundo real. ¿Consideraría usted que ha sido un error la concentración tradicional de la IA en las capacidades cognitivas de alto nivel?

1.9 ¿Por qué la evolución tiende a generar sistemas que actúan racionalmente? ¿Qué objetivos deben intentar alcanzar estos sistemas?

1.10 ¿Son racionales las acciones reflejas (como retirar la mano de una estufa caliente)? ¿Son inteligentes?

1.11 «En realidad los computadores no son inteligentes, hacen solamente lo que le dicen los programadores». ¿Es cierta la última aseveración, e implica a la primera?

1.12 «En realidad los animales no son inteligentes, hacen solamente lo que le dicen sus genes». ¿Es cierta la última aseveración, e implica a la primera?

1.13 «En realidad los animales, los humanos y los computadores no pueden ser inteligentes, ellos sólo hacen lo que los átomos que los forman les dictan siguiendo las leyes de la física». ¿Es cierta la última aseveración, e implica a la primera?

2 Agentes inteligentes

Donde se discutirá la naturaleza de los agentes ideales, sus diversos hábitats y las formas de organizar los tipos de agentes existentes.

El Capítulo 1 identifica el concepto de **agente racional** como central en la perspectiva de la inteligencia artificial que presenta este libro. Esta noción se concreta más a lo largo de este capítulo. Se mostrará como el concepto de racionalidad se puede aplicar a una amplia variedad de agentes que operan en cualquier medio imaginable. En el libro, la idea es utilizar este concepto para desarrollar un pequeño conjunto de principios de diseño que sirvan para construir agentes útiles, sistemas que se puedan llamar razonablemente **inteligentes**.

Se comienza examinando los agentes, los medios en los que se desenvuelven, y la interacción entre éstos. La observación de que algunos agentes se comportan mejor que otros nos lleva naturalmente a la idea de agente racional, aquel que se comporta tan bien como puede. La forma de actuar del agente depende de la naturaleza del medio; algunos hábitats son más complejos que otros. Se proporciona una categorización cruda del medio y se muestra cómo las propiedades de un hábitat influyen en el diseño de agentes adecuados para ese entorno. Se presenta un número de «esquemas» básicos para el diseño de agentes, a los que se dará cuerpo a lo largo del libro.

2.1 Agentes y su entorno

MEDIOAMBIENTE

Un **agente** es cualquier cosa capaz de percibir su **medioambiente** con la ayuda de **sensores** y actuar en ese medio utilizando **actuadores**[1]. La Figura 2.1 ilustra esta idea sim-

[1] Se usa este término para indicar el elemento que reacciona a un estímulo realizando una acción (*N. del RT*).

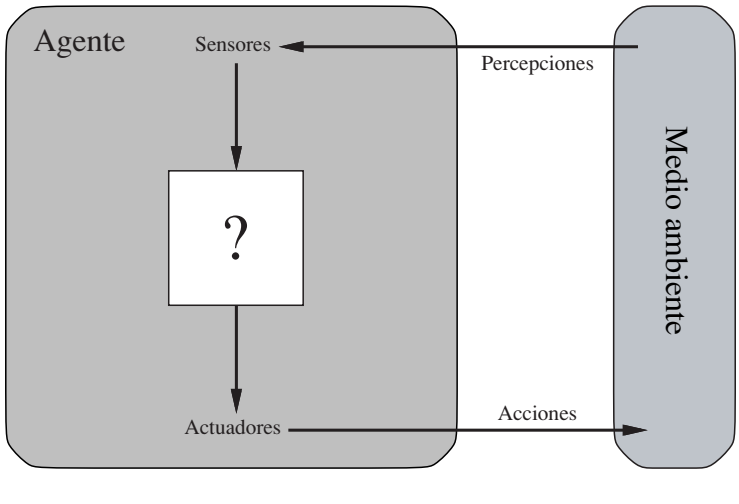

Figura 2.1 Los agentes interactúan con el medioambiente mediante sensores y efectores.

SENSOR

ACTUADOR

ple. Un agente humano tiene ojos, oídos y otros órganos sensoriales además de manos, piernas, boca y otras partes del cuerpo para actuar. Un agente robot recibe pulsaciones del teclado, archivos de información y paquetes vía red a modo de entradas sensoriales y actúa sobre el medio con mensajes en el monitor, escribiendo ficheros y enviando paquetes por la red. Se trabajará con la hipótesis general de que cada agente puede percibir sus propias acciones (pero no siempre sus efectos).

PERCEPCIÓN

SECUENCIA DE PERCEPTORES

El término **percepción** se utiliza en este contexto para indicar que el agente puede recibir entradas en cualquier instante. La **secuencia de percepciones** de un agente refleja el historial completo de lo que el agente ha recibido. En general, *un agente tomará una decisión en un momento dado dependiendo de la secuencia completa de percepciones hasta ese instante*. Si se puede especificar qué decisión tomará un agente para cada una de las posibles secuencias de percepciones, entonces se habrá explicado más o menos todo lo que se puede decir de un agente. En términos matemáticos se puede decir que el comportamiento del agente viene dado por la **función del agente** que proyecta una percepción dada en una acción.

FUNCIÓN DEL AGENTE

La función que describe el comportamiento de un agente se puede presentar en *forma de tabla*; en la mayoría de los casos esta tabla sería muy grande (infinita a menos que se limite el tamaño de la secuencia de percepciones que se quiera considerar). Dado un agente, con el que se quiera experimentar, se puede, en principio, construir esta tabla teniendo en cuenta todas las secuencias de percepción y determinando qué acción lleva a cabo el agente en respuesta[2]. La tabla es, por supuesto, una caracterización *externa* del agente. *Inicialmente*, la función del agente para un agente artificial se imple-

[2] Si el agente selecciona la acción de manera aleatoria, entonces sería necesario probar cada secuencia muchas veces para identificar la probabilidad de cada acción. Se puede pensar que actuar de manera aleatoria es ridículo, pero como se verá posteriormente puede ser muy inteligente.

PROGRAMA DEL AGENTE

mentará mediante el **programa del agente**. Es importante diferenciar estas dos ideas. La función del agente es una descripción matemática abstracta; el programa del agente es una implementación completa, que se ejecuta sobre la arquitectura del agente.

Para ilustrar esta idea se utilizará un ejemplo muy simple, el mundo de la aspiradora presentado en la Figura 2.2. Este mundo es tan simple que se puede describir todo lo que en él sucede; es un mundo hecho a medida, para el que se pueden inventar otras variaciones. Este mundo en particular tiene solamente dos localizaciones: cuadrícula *A* y *B*. La aspiradora puede percibir en qué cuadrante se encuentra y si hay suciedad en él. Puede elegir si se mueve hacia la izquierda, derecha, aspirar la suciedad o no hacer nada. Una función muy simple para el agente vendría dada por: si la cuadrícula en la que se encuentra está sucia, entonces aspirar, de otra forma cambiar de cuadrícula. Una muestra parcial de la función del agente representada en forma de tabla aparece en la Figura 2.3. Un programa de agente simple para esta función de agente se mostrará posteriormente en la Figura 2.8.

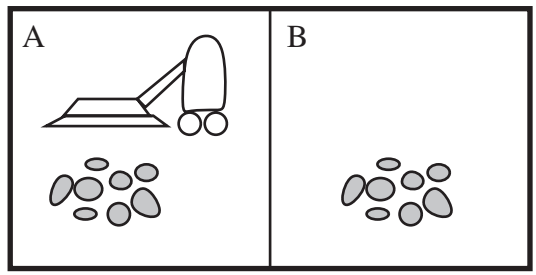

Figura 2.2 El mundo de la aspiradora con dos localizaciones solamente.

Secuencia de percepciones	Acción
[*A, Limpio*]	*Derecha*
[*A, Sucio*]	*Aspirar*
[*B, Limpio*]	*Izquierda*
[*B, Sucio*]	*Aspirar*
[*A, Limpio*], [*A, Limpio*]	*Derecha*
[*A, Limpio*], [*A, Sucio*]	*Aspirar*
—	—
—	—
—	—
[*A, Limpio*], [*A, Limpio*], [*A, Limpio*]	*Derecha*
[*A, Limpio*], [*A, Limpio*], [*A, Sucio*]	*Aspirar*
—	—
—	—
—	—

Figura 2.3 Tabla parcial de una función de agente sencilla para el mundo de la aspiradora que se muestra en la Figura 2.2.

Revisando la Figura 2.3, se aprecia que se pueden definir varios agentes para el mundo de la aspiradora simplemente rellenando la columna de la derecha de formas distintas. La pregunta obvia, entonces es: *¿cuál es la mejor forma de rellenar una tabla*? En otras palabras, ¿qué hace que un agente sea bueno o malo, inteligente o estúpido? Estas preguntas se responden en la siguiente sección.

Antes de terminar esta sección, es necesario remarcar que la noción de agente es supuestamente una herramienta para el análisis de sistemas, y no una caracterización absoluta que divida el mundo entre agentes y no agentes. Se puede ver una calculadora de mano como un agente que elige la acción de mostrar «4» en la pantalla, dada la secuencia de percepciones «2 + 2 =». Pero este análisis difícilmente puede mejorar nuestro conocimiento acerca de las calculadoras.

2.2 Buen comportamiento: el concepto de racionalidad

AGENTE RACIONAL

Un **agente racional** es aquel que hace lo correcto; en términos conceptuales, cada elemento de la tabla que define la función del agente se tendría que rellenar correctamente. Obviamente, hacer lo correcto es mejor que hacer algo incorrecto, pero ¿qué significa hacer lo correcto? Como primera aproximación, se puede decir que lo correcto es aquello que permite al agente obtener un resultado mejor. Por tanto, se necesita determinar una forma de medir el éxito. Ello, junto a la descripción del entorno y de los sensores y actuadores del agente, proporcionará una especificación completa de la tarea que desempeña el agente. Dicho esto, ahora es posible definir de forma más precisa qué significa la racionalidad.

Medidas de rendimiento

MEDIDAS DE RENDIMIENTO

Las **medidas de rendimiento** incluyen los criterios que determinan el éxito en el comportamiento del agente. Cuando se sitúa un agente en un medio, éste genera una secuencia de acciones de acuerdo con las percepciones que recibe. Esta secuencia de acciones hace que su hábitat pase por una secuencia de estados. Si la secuencia es la deseada, entonces el agente habrá actuado correctamente. Obviamente, no hay una única medida adecuada para todos los agentes. Se puede preguntar al agente por su opinión subjetiva acerca de su propia actuación, pero muchos agentes serían incapaces de contestar, y otros podrían engañarse a sí mismos[3]. Por tanto hay que insistir en la importancia de utilizar medidas de rendimiento objetivas, que normalmente determinará el diseñador encargado de la construcción del agente.

Si retomamos el ejemplo de la aspiradora de la sección anterior, se puede proponer utilizar como medida de rendimiento la cantidad de suciedad limpiada en un período de

[3] Los agentes humanos son conocidos en particular por su «acidez», hacen creer que no quieren algo después de no haberlo podido conseguir, por ejemplo, «Ah bueno, de todas formas no quería ese estúpido Premio Nobel».

ocho horas. Con agentes racionales, por supuesto, se obtiene lo que se demanda. Un agente racional puede maximizar su medida de rendimiento limpiando la suciedad, tirando la basura al suelo, limpiándola de nuevo, y así sucesivamente. Una medida de rendimiento más adecuada recompensaría al agente por tener el suelo limpio. Por ejemplo, podría ganar un punto por cada cuadrícula limpia en cada período de tiempo (quizás habría que incluir algún tipo de penalización por la electricidad gastada y el ruido generado). *Como regla general, es mejor diseñar medidas de utilidad de acuerdo con lo que se quiere para el entorno, más que de acuerdo con cómo se cree que el agente debe comportarse*.

La selección de la medida de rendimiento no es siempre fácil. Por ejemplo, la noción de «suelo limpio» del párrafo anterior está basada en un nivel de limpieza medio a lo largo del tiempo. Además, este nivel medio de limpieza se puede alcanzar de dos formas diferentes, llevando a cabo una limpieza mediocre pero continua o limpiando en profundidad, pero realizando largos descansos. La forma más adecuada de hacerlo puede venir dada por la opinión de un encargado de la limpieza profesional, pero en realidad es una cuestión filosófica profunda con fuertes implicaciones. ¿Qué es mejor, una vida temeraria con altos y bajos, o una existencia segura pero aburrida? ¿Qué es mejor, una economía en la que todo el mundo vive en un estado de moderada pobreza o una en la que algunos viven en la abundancia y otros son muy pobres? Estas cuestiones se dejan como ejercicio para los lectores diligentes.

Racionalidad

La racionalidad en un momento determinado depende de cuatro factores:

- La medida de rendimiento que define el criterio de éxito.
- El conocimiento del medio en el que habita acumulado por el agente.
- Las acciones que el agente puede llevar a cabo.
- La secuencia de percepciones del agente hasta este momento.

DEFINICIÓN DE AGENTE RACIONAL

Esto nos lleva a la **definición de agente racional**:

> *En cada posible secuencia de percepciones, un agente racional deberá emprender aquella acción que supuestamente maximice su medida de rendimiento, basándose en las evidencias aportadas por la secuencia de percepciones y en el conocimiento que el agente mantiene almacenado.*

Considerando que el agente aspiradora limpia una cuadrícula si está sucia y se mueve a la otra si no lo está (ésta es la función del agente que aparece en la tabla de la Figura 2.3), ¿se puede considerar racional? ¡Depende! Primero, se debe determinar cuál es la medida de rendimiento, qué se conoce del entorno, y qué sensores y actuadores tiene el agente. Si asumimos que:

- La medida de rendimiento premia con un punto al agente por cada recuadro limpio en un período de tiempo concreto, a lo largo de una «vida» de 1.000 períodos.
- La «geografía» del medio se conoce *a priori* (Figura 2.2), pero que la distribución de la suciedad y la localización inicial del agente no se conocen. Las cuadrículas se mantienen limpias y aspirando se limpia la cuadrícula en que se encuentre el agente. Las acciones *Izquierda* y *Derecha* mueven al agente hacia la izquierda y

derecha excepto en el caso de que ello pueda llevar al agente fuera del recinto, en este caso el agente permanece donde se encuentra.
- Las únicas acciones permitidas son *Izquierda*, *Derecha*, *Aspirar* y *NoOp* (no hacer nada).
- El agente percibe correctamente su localización y si esta localización contiene suciedad.

Puede afirmarse que *bajo estas circunstancias* el agente es verdaderamente racional; el rendimiento que se espera de este agente es por lo menos tan alto como el de cualquier otro agente. El Ejercicio 2.4 pide que se pruebe este hecho.

Fácilmente se puede observar que el agente puede resultar irracional en circunstancias diferentes. Por ejemplo, cuando toda la suciedad se haya eliminado el agente oscilará innecesariamente hacia delante y atrás; si la medida de rendimiento incluye una penalización de un punto por cada movimiento hacia la derecha e izquierda, la respuesta del agente será pobre. Un agente más eficiente no hará nada si está seguro de que todas las cuadrículas están limpias. Si una cuadrícula se ensucia de nuevo, el agente debe identificarlo en una de sus revisiones ocasionales y limpiarla. Si no se conoce la geografía del entorno, el agente tendrá que explorarla y no quedarse parado en las cuadrículas *A* y *B*. El Ejercicio 2.4 pide que se diseñen agentes para estos casos.

Omnisciencia, aprendizaje y autonomía

OMNISCIENCIA

Es necesario tener cuidado al distinguir entre racionalidad y **omnisciencia**. Un agente omnisciente conoce el resultado de su acción y actúa de acuerdo con él; sin embargo, en realidad la omnisciencia no es posible. Considerando el siguiente ejemplo: estoy paseando por los Campos Elíseos y veo un amigo al otro lado de la calle. No hay tráfico alrededor y no tengo ningún compromiso, entonces, actuando racionalmente, comenzaría a cruzar la calle. Al mismo tiempo, a 33.000 pies de altura, se desprende la puerta de un avión[4], y antes de que termine de cruzar al otro lado de la calle me encuentro aplastado. ¿Fue irracional cruzar la calle? Sería de extrañar que en mi nota necrológica apareciera «Un idiota intentando cruzar la calle».

Este ejemplo muestra que la racionalidad no es lo mismo que la perfección. La racionalidad maximiza el rendimiento esperado, mientras la perfección maximiza el resultado real. Alejarse de la necesidad de la perfección no es sólo cuestión de hacer justicia con los agentes. El asunto es que resulta imposible diseñar un agente que siempre lleve a cabo, de forma sucesiva, las mejores acciones después de un acontecimiento, a menos que se haya mejorado el rendimiento de las bolas de cristal o las máquinas de tiempo.

La definición propuesta de racionalidad no requiere omnisciencia, ya que la elección racional depende sólo de la secuencia de percepción hasta *la fecha*. Es necesario asegurase de no haber permitido, por descuido, que el agente se dedique decididamente a llevar a cabo acciones poco inteligentes. Por ejemplo, si el agente no mirase a ambos lados de la calle antes de cruzar una calle muy concurrida, entonces su secuencia de per-

[4] *Véase* N. Henderson, «New door latches urged for Boeing 747 jumbo jets» (es urgente dotar de nuevas cerraduras a las puertas de los Boeing jumbo 747), *Washington Post*, 24 de agosto de 1989.

cepción no le indicaría que se está acercando un gran camión a gran velocidad. ¿La definición de racionalidad nos está indicando que está bien cruzar la calle? ¡Todo lo contrario! Primero, no sería racional cruzar la calle sólo teniendo esta secuencia de percepciones incompleta: el riesgo de accidente al cruzarla sin mirar es demasiado grande. Segundo, un agente racional debe elegir la acción de «mirar» antes de intentar cruzar la calle, ya que el mirar maximiza el rendimiento esperado. Llevar a cabo acciones *con la intención de modificar percepciones futuras*, en ocasiones proceso denominado **recopilación de información**, es una parte importante de la racionalidad y se comenta en profundidad en el Capítulo 16. Un segundo ejemplo de recopilación de información lo proporciona la **exploración** que debe llevar a cabo el agente aspiradora en un medio inicialmente desconocido.

RECOPILACIÓN DE INFORMACIÓN

EXPLORACIÓN

La definición propuesta implica que el agente racional no sólo recopile información, sino que **aprenda** lo máximo posible de lo que está percibiendo. La configuración inicial del agente puede reflejar un conocimiento preliminar del entorno, pero a medida que el agente adquiere experiencia éste puede modificarse y aumentar. Hay casos excepcionales en los que se conoce totalmente el entorno *a priori*. En estos casos, el agente no necesita percibir y aprender; simplemente actúa de forma correcta. Por supuesto, estos agentes son muy frágiles. Considérese el caso del humilde escarabajo estercolero. Después de cavar su nido y depositar en él su huevos, tomó una bola de estiércol de una pila cercana para tapar su entrada. Si *durante el trayecto* se le quita la bola, el escarabajo continuará su recorrido y hará como si estuviera tapando la entrada del nido, sin tener la bola y sin darse cuanta de ello. La evolución incorporó una suposición en la conducta del escarabajo, y cuando se viola, el resultado es un comportamiento insatisfactorio. La avispa cavadora es un poco más inteligente. La avispa hembra cavará una madriguera, saldrá de ella, picará a una oruga y la llevará a su madriguera, se introducirá en la madriguera para comprobar que todo está bien, arrastrará la oruga hasta el fondo y pondrá sus huevos. La oruga servirá como fuente de alimento cuando los huevos se abran. Hasta ahora todo bien, pero si un entomólogo desplaza la oruga unos centímetros fuera cuando la avispa está revisando la situación, ésta volverá a la etapa de «arrastre» que figura en su plan, y continuará con el resto del plan sin modificación alguna, incluso después de que se intervenga para desplazar la oruga. La avispa cavadora no es capaz de aprender que su plan innato está fallando, y por tanto no lo cambiará.

APRENDIZAJE

Los agentes con éxito dividen las tareas de calcular la función del agente en tres períodos diferentes: cuando se está diseñando el agente, y están los diseñadores encargados de realizar algunos de estos cálculos; cuando está pensando en la siguiente operación, el agente realiza más cálculos; y cuando está aprendiendo de la experiencia, el agente lleva a cabo más cálculos para decidir cómo modificar su forma de comportarse.

AUTONOMÍA

Se dice que un agente carece de **autonomía** cuando se apoya más en el conocimiento inicial que le proporciona su diseñador que en sus propias percepciones. Un agente racional debe ser autónomo, debe saber aprender a determinar cómo tiene que compensar el conocimiento incompleto o parcial inicial. Por ejemplo, el agente aspiradora que aprenda a prever dónde y cuándo aparecerá suciedad adicional lo hará mejor que otro que no aprenda. En la práctica, pocas veces se necesita autonomía completa desde el comienzo: cuando el agente haya tenido poca o ninguna experiencia, tendrá que actuar de forma aleatoria a menos que el diseñador le haya proporcionado ayuda. Así, de la

misma forma que la evolución proporciona a los animales sólo los reactivos necesarios para que puedan sobrevivir lo suficiente para aprender por ellos mismos, sería razonable proporcionar a los agentes que disponen de inteligencia artificial un conocimiento inicial, así como de la capacidad de aprendizaje. Después de las suficientes experiencias interaccionando con el entorno, el comportamiento del agente racional será efectivamente *independiente* del conocimiento que poseía inicialmente. De ahí, que la incorporación del aprendizaje facilite el diseño de agentes racionales individuales que tendrán éxito en una gran cantidad de medios.

2.3 La naturaleza del entorno

ENTORNOS DE TRABAJO

Ahora que se tiene una definición de racionalidad, se está casi preparado para pensar en la construcción de agentes racionales. Primero, sin embargo, hay que centrarse en los **entornos de trabajo**, que son esencialmente los «problemas» para los que los agentes racionales son las «soluciones». Para ello se comienza mostrando cómo especificar un entorno de trabajo, ilustrando el proceso con varios ejemplos. Posteriormente se mostrará que el entorno de trabajo ofrece diferentes posibilidades, de forma que cada una de las posibilidades influyen directamente en el diseño del programa del agente.

Especificación del entorno de trabajo

REAS

En la discusión de la racionalidad de un agente aspiradora simple, hubo que especificar las medidas de rendimiento, el entorno, y los actuadores y sensores del agente. Todo ello forma lo que se llama el **entorno de trabajo**, para cuya denominación se utiliza el acrónimo **REAS** (**R**endimiento, **E**ntorno, **A**ctuadores, **S**ensores). En el diseño de un agente, el primer paso debe ser siempre especificar el entorno de trabajo de la forma más completa posible.

El mundo de la aspiradora fue un ejemplo simple; considérese ahora un problema más complejo: un taxista automático. Este ejemplo se utilizará a lo largo del capítulo. Antes de alarmar al lector, conviene aclarar que en la actualidad la construcción de un taxi automatizado está fuera del alcance de la tecnología actual. *Véase* en la página 31 la descripción de un robot conductor que ya existe en la actualidad, o lea las actas de la conferencia *Intelligent Transportation Systems*. La tarea de conducir un automóvil, en su totalidad, es extremadamente *ilimitada*. No hay límite en cuanto al número de nuevas combinaciones de circunstancias que pueden surgir (por esta razón se eligió esta actividad en la presente discusión). La Figura 2.4 resume la descripción REAS para el entorno de trabajo del taxi. El próximo párrafo explica cada uno de sus elementos en más detalle.

Primero, ¿cuál es el **entorno de trabajo** en el que el taxista automático aspira a conducir? Dentro de las cualidades deseables que debería tener se incluyen el que llegue al destino correcto; que minimice el consumo de combustible; que minimice el tiempo de viaje y/o coste; que minimice el número de infracciones de tráfico y de molestias a otros conductores; que maximice la seguridad, la comodidad del pasajero y el

Tipo de agente	Medidas de rendimiento	Entorno	Actuadores	Sensores
Taxista	Seguro, rápido, legal, viaje confortable, maximización del beneficio	Carreteras, otro tráfico, peatones, clientes	Dirección, acelerador, freno, señal, bocina, visualizador	Cámaras, sónar, velocímetro, GPS, tacómetro, visualizador de la aceleración, sensores del motor, teclado

Figura 2.4 Descripción REAS del entorno de trabajo de un taxista automático.

beneficio. Obviamente, alguno de estos objetivos entran en conflicto por lo que habrá que llegar a acuerdos.

Siguiente, ¿cuál es el **entorno** en el que se encontrará el taxi? Cualquier taxista debe estar preparado para circular por distintas carreteras, desde caminos rurales y calles urbanas hasta autopistas de 12 carriles. En las carreteras se pueden encontrar con tráfico, peatones, animales, obras, coches de policía, charcos y baches. El taxista también tiene que comunicarse tanto con pasajeros reales como potenciales. Hay también elecciones opcionales. El taxi puede operar en California del Sur, donde la nieve es raramente un problema, o en Alaska, donde raramente no lo es. Puede conducir siempre por la derecha, o puede ser lo suficientemente flexible como para que circule por la izquierda cuando se encuentre en el Reino Unido o en Japón. Obviamente, cuanto más restringido esté el entorno, más fácil será el problema del diseño.

Los **actuadores** disponibles en un taxi automático serán más o menos los mismos que los que tiene a su alcance un conductor humano: el control del motor a través del acelerador y control sobre la dirección y los frenos. Además, necesitará tener una pantalla de visualización o un sintetizador de voz para responder a los pasajeros, y quizás algún mecanismo para comunicarse, educadamente o de otra forma, con otros vehículos.

Para alcanzar sus objetivos en el entorno en el que circula, el taxi necesita saber dónde está, qué otros elementos están en la carretera, y a qué velocidad circula. Sus **sensores** básicos deben, por tanto, incluir una o más cámaras de televisión dirigidas, un velocímetro y un tacómetro. Para controlar el vehículo adecuadamente, especialmente en las curvas, debe tener un acelerador; debe conocer el estado mecánico del vehículo, de forma que necesitará sensores que controlen el motor y el sistema eléctrico. Debe tener instrumentos que no están disponibles para un conductor medio: un sistema de posicionamiento global vía satélite (GPS) para proporcionarle información exacta sobre su posición con respecto a un mapa electrónico, y sensores infrarrojos o sonares para detectar las distancias con respecto a otros coches y obstáculos. Finalmente, necesitará un teclado o micrófono para que el pasajero le indique su destino.

La Figura 2.5 muestra un esquema con los elementos REAS básicos para diferentes clases de agentes adicionales. Más ejemplos aparecerán en el Ejercicio 2.5. Puede sorprender a algunos lectores que se incluya en la lista de tipos de agente algunos programas que operan en la totalidad del entorno artificial definido por las entradas del teclado y los caracteres impresos en el monitor. «Seguramente», nos podamos pregun-

Tipo de agente	Medidas de rendimiento	Entorno	Actuadores	Sensores
Sistema de diagnóstico médico	Pacientes sanos, reducir costes, demandas	Pacientes, hospital, personal	Visualizar preguntas, pruebas, diagnósticos, tratamientos, casos	Teclado para la entrada de síntomas, conclusiones, respuestas de pacientes
Sistema de análisis de imágenes de satélites	Categorización de imagen correcta	Conexión con el satélite en órbita	Visualizar la categorización de una escena	Matriz de pixels de colores
Robot para la selección de componentes	Porcentaje de componentes clasificados en los cubos correctos	Cinta transportadora con componentes, cubos	Brazo y mano articulados	Cámara, sensor angular
Controlador de una refinería	Maximizar la pureza, producción y seguridad	Refinería, operadores	Válvulas, bombas, calentadores, monitores	Temperatura, presión, sensores químicos
Tutor de inglés interactivo	Maximizar la puntuación de los estudiantes en los exámenes	Conjunto de estudiantes, agencia examinadora	Visualizar los ejercicios, sugerencias, correcciones	Teclado de entrada

Figura 2.5 Ejemplos de tipos de agentes y sus descripciones REAS.

tar, «¿este no es un entorno real, verdad?». De hecho, lo que importa no es la distinción entre un medio «real» y «artificial», sino la complejidad de la relación entre el comportamiento del agente, la secuencia de percepción generada por el medio y la medida de rendimiento. Algunos entornos «reales» son de hecho bastante simples. Por ejemplo, un robot diseñado para inspeccionar componentes según pasan por una cinta transportadora puede hacer uso de varias suposiciones simples: que la cinta siempre estará iluminada, que conocerá todos los componentes que circulen por la cinta, y que hay solamente dos acciones (aceptar y rechazar).

AGENTES *SOFTWARE*

SOFTBOTS

En contraste, existen algunos **agentes *software*** (o robots *software* o **softbots**) en entornos ricos y prácticamente ilimitados. Imagine un softbot diseñado para pilotar el simulador de vuelo de un gran avión comercial. El simulador constituye un medio muy detallado y complejo que incluye a otros aviones y operaciones de tierra, y el agente *software* debe elegir, en tiempo real, una de entre un amplio abanico de posibilidades. O imagine un robot diseñado para que revise fuentes de información en Internet y para que muestre aquellas que sean interesantes a sus clientes. Para lograrlo, deberá poseer cierta habilidad en el procesamiento de lenguaje natural, tendrá que aprender qué es lo que le interesa a cada cliente, y tendrá que ser capaz de cambiar sus planes dinámica-

mente, por ejemplo, cuando se interrumpa la conexión con una fuente de información o cuando aparezca una nueva. Internet es un medio cuya complejidad rivaliza con la del mundo físico y entre cuyos habitantes se pueden incluir muchos agentes artificiales.

Propiedades de los entornos de trabajo

El rango de los entornos de trabajo en los que se utilizan técnicas de IA es obviamente muy grande. Sin embargo, se puede identificar un pequeño número de dimensiones en las que categorizar estos entornos. Estas dimensiones determinan, hasta cierto punto, el diseño más adecuado para el agente y la utilización de cada una de las familias principales de técnicas en la implementación del agente. Primero se enumeran la dimensiones, y después se analizan varios entornos de trabajo para ilustrar estas ideas. Las definiciones dadas son informales; capítulos posteriores proporcionan definiciones más precisas y ejemplos de cada tipo de entorno.

TOTALMENTE OBSEVABLE

- **Totalmente observable** *vs.* **parcialmente observable.**
  Si los sensores del agente le proporcionan acceso al estado completo del medio en cada momento, entonces se dice que el entorno de trabajo es totalmente observable[5]. Un entorno de trabajo es, efectivamente, totalmente observable si los sensores detectan todos los aspectos que son relevantes en la toma de decisiones; la relevancia, en cada momento, depende de las medidas de rendimiento. Entornos totalmente observables son convenientes ya que el agente no necesita mantener ningún estado interno para saber qué sucede en el mundo. Un entorno puede ser parcialmente observable debido al ruido y a la existencia de sensores poco exactos o porque los sensores no reciben información de parte del sistema, por ejemplo, un agente aspiradora con sólo un sensor de suciedad local no puede saber si hay suciedad en la otra cuadrícula, y un taxi automatizado no pude saber qué están pensando otros conductores.

DETERMINISTA

ESTOCÁSTICO

- **Determinista** *vs.* **estocástico.**
  Si el siguiente estado del medio está totalmente determinado por el estado actual y la acción ejecutada por el agente, entonces se dice que el entorno es determinista; de otra forma es estocástico. En principio, un agente no se tiene que preocupar de la incertidumbre en un medio totalmente observable y determinista. Sin embargo, si el medio es parcialmente observable entonces puede *parecer* estocástico. Esto es particularmente cierto si se trata de un medio complejo, haciendo difícil el mantener constancia de todos las aspectos observados. Así, a menudo es mejor pensar en entornos deterministas o estocásticos *desde el punto de vista del agente*. El agente taxi es claramente estocástico en este sentido, ya que no se puede predecir el comportamiento del tráfico exactamente; más aún, una rueda se puede reventar y un motor se puede gripar sin previo aviso. El mundo de la aspiradora es deter-

[5] La primera edición de este libro utiliza los términos **accesible** e **inaccesible** en vez de **total** y **parcialmente observable; no determinista** en vez de **estocástico;** y **no episódico** en vez de **secuencial.** La nueva terminología es más consistente con el uso establecido.

minista, como ya se describió, pero las variaciones pueden incluir elementos estocásticos como la aparición de suciedad aleatoria y un mecanismo de succión ineficiente (Ejercicio 2.12). Si el medio es determinista, excepto para las acciones de otros agentes, decimos que el medio es **estratégico**.

ESTRATÉGICO

EPISÓDICO

SECUENCIAL

- **Episódico** *vs.* **secuencial**[6]**.**
  En un entorno de trabajo episódico, la experiencia del agente se divide en episodios atómicos. Cada episodio consiste en la percepción del agente y la realización de una única acción posterior. Es muy importante tener en cuenta que el siguiente episodio no depende de las acciones que se realizaron en episodios previos. En los medios episódicos la elección de la acción en cada episodio depende sólo del episodio en sí mismo. Muchas tareas de clasificación son episódicas. Por ejemplo, un agente que tenga que seleccionar partes defectuosas en una cadena de montaje basa sus decisiones en la parte que está evaluando en cada momento, sin tener en cuenta decisiones previas; más aún, a la decisión presente no le afecta el que la próxima fase sea defectuosa. En entornos secuenciales, por otro lado, la decisión presente puede afectar a decisiones futuras. El ajedrez y el taxista son secuenciales: en ambos casos, las acciones que se realizan a corto plazo pueden tener consecuencias a largo plazo. Los medios episódicos son más simples que los secuenciales porque la gente no necesita pensar con tiempo.

ESTÁTICO

DINAMICO

- **Estático** *vs.* **dinámico.**
  Si el entorno puede cambiar cuando el agente está deliberando, entonces se dice que el entorno es dinámico para el agente; de otra forma se dice que es estático. Los medios estáticos son fáciles de tratar ya que el agente no necesita estar pendiente del mundo mientras está tomando una decisión sobre una acción, ni necesita preocuparse sobre el paso del tiempo. Los medios dinámicos, por el contrario, están preguntando continuamente al agente qué quiere hacer; si no se ha decidido aún, entonces se entiende que ha tomado la decisión de no hacer nada. Si el entorno no cambia con el paso del tiempo, pero el rendimiento del agente cambia, entonces se dice que el medio es **semidinámico**. El taxista es claramente dinámico: tanto los otros coches como el taxi se están moviendo mientras el algoritmo que guía la conducción indica qué es lo próximo a hacer. El ajedrez, cuando se juega con un reloj, es semideterminista. Los crucigramas son estáticos.

SEMIDINÁMICO

DISCRETO

CONTINUO

- **Discreto** *vs.* **continuo.**
  La distinción entre discreto y continuo se puede aplicar al *estado* del medio, a la forma en la que se maneja el *tiempo* y a las *percepciones* y *acciones* del agente. Por ejemplo, un medio con estados discretos como el del juego del ajedrez tiene un número finito de estados distintos. El ajedrez tiene un conjunto discreto de percepciones y acciones. El taxista conduciendo define un estado continuo y un problema de tiempo continuo: la velocidad y la ubicación del taxi y de los otros vehículos pasan por un rango de valores continuos de forma suave a lo largo del

[6] La palabra «secuencial» se utiliza también en el campo de la informática como antónimo de «paralelo». Los dos significados no están relacionados.

tiempo. Las conducción del taxista es también continua (ángulo de dirección, etc.). Las imágenes captadas por cámaras digitales son discretas, en sentido estricto, pero se tratan típicamente como representaciones continuas de localizaciones e intensidades variables.

AGENTE INDIVIDUAL

MULTIAGENTE

- **Agente individual** *vs.* **multiagente.**
  La distinción entre el entorno de un agente individual y el de un sistema multiagente puede parecer suficientemente simple. Por ejemplo, un agente resolviendo un crucigrama por sí mismo está claramente en un entorno de agente individual, mientras que un agente que juega al ajedrez está en un entorno con dos agentes. Sin embargo hay algunas diferencias sutiles. Primero, se ha descrito que una entidad *puede* percibirse como un agente, pero no se ha explicado qué entidades se *deben* considerar agentes. ¿Tiene el agente *A* (por ejemplo el agente taxista) que tratar un objeto *B* (otro vehículo) como un agente, o puede tratarse méramente como un objeto con un comportamiento estocástico, como las olas de la playa o las hojas que mueve el viento? La distinción clave está en identificar si el comportamiento de *B* está mejor descrito por la maximización de una medida de rendimiento cuyo valor depende del comportamiento de *A*. Por ejemplo, en el ajedrez, la entidad oponente *B* intenta maximizar su medida de rendimiento, la cual, según las reglas, minimiza la medida de rendimiento del agente *A*. Por tanto, el ajedrez es un entorno multiagente **competitivo**. Por otro lado, en el medio definido por el taxista circulando, el evitar colisiones maximiza la medida de rendimiento de todos los agentes, así pues es un entorno multiagente parcialmente **cooperativo**. Es también parcialmente competitivo ya que, por ejemplo, sólo un coche puede ocupar una plaza de aparcamiento. Los problemas en el diseño de agentes que aparecen en los entornos multiagente son a menudo bastante diferentes de los que aparecen en entornos con un único agente; por ejemplo, la **comunicación** a menudo emerge como un comportamiento racional en entornos multiagente; en algunos entornos competitivos parcialmente observables el **comportamiento estocástico** es racional ya que evita las dificultades de la predicción.

COMPETITIVO

COOPERATIVO

Como es de esperar, el caso más complejo es el *parcialmente observable*, *estocástico*, *secuencial*, *dinámico*, *continuo* y *multiagente*. De hecho, suele suceder que la mayoría de las situaciones reales son tan complejas que sería discutible clasificarlas como *realmente* deterministas. A efectos prácticos, se deben tratar como estocásticas. Un taxista circulando es un problema, complejo a todos los efectos.

La Figura 2.6 presenta las propiedades de un número de entornos familiares. Hay que tener en cuenta que las respuestas no están siempre preparadas de antemano. Por ejemplo, se ha presentado el ajedrez como totalmente observable; en sentido estricto, esto es falso porque ciertas reglas que afectan al movimiento de las torres, el enroque y a movimientos por repetición requieren que se recuerden algunos hechos sobre la historia del juego que no están reflejados en el estado del tablero. Estas excepciones, por supuesto, no tienen importancia si las comparamos con aquellas que aparecen en el caso del taxista, el tutor de inglés, o el sistema de diagnóstico médico.

Entornos de trabajo	Observable	Determinista	Episódico	Estático	Discreto	Agentes
Crucigrama	Totalmente	Determinista	Secuencial	Estático	Discreto	Individual
Ajedrez con reloj	Totalmente	Estratégico	Secuencial	Semi	Discreto	Multi
Póker	Parcialmente	Estratégico	Secuencial	Estático	Discreto	Multi
Backgammon	Totalmente	Estocástico	Secuencial	Estático	Discreto	Multi
Taxi circulando	Parcialmente	Estocástico	Secuencial	Dinámico	Continuo	Multi
Diagnóstico médico	Parcialmente	Estocástico	Secuencial	Dinámico	Continuo	Individual
Análisis de imagen	Totalmente	Determinista	Episódico	Semi	Continuo	Individual
Robot clasificador	Parcialmente	Estocástico	Episódico	Dinámico	Continuo	Individual
Controlador de refinería	Parcialmente	Estocástico	Secuencial	Dinámico	Continuo	Individual
Tutor interactivo de inglés	Parcialmente	Estocástico	Secuencial	Dinámico	Discreto	Multi

Figura 2.6 Ejemplos de entornos de trabajo y sus características.

Otras entradas de la tabla dependen de cómo se haya definido el entorno de trabajo. Se ha definido el sistema de diagnóstico médico como un único agente porque no es rentable modelar el proceso de la enfermedad en un paciente como un agente; pero incluso el sistema de diagnóstico médico podría necesitar tener en cuenta a pacientes recalcitrantes y empleados escépticos, de forma que el entorno podría tener un aspecto multiagente. Más aún, el diagnóstico médico es episódico si se concibe como proporcionar un diagnóstico a partir de una lista de síntomas; el problema es secuencial si ello trae consigo la propuesta de una serie de pruebas, un proceso de evaluación a lo largo del tratamiento, y demás aspectos. Muchos entornos son, también, episódicos si se observan desde un nivel de abstracción más alto que el de las acciones individuales del agente. Por ejemplo, un torneo de ajedrez consiste en una secuencia de juegos; cada juego es un episodio, pero (a la larga) la contribución de los movimientos en una partida al resultado general que obtenga el agente no se ve afectada por los movimientos realizados en la partida anterior. Por otro lado, las decisiones tomadas en una partida concreta son ciertamente de tipo secuencial.

El repositorio de código asociado a este libro (aima.cs.berkeley.edu) incluye la implementación de un número de entornos, junto con un simulador de entornos de propósito general que sitúa uno o más agentes en un entorno simulado, observa su comportamiento a lo largo del tiempo, y los evalúa de acuerdo con una medida de rendimiento dada. Estos experimentos no sólo se han realizado para un medio concreto, sino que se han realizado con varios problemas obtenidos de una **clase de entornos**. Por ejemplo, para evaluar un taxista en un tráfico simulado, sería interesante hacer varias simulaciones con diferente tipo de tráfico, claridad y condiciones atmosféricas. Si se diseña un agente para un escenario concreto, se pueden sacar ventajas de las propiedades específicas de ese caso en particular, pero puede no identificarse un buen diseño para conducir en general. Por esta razón, el repositorio de código también incluye un **generador de entornos** para cada clase de medios que selecciona hábitats particulares (con ciertas posibilidades) en los que ejecutar los agentes. Por ejemplo, el generador de un entorno

CLASE DE ENTORNOS

GENERADOR DE ENTORNOS

para un agente aspiradora inicializa el patrón de suciedad y la localización del agente de forma aleatoria. Después, es interesante evaluar la eficacia media del agente en el contexto de la clase del entorno. Un agente racional para una clase de entorno maximiza el rendimiento medio. Los Ejercicios del 2.7 al 2.12 guían el proceso de desarrollo de una clase de entornos y la evaluación de varios agentes.

2.4 Estructura de los agentes

PROGRAMA DEL AGENTE

ARQUITECTURA

Hasta este momento se ha hablado de los agentes describiendo su *conducta*, la acción que se realiza después de una secuencia de percepciones dada. Ahora, se trata de centrarse en el núcleo del problema y hablar sobre cómo trabajan internamente. El trabajo de la IA es diseñar el **programa del agente** que implemente la función del agente que proyecta las percepciones en las acciones. Se asume que este programa se ejecutará en algún tipo de computador con sensores físicos y actuadores, lo cual se conoce como **arquitectura:**

$$Agente = arquitectura + programa$$

Obviamente, el programa que se elija tiene que ser apropiado para la arquitectura. Si el programa tiene que recomendar acciones como *Caminar*, la arquitectura tiene que tener piernas. La arquitectura puede ser un PC común, o puede ser un coche robotizado con varios computadores, cámaras, y otros sensores a bordo. En general, la arquitectura hace que las percepciones de los sensores estén disponibles para el programa, ejecuta los programas, y se encarga de que los actuadores pongan en marcha las acciones generadas. La mayor parte de este libro se centra en el diseño de programas para agentes, aunque los Capítulos 24 y 25 tratan sobre sensores y actuadores.

Programas de los agentes

Los programas de los agentes que se describen en este libro tienen la misma estructura: reciben las percepciones actuales como entradas de los sensores y devuelven una acción a los actuadores[7]. Hay que tener en cuenta la diferencia entre los programas de los agentes, que toman la percepción actual como entrada, y la función del agente, que recibe la percepción histórica completa. Los programas de los agentes reciben sólo la percepción actual como entrada porque no hay nada más disponible en el entorno; si las acciones del agente dependen de la secuencia completa de percepciones, el agente tendría que recordar las percepciones.

Los programas de los agente se describirán con la ayuda de un sencillo lenguaje pseudocódigo que se define en el Apéndice B. El repositorio de código disponible en Inter-

[7] Hay otras posibilidades para definir la estructura del programa para el agente; por ejemplo, los programas para agentes pueden ser **subrutinas** que se ejecuten asincrónicamente en el entorno de trabajo. Cada una de estas subrutinas tienen un puerto de entrada y salida y consisten en un bucle que interpreta las entradas del puerto como percepciones y escribe acciones en el puerto de salida.

```
función AGENTE-DIRIGIDO-MEDIANTE TABLA(percepción) devuelve una acción
    variables estáticas: percepciones, una secuencia, vacía inicialmente
                         tabla, una tabla de acciones, indexada por las secuencias de
                         percepciones, totalmente definida inicialmente

    añadir la percepción al final de las percepciones
    acción ← CONSULTA(percepciones, tabla)
    devolver acción
```

Figura 2.7 El programa AGENTE-DIRIGIDO-MEDIANTE TABLA se invoca con cada nueva percepción y devuelve una acción en cada momento. Almacena la secuencia de percepciones utilizando su propia estructura de datos privada.

net contiene implementaciones en lenguajes de programación reales. Por ejemplo, la Figura 2.7 muestra un programa de agente muy sencillo que almacena la secuencia de percepciones y después las compara con las secuencias almacenadas en la tabla de acciones para decidir qué hacer. La tabla representa explícitamente la función que define el programa del agente. Para construir un agente racional de esta forma, los diseñadores deben realizar una tabla que contenga las acciones apropiadas para cada secuencia posible de percepciones.

Intuitivamente se puede apreciar por qué la propuesta de dirección-mediante-tabla para la construcción de agentes está condenada al fracaso. Sea P el conjunto de posibles percepciones y T el tiempo de vida del agente (el número total de percepciones que recibirá). La tabla de búsqueda contendrá $\sum_{t=1}^{T} |P|^t$ entradas. Si consideramos ahora el taxi automatizado: la entrada visual de una cámara individual es de 27 megabytes por segundo (30 fotografías por segundo, 640 × 480 pixels con 24 bits de información de colores). Lo cual genera una tabla de búsqueda con más de $10^{250.000.000.000}$ entradas por hora de conducción. Incluso la tabla de búsqueda del ajedrez (un fragmento del mundo real pequeño y obediente) tiene por lo menos 10^{150} entradas. El tamaño exageradamente grande de estas tablas (el número de átomos en el universo observable es menor que 10^{80}) significa que (a) no hay agente físico en este universo que tenga el espacio suficiente como para almacenar la tabla, (b) el diseñador no tendrá tiempo para crear la tabla, (c) ningún agente podría aprender todas las entradas de la tabla a partir de su experiencia, y (d) incluso si el entorno es lo suficientemente simple para generar una tabla de un tamaño razonable, el diseñador no tiene quien le asesore en la forma en la que rellenar la tabla.

A pesar de todo ello, el AGENTE-DIRIGIDO-MEDIANTE TABLA *hace* lo que nosotros queremos: implementa la función deseada para el agente. El desafío clave de la IA es encontrar la forma de escribir programas, que en la medida de lo posible, reproduzcan un comportamiento racional a partir de una pequeña cantidad de código en vez de a partir de una tabla con un gran número de entradas. Existen bastantes ejemplos que muestran qué se puede hacer con éxito en otras áreas: por ejemplo, las grandes tablas de las raíces cuadradas utilizadas por ingenieros y estudiantes antes de 1970 se han reemplazado por un programa de cinco líneas que implementa el método de Newton en las calculadoras electrónicas. La pregunta es, en el caso del comportamiento inteligente general, ¿puede la IA hacer lo que Newton hizo con las raíces cuadradas? Creemos que la respuesta es afirmativa.

En lo que resta de esta sección se presentan los cuatro tipos básicos de programas para agentes que encarnan los principios que subyacen en casi todos los sistemas inteligentes.

- Agentes reactivos simples.
- Agentes reactivos basados en modelos.
- Agentes basados en objetivos.
- Agentes basados en utilidad.

Después se explica, en términos generales, cómo convertir todos ellos en *agentes que aprendan*.

Agentes reactivos simples

AGENTE REACTIVO SIMPLE

El tipo de agente más sencillo es el **agente reactivo simple**. Estos agentes seleccionan las acciones sobre la base de las percepciones *actuales*, ignorando el resto de las percepciones históricas. Por ejemplo, el agente aspiradora cuya función de agente se presentó en la Figura 2.3 es un agente reactivo simple porque toma sus decisiones sólo con base en la localización actual y si ésta está sucia. La Figura 2.8 muestra el programa para este agente.

Hay que tener en cuenta que el programa para el agente aspiradora es muy pequeño comparado con su tabla correspondiente. La reducción más clara se obtiene al ignorar la historia de percepción, que reduce el número de posibilidades de 4^T a sólo 4. Otra reducción se basa en el hecho de que cuando la cuadrícula actual está sucia, la acción no depende de la localización.

Imagínese que es el conductor del taxi automático. Si el coche que circula delante frena, y las luces de freno se encienden, entonces lo advertiría y comenzaría a frenar. En otras palabras, se llevaría a cabo algún tipo de procesamiento sobre las señales visuales para establecer la condición que se llama «El coche que circula delante está frenando». Esto dispara algunas conexiones establecidas en el programa del agente para que se ejecute la acción «iniciar frenado». Esta conexión se denomina **regla de condición-acción**[8], y se representa por

REGLA DE CONDICIÓN-ACCIÓN

si *el-coche-que-circula-delante-está-frenando* **entonces** *iniciar-frenada*.

función AGENTE-ASPIRADORA-REACTIVO([*localización*, *estado*]) **devuelve** una acción

si *estado* = *Sucio* **entonces devolver** *Aspirar*
de otra forma, si *localización* = *A* **entonces devolver** *Derecha*
de otra forma, si *localización* = *B* **entonces devolver** *Izquierda*

Figura 2.8 Programa para el agente aspiradora de reactivo simple en el entorno definido por las dos cuadrículas. Este programa implementa la función de agente presentada en la Figura 2.3.

[8] También llamadas **reglas de situación-acción**, **producciones**, o **reglas si-entonces**.

Los humanos también tienen muchas de estas conexiones, algunas de las cuales son respuestas aprendidas (como en el caso de la conducción) y otras son reacciones innatas (como parpadear cuando algo se acerca al ojo). A lo largo de esta obra, se estudiarán diferentes formas en las que se pueden aprender e implementar estas conexiones.

El programa de la Figura 2.8 es específico para el entorno concreto de la aspiradora. Una aproximación más general y flexible es la de construir primero un intérprete de propósito general para reglas de condición-acción y después crear conjuntos de reglas para entornos de trabajo específicos. La Figura 2.9 presenta la estructura de este programa general de forma esquemática, mostrando cómo las reglas de condición-acción permiten al agente generar la conexión desde las percepciones a las acciones. No se preocupe si le parece trivial; pronto se complicará. Se utilizan rectángulos para denotar el estado interno actual del proceso de toma de decisiones del agente y óvalos para representar la información base utilizada en el proceso. El programa del agente, que es también muy simple, se muestra en la Figura 2.10. La función INTERPRETAR-ENTRADA genera una descripción abstracta del estado actual a partir de la percepción, y la función REGLA-COINCIDENCIA devuelve la primera regla del conjunto de reglas que coincide con la descripción del estado dada. Hay que tener en cuenta que la descripción en términos de «reglas»

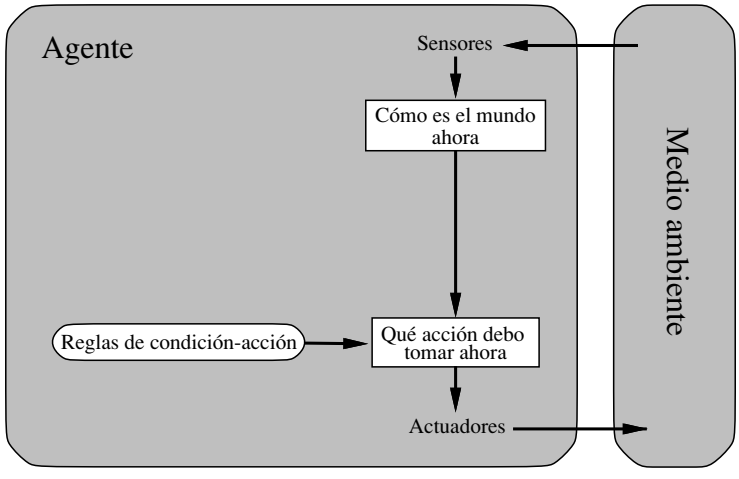

Figura 2.9 Diagrama esquemático de un agente reactivo simple.

```
función AGENTE-REACTIVO-SIMPLE(percepción) devuelve una acción
    estático: reglas, un conjunto de reglas condición-acción

    estado ← INTERPRETAR-ENTRADA(percepción)
    regla ← REGLA-COINCIDENCIA(estado, reglas)
    acción ← REGLA-ACCIÓN[regla]
    devolver acción
```

Figura 2.10 Un agente reactivo simple, que actúa de acuerdo a la regla cuya condición coincida con el estado actual, definido por la percepción.

y «coincidencias» es puramente conceptual; las implementaciones reales pueden ser tan simples como colecciones de puertas lógicas implementando un circuito booleano.

Los agentes reactivos simples tienen la admirable propiedad de ser simples, pero poseen una inteligencia muy limitada. El agente de la Figura 2.10 funcionará *sólo si se puede tomar la decisión correcta sobre la base de la percepción actual, lo cual es posible sólo si el entorno es totalmente observable*. Incluso el que haya una pequeña parte que no se pueda observar puede causar serios problemas. Por ejemplo, la regla de frenado dada anteriormente asume que la condición *el-coche-que-circula-delante-está-frenando* se puede determinar a partir de la percepción actual (imagen de vídeo actual) si el coche de enfrente tiene un sistema centralizado de luces de freno. Desafortunadamente, los modelos antiguos tienen diferentes configuraciones de luces traseras, luces de frenado, y de intermitentes, y no es siempre posible saber a partir de una única imagen si el coche está frenando. Un agente reactivo simple conduciendo detrás de un coche de este tipo puede frenar continuamente y de manera innecesaria, o peor, no frenar nunca.

Un problema similar aparece en el mundo de la aspiradora. Supongamos que se elimina el sensor de localización de un agente aspiradora reactivo simple, y que sólo tiene un sensor de suciedad. Un agente de este tipo tiene sólo dos percepciones posibles: [*Sucio*] y [*Limpio*]. Puede *Aspirar* cuando se encuentra con [*Sucio*]. ¿Qué debe hacer cuando se encuentra con [*Limpio*]? Si se desplaza a la *Izquierda* se equivoca (siempre) si está en la cuadrícula *A*, y si de desplaza a la *Derecha* se equivoca (siempre) si está en la cuadrícula *B*. Los bucles infinitos son a menudo inevitables para los agentes reactivos simples que operan en algunos entornos parcialmente observables.

ALEATORIO

Salir de los bucles infinitos es posible si los agentes pueden seleccionar sus acciones **aleatoriamente**. Por ejemplo, si un agente aspiradora percibe [*Limpio*], puede lanzar una moneda y elegir entre *Izquierda* y *Derecha*. Es fácil mostrar que el agente se moverá a la otra cuadrícula en una media de dos pasos. Entonces, si la cuadrícula está sucia, la limpiará y la tarea de limpieza se completará. Por tanto, un agente reactivo simple con capacidad para elegir acciones de manera aleatoria puede mejorar los resultados que proporciona un agente reactivo simple determinista.

En la Sección 2.3 se mencionó que un comportamiento aleatorio de un tipo adecuado puede resultar racional en algunos entornos multiagente. En entornos de agentes individuales, el comportamiento aleatorio *no* es normalmente racional. Es un truco útil que ayuda a los agentes reactivos simples en algunas situaciones, pero en la mayoría de los casos se obtendrán mejores resultados con agentes deterministas más sofisticados.

Agentes reactivos basados en modelos

La forma más efectiva que tienen los agentes de manejar la visibilidad parcial es *almacenar información de las partes del mundo que no pueden ver*. O lo que es lo mismo,

ESTADO INTERNO

el agente debe mantener algún tipo de **estado interno** que dependa de la historia percibida y que de ese modo refleje por lo menos alguno de los aspectos no observables del estado actual. Para el problema de los frenos, el estado interno no es demasiado extenso, sólo la fotografía anterior de la cámara, facilitando al agente la detección de dos luces rojas encendiéndose y apagándose simultáneamente a los costados del vehículo. Para

otros aspectos de la conducción, como un cambio de carril, el agente tiene que mantener información de la posición del resto de los coches si no los puede ver.

La actualización de la información de estado interno según pasa el tiempo requiere codificar dos tipos de conocimiento en el programa del agente. Primero, se necesita alguna información acerca de cómo evoluciona el mundo independientemente del agente, por ejemplo, que un coche que está adelantando estará más cerca, detrás, que en un momento inmediatamente anterior. Segundo, se necesita más información sobre cómo afectan al mundo las acciones del agente, por ejemplo, que cuando el agente gire hacia la derecha, el coche gira hacia la derecha o que después de conducir durante cinco minutos hacia el norte en la autopista se avanzan cinco millas hacia el norte a partir del punto en el que se estaba cinco minutos antes. Este conocimiento acerca de «cómo funciona el mundo», tanto si está implementado con un circuito booleano simple o con teorías científicas completas, se denomina **modelo** del mundo. Un agente que utilice este modelo es un **agente basado en modelos**.

AGENTE BASADO EN MODELOS

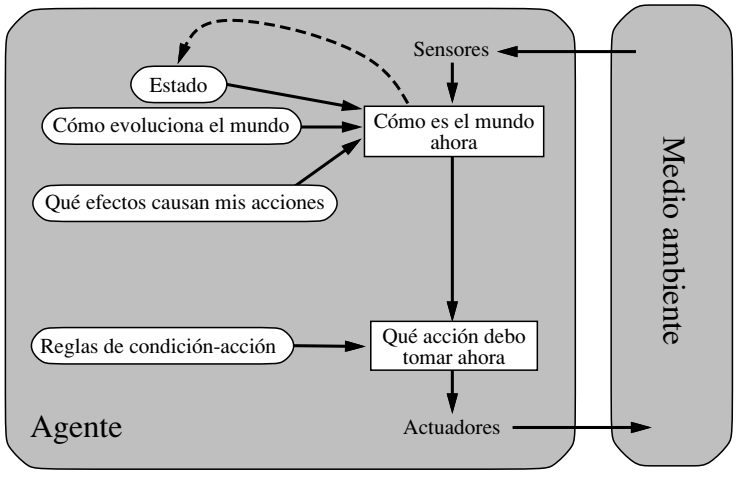

Figura 2.11 Un agente reactivo basado en modelos.

```
función AGENTE-REACTIVO-CON-ESTADO(percepción) devuelve una acción
    estático: estado, una descripción actual del estado del mundo
              reglas, un conjunto de reglas condición-acción
              acción, la acción más reciente, inicialmente ninguna

    estado ← ACTUALIZAR-ESTADO(estado, acción, percepción)
    regla ← REGLA-COINCIDENCIA(estado, reglas)
    acción ← REGLA-ACCIÓN[regla]
    devolver acción
```

Figura 2.12 Un agente reactivo basado en modelos, que almacena información sobre el estado actual del mundo utilizando un modelo interno. Después selecciona una acción de la misma forma que el agente reactivo.

La Figura 2.11 proporciona la estructura de un agente reactivo simple con estado interno, muestra cómo la percepción actual se combina con el estado interno antiguo para generar la descripción actualizada del estado actual. La Figura 2.12 muestra el programa del agente. La parte interesante es la correspondiente a la función Actualizar-Estado, que es la responsable de la creación de la nueva descripción del estado interno. Además de interpretar la nueva percepción a partir del conocimiento existente sobre el estado, utiliza información relativa a la forma en la que evoluciona el mundo para conocer más sobre las partes del mundo que no están visibles; para ello debe conocer cuál es el efecto de las acciones del agente sobre el estado del mundo. Los Capítulos 10 y 17 ofrecen ejemplos detallados.

Agentes basados en objetivos

El conocimiento sobre el estado actual del mundo no es siempre suficiente para decidir qué hacer. Por ejemplo, en un cruce de carreteras, el taxista puede girar a la izquierda, girar a la derecha o seguir hacia adelante. La decisión correcta depende de dónde quiere ir el taxi. En otras palabras, además de la descripción del estado actual, el agente necesita algún tipo de información sobre su **meta** que describa las situaciones que son deseables, por ejemplo, llegar al destino propuesto por el pasajero. El programa del agente se puede combinar con información sobre los resultados de las acciones posibles (la misma información que se utilizó para actualizar el estado interno en el caso del agente reflexivo) para elegir las acciones que permitan alcanzar el objetivo. La Figura 2.13 muestra la estructura del agente basado en objetivos.

META

En algunas ocasiones, la selección de acciones basadas en objetivos es directa, cuando alcanzar los objetivos es el resultado inmediato de una acción individual. En otras oca-

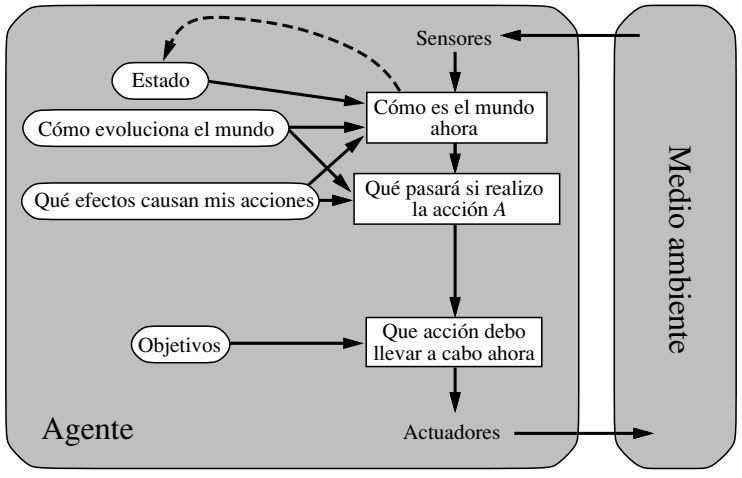

Figura 2.13 Un agente basado en objetivos y basado en modelos, que almacena información del estado del mundo así como del conjunto de objetivos que intenta alcanzar, y que es capaz de seleccionar la acción que eventualmente lo guiará hacia la consecución de sus objetivos.

siones, puede ser más complicado, cuando el agente tiene que considerar secuencias complejas para encontrar el camino que le permita alcanzar el objetivo. **Búsqueda** (Capítulos del 3 al 6) y **planificación** (Capítulos 11 y 12) son los subcampos de la IA centrados en encontrar secuencias de acciones que permitan a los agentes alcanzar sus metas.

Hay que tener en cuenta que la toma de decisiones de este tipo es fundamentalmente diferente de las reglas de condición–acción descritas anteriormente, en las que hay que tener en cuenta consideraciones sobre el futuro (como «¿qué pasará si yo hago esto y esto?» y «¿me hará esto feliz?»). En los diseños de agentes reactivos, esta información no está representada explícitamente, porque las reglas que maneja el agente proyectan directamente las percepciones en las acciones. El agente reactivo frena cuando ve luces de freno. Un agente basado en objetivos, en principio, puede razonar que si el coche que va delante tiene encendidas las luces de frenado, está reduciendo su velocidad. Dada la forma en la que el mundo evoluciona normalmente, la única acción que permite alcanzar la meta de no chocarse con otros coches, es frenar.

Aunque el agente basado en objetivos pueda parecer menos eficiente, es más flexible ya que el conocimiento que soporta su decisión está representado explícitamente y puede modificarse. Si comienza a llover, el agente puede actualizar su conocimiento sobre cómo se comportan los frenos; lo cual implicará que todas las formas de actuar relevantes se alteren automáticamente para adaptarse a las nuevas circunstancias. Para el agente reactivo, por otro lado, se tendrán que rescribir muchas reglas de condición-acción. El comportamiento del agente basado en objetivos puede cambiarse fácilmente para que se dirija a una localización diferente. Las reglas de los agentes reactivos relacionadas con cuándo girar y cuándo seguir recto son válidas sólo para un destino concreto y tienen que modificarse cada vez que el agente se dirija a cualquier otro lugar distinto.

Agentes basados en utilidad

Las metas por sí solas no son realmente suficientes para generar comportamiento de gran calidad en la mayoría de los entornos. Por ejemplo, hay muchas secuencias de acciones que llevarán al taxi a su destino (y por tanto a alcanzar su objetivo), pero algunas son más rápidas, más seguras, más fiables, o más baratas que otras. Las metas sólo proporcionan una cruda distinción binaria entre los estados de «felicidad» y «tristeza», mientras que una medida de eficiencia más general debería permitir una comparación entre estados del mundo diferentes de acuerdo al nivel exacto de felicidad que el agente alcance cuando se llegue a un estado u otro. Como el término «felicidad» no suena muy científico, la terminología tradicional utilizada en estos casos para indicar que se prefiere un estado del mundo a otro es que un estado tiene más **utilidad** que otro para el agente[9].

UTILIDAD

FUNCIÓN DE UTILIDAD

Una **función de utilidad** proyecta un estado (o una secuencia de estados) en un número real, que representa un nivel de felicidad. La definición completa de una función de utilidad permite tomar decisiones racionales en dos tipos de casos en los que las metas son inadecuadas. Primero, cuando haya objetivos conflictivos, y sólo se puedan al-

[9] La palabra «utilidad» aquí se refiere a «la cualidad de ser útil».

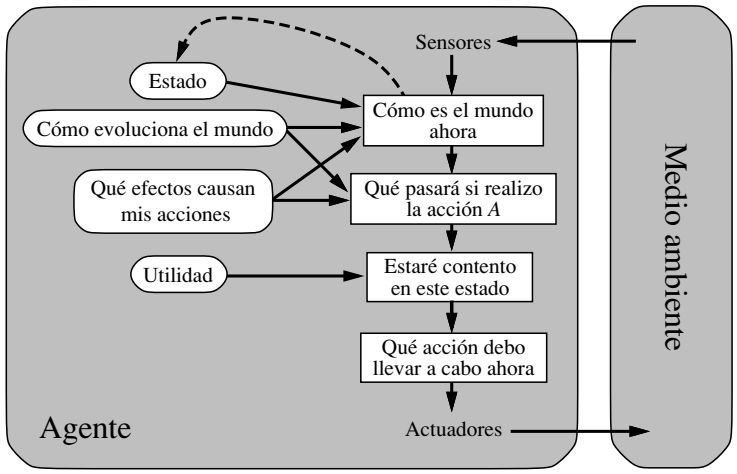

Figura 2.14 Un agente basado en utilidad y basado en modelos. Utiliza un modelo del mundo, junto con una función de utilidad que calcula sus preferencias entre los estados del mundo. Después selecciona la acción que le lleve a alcanzar la mayor utilidad esperada, que se calcula haciendo la media de todos los estados resultantes posibles, ponderado con la probabilidad del resultado.

canzar algunos de ellos (por ejemplo, velocidad y seguridad), la función de utilidad determina el equilibrio adecuado. Segundo, cuando haya varios objetivos por los que se pueda guiar el agente, y ninguno de ellos se pueda alcanzar con certeza, la utilidad proporciona un mecanismo para ponderar la probabilidad de éxito en función de la importancia de los objetivos.

En el Capítulo 16, se mostrará cómo cualquier agente racional debe comportarse *como si* tuviese una función de utilidad cuyo valor esperado tiene que maximizar. Por tanto, un agente que posea una función de utilidad *explícita* puede tomar decisiones racionales, y lo puede hacer con la ayuda de un algoritmo de propósito general que no dependa de la función específica de utilidad a maximizar. De esta forma, la definición «global» de racionalidad (identificando como racionales aquellas funciones de los agentes que proporcionan el mayor rendimiento) se transforma en una restricción «local» en el diseño de agentes racionales que se puede expresar con un simple programa.

La Figura 2.14 muestra la estructura de un agente basado en utilidad. En la Parte IV aparecen programas de agentes basados en utilidad, donde se presentan agentes que toman decisiones y que deben trabajar con la incertidumbre inherente a los entornos parcialmente observables.

Agentes que aprenden

Se han descrito programas para agentes que poseen varios métodos para seleccionar acciones. Hasta ahora no se ha explicado cómo *poner en marcha* estos programas de agentes. Turing (1950), en su temprano y famoso artículo, consideró la idea de programar sus máquinas inteligentes a mano. Estimó cuánto tiempo podía llevar y concluyó que «Se-

ría deseable utilizar algún método más rápido». El método que propone es construir máquinas que aprendan y después enseñarlas. En muchas áreas de IA, éste es ahora el método más adecuado para crear sistemas novedosos. El aprendizaje tiene otras ventajas, como se ha explicado anteriormente: permite que el agente opere en medios inicialmente desconocidos y que sea más competente que si sólo utilizase un conocimiento inicial. En esta sección, se introducen brevemente las principales ideas en las que se basan los agentes que aprenden. En casi todos los capítulos de este libro se comentan las posibilidades y métodos de aprendizaje de tipos de agentes concretos. La Parte VI profundiza más en los algoritmos de aprendizaje en sí mismos.

ELEMENTO DE APRENDIZAJE

ELEMENTO DE ACTUACIÓN

CRÍTICA

Un agente que aprende se puede dividir en cuatro componentes conceptuales, tal y como se muestra en la Figura 2.15. La distinción más importante entre el **elemento de aprendizaje** y el **elemento de actuación** es que el primero está responsabilizado de hacer mejoras y el segundo se responsabiliza de la selección de acciones externas. El elemento de actuación es lo que anteriormente se había considerado como el agente completo: recibe estímulos y determina las acciones a realizar. El elemento de aprendizaje se realimenta con las **críticas** sobre la actuación del agente y determina cómo se debe modificar el elemento de actuación para proporcionar mejores resultados en el futuro.

El diseño del elemento de aprendizaje depende mucho del diseño del elemento de actuación. Cuando se intenta diseñar un agente que tenga capacidad de aprender, la primera cuestión a solucionar no es ¿cómo se puede enseñar a aprender?, sino ¿qué tipo de elemento de actuación necesita el agente para llevar a cabo su objetivo, cuando haya aprendido cómo hacerlo? Dado un diseño para un agente, se pueden construir los mecanismos de aprendizaje necesarios para mejorar cada una de las partes del agente.

La crítica indica al elemento de aprendizaje qué tal lo está haciendo el agente con respecto a un nivel de actuación fijo. La crítica es necesaria porque las percepciones por sí mismas no prevén una indicación del éxito del agente. Por ejemplo, un programa de

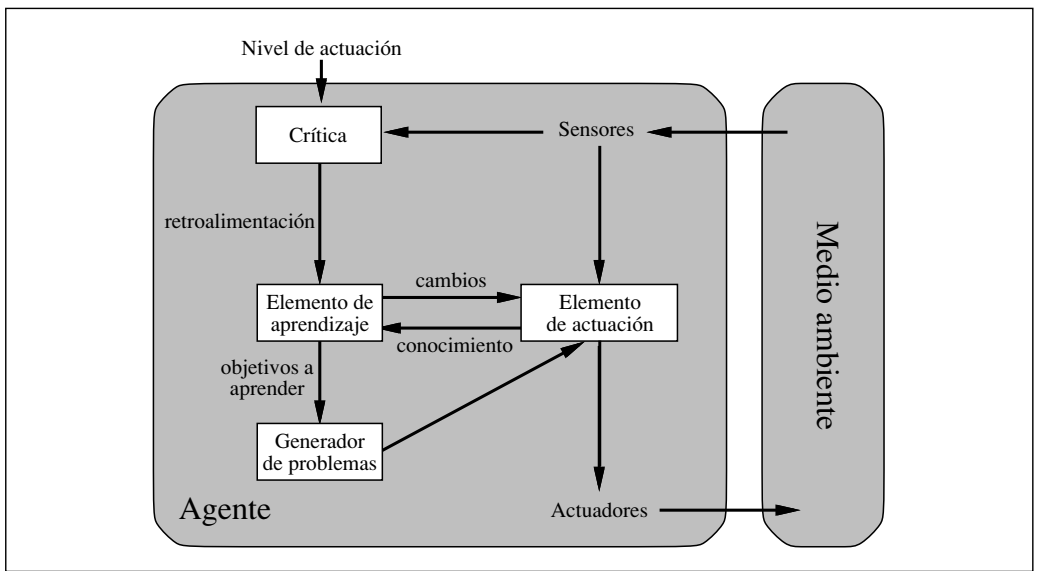

Figura 2.15 Modelo general para agentes que aprenden.

ajedrez puede recibir una percepción indicando que ha dado jaque mate a su oponente, pero necesita tener un nivel de actuación que le indique que ello es bueno; la percepción por sí misma no lo indica. Es por tanto muy importante fijar el nivel de actuación. Conceptualmente, se debe tratar con él como si estuviese fuera del agente, ya que éste no debe modificarlo para satisfacer su propio interés.

GENERADOR DE PROBLEMAS

El último componente del agente con capacidad de aprendizaje es el **generador de problemas**. Es responsable de sugerir acciones que lo guiarán hacia experiencias nuevas e informativas. Lo interesante es que si el elemento de actuación sigue su camino, puede continuar llevando a cabo las acciones que sean mejores, dado su conocimiento. Pero si el agente está dispuesto a explorar un poco, y llevar a cabo algunas acciones que no sean totalmente óptimas a corto plazo, puede descubrir acciones mejores a largo plazo. El trabajo del generador de problemas es sugerir estas acciones exploratorias. Esto es lo que los científicos hacen cuando llevan a cabo experimentos. Galileo no pensaba que tirar piedras desde lo alto de una torre en Pisa tenía un valor por sí mismo. Él no trataba de romper piedras ni de cambiar la forma de pensar de transeúntes desafortunados que paseaban por el lugar. Su intención era adaptar su propia mente, para identificar una teoría que definiese mejor el movimiento de los objetos.

Para concretar el diseño total, se puede volver a utilizar el ejemplo del taxi automatizado. El elemento de actuación consiste en la colección de conocimientos y procedimientos que tiene el taxi para seleccionar sus acciones de conducción. El taxi se pone en marcha y circula utilizando este elemento de actuación. La crítica observa el mundo y proporciona información al elemento de aprendizaje. Por ejemplo, después de que el taxi se sitúe tres carriles hacia la izquierda de forma rápida, la crítica observa el lenguaje escandaloso que utilizan otros conductores. A partir de esta experiencia, el elemento de aprendizaje es capaz de formular una regla que indica que ésta fue una mala acción, y el elemento de actuación se modifica incorporando la nueva regla. El generador de problemas debe identificar ciertas áreas de comportamiento que deban mejorarse y sugerir experimentos, como probar los frenos en carreteras con tipos diferentes de superficies y bajo condiciones distintas.

El elemento de aprendizaje puede hacer cambios en cualquiera de los componentes de «conocimiento» que se muestran en los diagramas de agente (Figuras 2.9, 2.11, 2.13, y 2.14). Los casos más simples incluyen el aprendizaje directo a partir de la secuencia percibida. La observación de pares de estados sucesivos del entorno puede permitir que el agente aprenda «cómo evoluciona el mundo», y la observación de los resultados de sus acciones puede permitir que el agente aprenda «qué hacen sus acciones». Por ejemplo, si el taxi ejerce una cierta presión sobre los frenos cuando está circulando por una carretera mojada, acto seguido conocerá cómo decelera el coche. Claramente, estas dos tareas de aprendizaje son más difíciles si sólo existe una vista parcial del medio.

Las formas de aprendizaje mostradas en los párrafos precedentes no necesitan el acceso a niveles de actuación externo, de alguna forma, el nivel es el que se utiliza universalmente para hacer pronósticos de acuerdo con la experimentación. La situación es ligeramente más compleja para un agente basado en utilidad que desee adquirir información para crear su función de utilidad. Por ejemplo, se supone que el agente conductor del taxi no recibe propina de los pasajeros que han recorrido un trayecto de forma incómoda debido a una mala conducción. El nivel de actuación externo debe informar

al agente de que la pérdida de propinas tiene una contribución negativa en su nivel de actuación medio; entonces el agente puede aprender que «maniobras violentas no contribuyen a su propia utilidad». De alguna manera, el nivel de actuación identifica parte de las percepciones entrantes como **recompensas** (o **penalizaciones**) que generan una respuesta directa en la calidad del comportamiento del agente. Niveles de actuación integrados como el dolor y el hambre en animales se pueden enmarcar en este contexto. El Capítulo 21 discute estos asuntos.

En resumen, los agentes tienen una gran variedad de componentes, y estos componentes se pueden representar de muchas formas en los programas de agentes, por lo que, parece haber una gran variedad de métodos de aprendizaje. Existe, sin embargo, una visión unificada sobre un tema fundamental. El aprendizaje en el campo de los agentes inteligentes puede definirse como el proceso de modificación de cada componente del agente, lo cual permite a cada componente comportarse más en consonancia con la información que se recibe, lo que por tanto permite mejorar el nivel medio de actuación del agente.

2.5 Resumen

En este capítulo se ha realizado un recorrido rápido por el campo de la IA, que se ha presentado como la ciencia del diseño de los agentes. Los puntos más importantes a tener en cuenta son:

- Un **agente** es algo que percibe y actúa en un medio. La **función del agente** para un agente especifica la acción que debe realizar un agente como respuesta a cualquier secuencia percibida.
- La **medida de rendimiento** evalúa el comportamiento del agente en un medio. Un **agente racional** actúa con la intención de maximizar el valor esperado de la medida de rendimiento, dada la secuencia de percepciones que ha observado hasta el momento.
- Las especificaciones del **entorno de trabajo** incluyen la medida de rendimiento, el medio externo, los actuadores y los sensores. El primer paso en el diseño de un agente debe ser siempre la especificación, tan completa como sea posible, del entorno de trabajo.
- El entorno de trabajo varía según distintos parámetros. Pueden ser total o parcialmente visibles, deterministas o estocásticos, episódicos o secuenciales, estáticos o dinámicos, discretos o continuos, y formados por un único agente o por varios agentes.
- El **programa del agente** implementa la función del agente. Existe una gran variedad de diseños de programas de agentes, y reflejan el tipo de información que se hace explícita y se utiliza en el proceso de decisión. Los diseños varían en eficiencia, solidez y flexibilidad. El diseño apropiado del programa del agente depende en gran medida de la naturaleza del medio.
- **Los gentes reactivos simples** responden directamente a las percepciones, mientras que los **agentes reactivos basados en modelos** mantienen un estado interno

que les permite seguir el rastro de aspectos del mundo que no son evidentes según las percepciones actuales. Los agentes basados en objetivos actúan con la intención de alcanzar sus metas, y los agentes basados en utilidad intentan maximizar su «felicidad» deseada.

- Todos los agentes pueden mejorar su eficacia con la ayuda de mecanismos de **aprendizaje**.

NOTAS BIBLIOGRÁFICAS E HISTÓRICAS

El papel central de la acción en la inteligencia (la noción del razonamiento práctico) se remonta por lo menos a la obra *Nicomachean Ethics* de Aristóteles. McCarthy (1958) trató también el tema del razonamiento práctico en su influyente artículo *Programs with Common Sense*. Los campos de la robótica y la teoría de control tienen interés, por su propia naturaleza, en la construcción de agentes físicos. El concepto de un **controlador**, en el ámbito de la teoría de control, es idéntico al de un agente en IA. Quizá sorprendentemente, la IA se ha concentrado durante la mayor parte de su historia en componentes aislados de agentes (sistemas que responden a preguntas, demostración de teoremas, sistemas de visión, y demás) en vez de en agentes completos. La discusión sobre agentes que se presenta en el libro de Genesereth y Nilsson (1987) fue una influyente excepción. El concepto de agente en sí está aceptado ampliamente ahora en el campo y es un tema central en libros recientes (Poole *et al*., 1998; Nilsson, 1998).

CONTROLADOR

El Capítulo 1 muestra las raíces del concepto de racionalidad en la Filosofía y la Economía. En la IA, el concepto tuvo un interés periférico hasta mediados de los 80, donde comenzó a suscitar muchas discusiones sobre los propios fundamentos técnicos del campo. Un artículo de Jon Doyle (1983) predijo que el diseño de agentes racionales podría llegar a ser la misión central de la IA, mientras otras áreas populares podrían separarse dando lugar a nuevas disciplinas.

Es muy importante tener muy en cuenta las propiedades del medio y sus consecuencias cuando se realiza el diseño de los agentes racionales ya que forma parte de la tradición ligada a la teoría de control [por ejemplo los sistemas de control clásicos (Dorf y Bishop, 1999) manejan medios deterministas y totalmente observables; el control óptimo estocástico (Kumar y Varaiya, 1986) maneja medios parcialmente observables y estocásticos y un control híbrido (Henzinger y Sastry, 1998) maneja entornos que contienen elementos discretos y continuos]. La distinción entre entornos totalmente y parcialmente observables es también central en la literatura sobre **programación dinámica** desarrollada en el campo de la investigación operativa (Puterman, 1994), como se comentará en el Capítulo 17.

Los agentes reactivos fueron los primeros modelos para psicólogos conductistas como Skinner (1953), que intentó reducir la psicología de los organismos estrictamente a correspondencias entrada/salida o estímulo/respuesta. La evolución del behaviourismo hacia el funcionalismo en el campo de la psicología, que estuvo, al menos de forma parcial, dirigida por la aplicación de la metáfora del computador a los agentes (Putnam, 1960; Lewis, 1966) introdujo el estado interno del agente en el nuevo escenario. La mayor par-

te del trabajo realizado en el campo de la IA considera que los agentes reactivos puros con estado interno son demasiado simples para ser muy influyentes, pero los trabajos de Rosenschein (1985) y Brooks (1986) cuestionan esta hipótesis (*véase* el Capítulo 25). En los últimos años, se ha trabajado intensamente para encontrar algoritmos eficientes capaces de hacer un buen seguimiento de entornos complejos (Hamscher *et al*., 1992). El programa del Agente Remoto que controla la nave espacial Deep Space One (descrito en la página 27) es un admirable ejemplo concreto (Muscettola *et al*., 1998; Jonsson *et al*., 2000).

Los agentes basados en objetivos están presentes tanto en las referencias de Aristóteles sobre el razonamiento práctico como en los primeros artículos de McCarthy sobre IA lógica. El robot Shakey (Fikes y Nilsson, 1971; Nilsson, 1984) fue el primer robot construido como un agente basado en objetivos. El análisis lógico completo de un agente basado en objetivos aparece en Genesereth y Nilsson (1987), y Shoham (1993) ha desarrollado una metodología de programación basada en objetivos llamada programación orientada a agentes.

La perspectiva orientada a objetivos también predomina en la psicología cognitiva tradicional, concretamente en el área de la resolución de problemas, como se muestra tanto en el influyente *Human Problem Solving* (Newell y Simon, 1972) como en los últimos trabajos de Newell (1990). Los objetivos, posteriormente definidos como *deseos* (generales) y las *intenciones* (perseguidas en un momento dado), son fundamentales en la teoría de agentes desarrollada por Bratman (1987). Esta teoría ha sido muy influyente tanto en el entendimiento del lenguaje natural como en los sistemas multiagente.

Horvitz *et al*. (1988) sugieren específicamente el uso de la maximización de la utilidad esperada concebida racionalmente como la base de la IA. El texto de Pearl (1988) fue el primero en IA que cubrió las teorías de la probabilidad y la utilidad en profundidad; su exposición de métodos prácticos de razonamiento y toma de decisiones con incertidumbre fue, posiblemente, el factor individual que más influyó en el desarrollo de los agentes basados en utilidad en los 90 (*véase* la Parte V).

El diseño general de agentes que aprenden representado en la Figura 2.15 es un clásico de la literatura sobre aprendizaje automático (Buchanan *et al*., 1978; Mitchell, 1997). Ejemplos de diseños, implementados en programas, se remontan, como poco, hasta los programas que aprendían a jugar al ajedrez de Arthur Samuel (1959, 1967). La Parte VI está dedicada al estudio en profundidad de los agentes que aprenden.

El interés en los agentes y en el diseño de agentes ha crecido rápidamente en los últimos años, en parte por la expansión de Internet y la necesidad observada de desarrollar **softbots** (robots *software*) automáticos y móviles (Etzioni y Weld, 1994). Artículos relevantes pueden encontrarse en *Readings in Agents* (Huhns y Singh, 1998) y *Fundations of Rational Agency* (Wooldridge y Rao, 1999). *Multiagent Systems* (Weiss, 1999) proporciona una base sólida para muchos aspectos del diseño de agentes. Conferencias dedicadas a agentes incluyen la International Conference on Autonomous Agents, la International Workshop on Agent Theories, Architectures, and Languages, y la International Conference on Multiagent Systems. Finalmente, *Dung Beetle Ecology* (Hanski y Cambefort, 1991) proporciona gran cantidad de información interesante sobre el comportamiento de los escarabajos estercoleros.

EJERCICIOS

2.1 Defina con sus propias palabras los siguientes términos: agente, función de agente, programa de agente, racionalidad, autonomía, agente reactivo, agente basado en modelo, agente basado en objetivos, agente basado en utilidad, agente que aprende.

2.2 Tanto la medida de rendimiento como la función de utilidad miden la eficiencia del agente. Explique la diferencia entre los dos conceptos.

2.3 Este ejercicio explora las diferencias entre las funciones de los agentes y los programas de los agentes.

- ***a***) ¿Puede haber más de un programa de agente que implemente una función de agente dada? Proponga un ejemplo, o muestre por qué una no es posible.
- ***b***) ¿Hay funciones de agente que no se pueden implementar con algún programa de agente?
- ***c***) Dada una arquitectura máquina, ¿implementa cada programa de agente exactamente una función de agente?
- ***d***) Dada una arquitectura con n bits de almacenamiento, ¿cuántos posibles programas de agente diferentes puede almacenar?

2.4 Examínese ahora la racionalidad de varias funciones de agentes aspiradora.

- ***a***) Muestre que la función de agente aspiradora descrita en la Figura 2.3 es realmente racional bajo la hipótesis presentada en la página 36.
- ***b***) Describa una función para un agente racional cuya medida de rendimiento modificada deduzca un punto por cada movimiento. ¿Requiere el correspondiente programa de agente estado interno?
- ***c***) Discuta posibles diseños de agentes para los casos en los que las cuadrículas limpias puedan ensuciarse y la geografía del medio sea desconocida. ¿Tiene sentido que el agente aprenda de su experiencia en estos casos? ¿Si es así, qué debe aprender?

2.5 Identifique la descripción REAS que define el entorno de trabajo para cada uno de los siguientes agentes:

- ***a***) Robot que juega al fútbol;
- ***b***) Agente para comprar libros en Internet;
- ***c***) Explorador autónomo de Marte;
- ***d***) Asistente matemático para la demostración de teoremas.

2.6 Para cada uno de los tipos de agente enumerados en el Ejercicio 2.5, caracterice el medio de acuerdo con las propiedades dadas en la Sección 2.3, y seleccione un diseño de agente adecuado.

Los siguientes ejercicios están relacionados con la implementación de entornos y agentes para el mundo de la aspiradora.

2.7 Implemente un simulador que determine la medida de rendimiento para el entorno del mundo de la aspiradora descrito en la Figura 2.2 y especificado en la página 36. La implementación debe ser modular, de forma que los sensores, actuadores, y las características del entorno (tamaño, forma, localización de la suciedad, etc.) puedan modificar-

se fácilmente. (*Nota:* hay implementaciones disponibles en el repositorio de Internet que pueden ayudar a decidir qué lenguaje de programación y sistema operativo seleccionar).

2.8 Implemente un agente reactivo simple para el entorno de la aspiradora del Ejercicio 2.7. Ejecute el simulador del entorno con este agente para todas las configuraciones iniciales posibles de suciedad y posiciones del agente. Almacene la puntuación de la actuación del agente para cada configuración y la puntuación media global.

2.9 Considere una versión modificada del entorno de la aspiradora del Ejercicio 2.7, en el que se penalice al agente con un punto en cada movimiento.

- ***a***) ¿Puede un agente reactivo simple ser perfectamente racional en este medio? Explíquese.
- ***b***) ¿Qué sucedería con un agente reactivo con estado? Diseñe este agente.
- ***c***) ¿Cómo se responderían las preguntas **a** y **b** si las percepciones proporcionan al agente información sobre el nivel de suciedad/limpieza de todas las cuadrículas del entorno?

2.10 Considere una versión modificada del entorno de la aspiradora del Ejercicio 2.7, en el que la geografía del entorno (su extensión, límites, y obstáculos) sea desconocida, así como, la disposición inicial de la suciedad. (El agente puede ir hacia *arriba*, *abajo,* así como, hacia la *derecha* y a la *izquierda*.)

- ***a***) ¿Puede un agente reactivo simple ser perfectamente racional en este medio? Explíquese.
- ***b***) ¿Puede un agente reactivo simple con una función de agente aleatoria superar a un agente reactivo simple? Diseñe un agente de este tipo y medir su rendimiento en varios medios.
- ***c***) ¿Se puede diseñar un entorno en el que el agente con la función aleatoria obtenga una actuación muy pobre? Muestre los resultados.
- ***d***) ¿Puede un agente reactivo con estado mejorar los resultados de un agente reactivo simple? Diseñe un agente de este tipo y medir su eficiencia en distintos medios. ¿Se puede diseñar un agente racional de este tipo?

2.11 Repítase el Ejercicio 2.10 para el caso en el que el sensor de localización sea reemplazado por un sensor «de golpes» que detecte si el agente golpea un obstáculo o si se sale fuera de los límites del entorno. Supóngase que el sensor de golpes deja de funcionar. ¿Cómo debe comportarse el agente?

2.12 Los entornos de la aspiradora en los ejercicios anteriores han sido todos deterministas. Discuta posibles programas de agentes para cada una de las siguientes versiones estocásticas:

- ***a***) Ley de Murphy: el 25 por ciento del tiempo, la acción de *Aspirar* falla en la limpieza del suelo si está sucio y deposita suciedad en el suelo si el suelo está limpio. ¿Cómo se ve afectado el agente si el sensor de suciedad da una respuesta incorrecta el diez por ciento de las veces?
- ***b***) Niño pequeño: en cada lapso de tiempo, cada recuadro limpio tiene un diez por ciento de posibilidad de ensuciarse. ¿Puede identificar un diseño para un agente racional en este caso?