

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*).

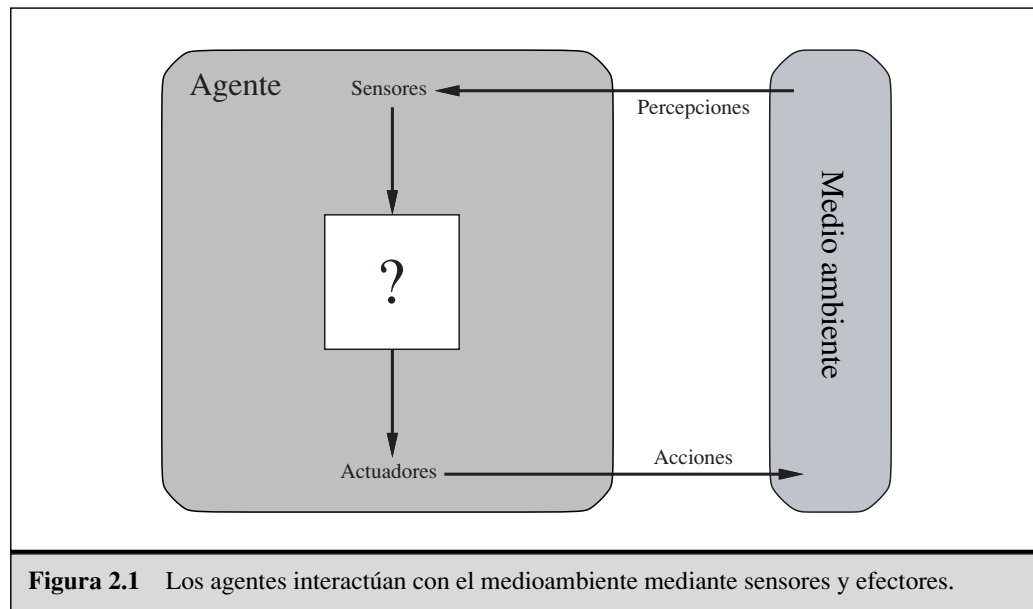


Figura 2.1 Los agentes interactúan con el medioambiente mediante sensores y efectores.

SENSOR

ACTUADOR

PERCEPCIÓN

SECUENCIA DE
PERCEPTORES



FUNCIÓN DEL AGENTE

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).

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.

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 respuesta². 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.

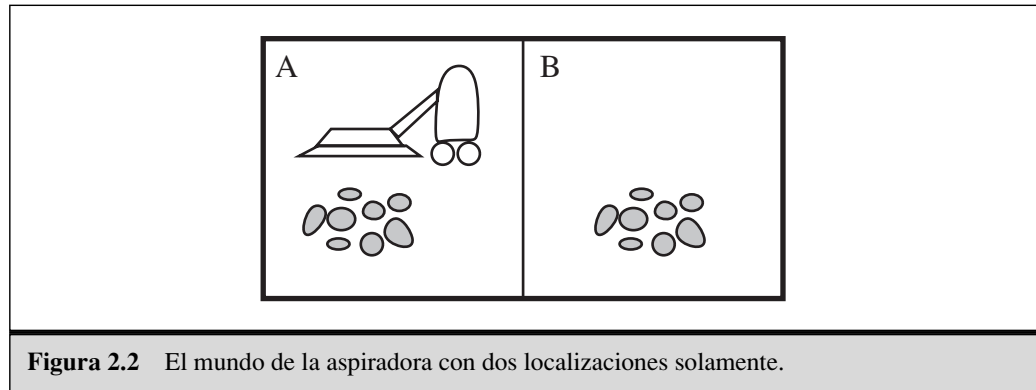


Figura 2.2 El mundo de la aspiradora con dos localizaciones solamente.

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 supelementalmente 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 mismos³. 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 avión⁴, 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 asegurarse 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.

RECOPIACIÓN DE INFORMACIÓN

EXPLORACIÓN

APRENDIZAJE

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.

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 cuenta 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.

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).

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-

AGENTES SOFTWARE

SOFTBOTS

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 las 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 OBSERVABLE

- **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 observable⁵. 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 puede 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 los 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. La 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 meramente 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 Ajedrez con reloj	Totalmente Totalmente	Determinista Estratégico	Secuencial Secuencial	Estático Semi	Discreto Discreto	Individual Multi
Póker Backgammon	Parcialmente Totalmente	Estratégico Estocástico	Secuencial Secuencial	Estático Estático	Discreto Discreto	Multi Multi
Taxi circulando Diagnóstico médico	Parcialmente Parcialmente	Estocástico Estocástico	Secuencial Secuencial	Dinámico Dinámico	Continuo Continuo	Multi Individual
Análisis de imagen Robot clasificador	Totalmente Parcialmente	Determinista Estocástico	Episódico Episódico	Semi Dinámico	Continuo Continuo	Individual Individual
Controlador de refinería Tutor interactivo de inglés	Parcialmente Parcialmente	Estocástico Estocástico	Secuencial Secuencial	Dinámico Dinámico	Continuo Discreto	Individual 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**:

$$\text{Agente} = \text{arquitectura} + \text{programa}$$

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 actuadores⁷. 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 \leftarrow 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.

Imáginese 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»

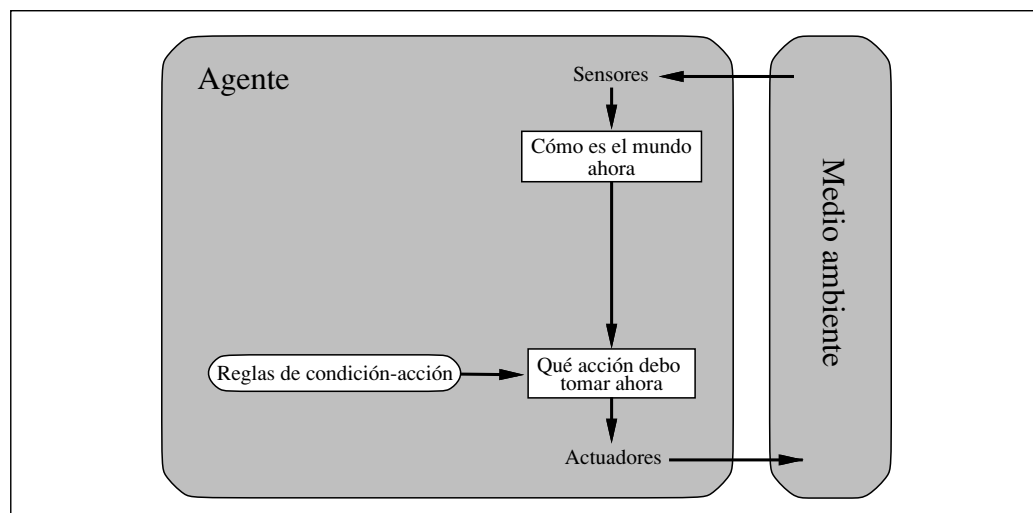


Figura 2.9 Diagrama esquemático de un agente reactivo simple.

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

estado \leftarrow INTERPRETAR-ENTRADA(*percepción*)
regla \leftarrow REGLA-COINCIDENCIA(*estado*, *reglas*)
acción \leftarrow 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 se 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, 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

ESTADO INTERNO

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

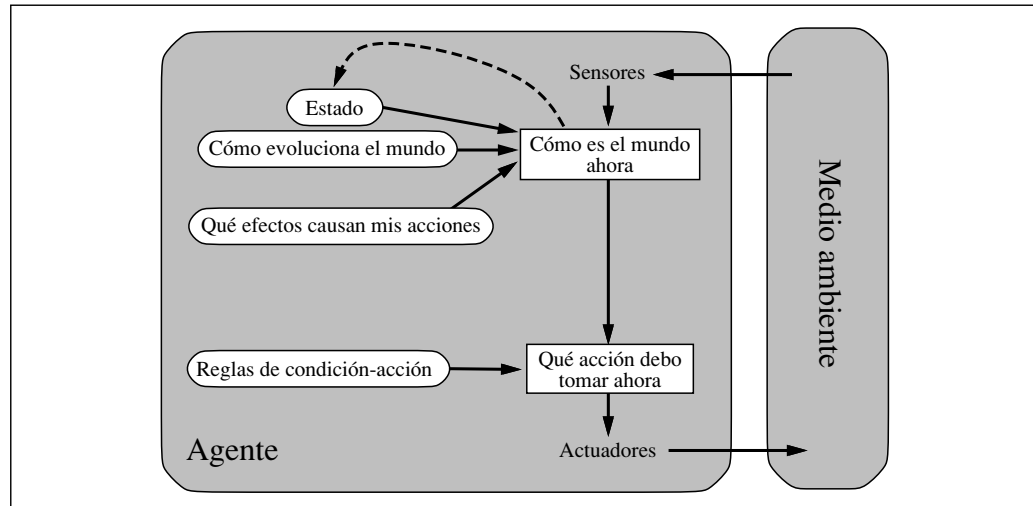


Figura 2.11 Un agente reactivo basado en modelos.

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.

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-

META

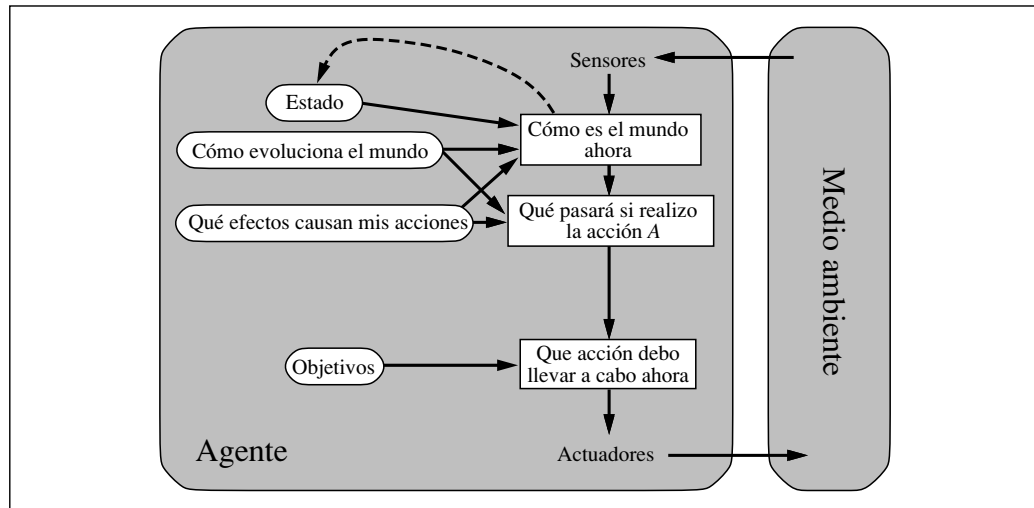


Figura 2.13 Un agente basado en objetivos y basado en modelos, que almacena información del estado del mundo así como del conjunto de objetivos que intenta alcanzar, y que es capaz de seleccionar la acción que eventualmente lo guiará hacia la consecución de sus objetivos.

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 agente⁹.

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-

UTILIDAD

FUNCIÓN DE UTILIDAD

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

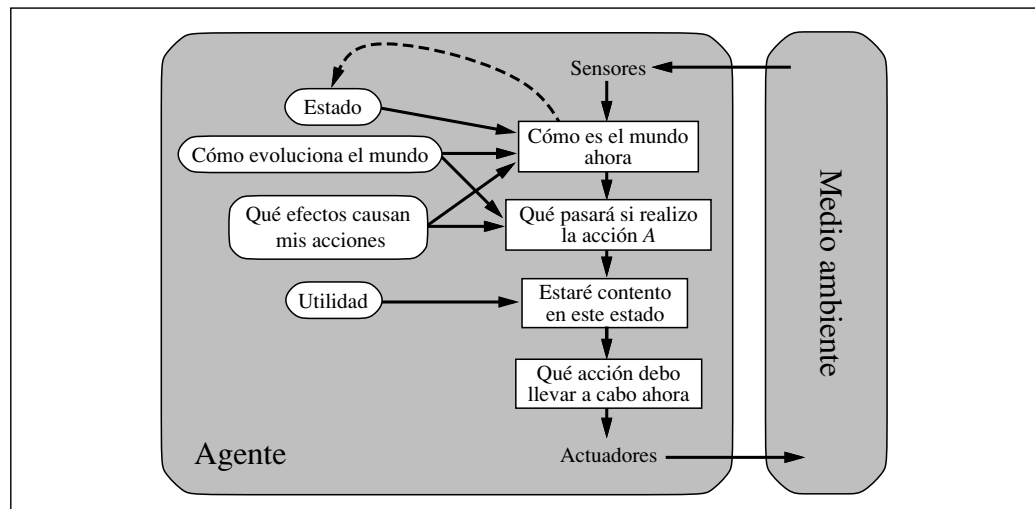


Figura 2.14 Un agente basado en utilidad y basado en modelos. Utiliza un modelo del mundo, junto con una función de utilidad que calcula sus preferencias entre los estados del mundo. Después selecciona la acción que le lleve a alcanzar la mayor utilidad esperada, que se calcula haciendo la media de todos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

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.

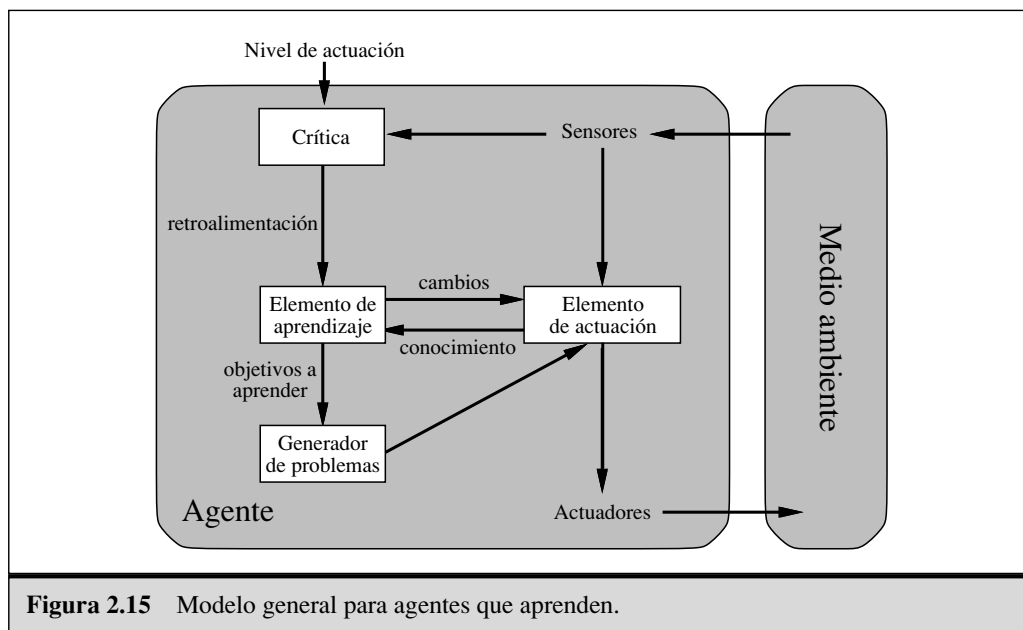
El diseño del elemento de aprendizaje depende mucho del diseño del elemento de actuación. Cuando se intenta diseñar un agente que tenga capacidad de aprender, la primera cuestión a solucionar no es ¿cómo se puede enseñar a aprender?, sino ¿qué tipo de elemento de actuación necesita el agente para llevar 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

ELEMENTO DE APRENDIZAJE

ELEMENTO DE ACTUACIÓN

CRÍTICA



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.

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**.



CONTROLADOR

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).

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 *Foundations 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 Exámínesse 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?