

Curso de Inteligencia Artificial

Alan Gerardo Reyes Figueroa

Agentes Inteligentes | Aula 03 Enero 20, 2022



Agentes

Agente = cualquier dispositivo (máquina, robot, autómata, algoritmo) que percibe su ambiente y actúa.

Ambiente = contexto o espacio donde se desarrolla el agente

Un agente, para actuar, debe poseer:

- Sensores = todo lo que le sirva para percibir su ambiente.
- Actuadores = todo lo que le sirva para actuar.

Ejemplo: Un ser vivo.

- sensores: (ojos, oídos, nariz y olfato, gusto, tacto, ...)
- actuadores: (brazos, piernas, manos, ...)

Ejemplo: Máquina o Robot.

- sensores: (cámaras –visual, infrarrojo, etc.—, sensores, GPS, radar, Lidar,)
- actuadores: (pantalla, mensajes, luces y señales, brazos, piernas, ruedas)

Percepción

Percepto (*percept*) = contenido que los sensores perciben.

Secuencia de percepción (*percept sequence*) = historial de lo que el agente percibe.

Ejemplo: Un computador moderno.

- Archivos, paquetes de información, inputs (teclado, mouse, periféricos), impresoras, archivos de escritura, ...

Función de actuación: A cada secuencia de percepción le asigna una acción específica.

S_A = Estados del agente

S_E = Estados del ambiente

A = Conjunto de acciones

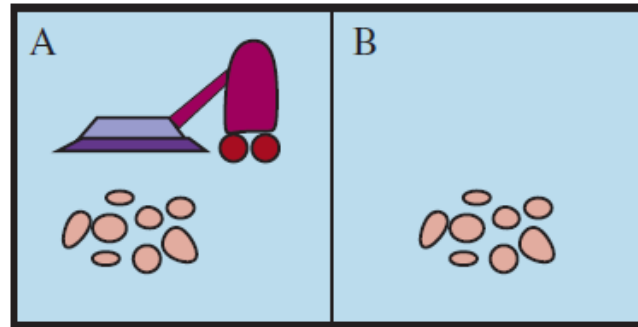
$S_A \times S_E$ = Estados del sistema

$f: IN^{S_A \times S_E} \rightarrow A$

función de actuación

Percepción

Ejemplo: Máquina aspiradora.



$S_A = \{A, B\}$ (posición del agente)

$S_E = \{0, 1\}$ (0 = limpio, 1=sucio)

$A = \{\rightarrow, \leftarrow, \text{aspirar}\}$

Si ocurre

$(A, 0)$ (= está limpio en A)

acción = \rightarrow

$f(A, 0) = \rightarrow$

$(A, 1)$ (= está sucio en A)

acción = asp

$f(A, 0) = \text{asp}$

Percepción

Percept sequence	Action
<i>[A, Clean]</i>	<i>Right</i>
<i>[A, Dirty]</i>	<i>Suck</i>
<i>[B, Clean]</i>	<i>Left</i>
<i>[B, Dirty]</i>	<i>Suck</i>
<i>[A, Clean], [A, Clean]</i>	<i>Right</i>
<i>[A, Clean], [A, Dirty]</i>	<i>Suck</i>
<i>⋮</i>	<i>⋮</i>
<i>[A, Clean], [A, Clean], [A, Clean]</i>	<i>Right</i>
<i>[A, Clean], [A, Clean], [A, Dirty]</i>	<i>Suck</i>
<i>⋮</i>	<i>⋮</i>

Si ocurren las secuencias

(A, 0), (A, 0)

acción = \rightarrow

$f(\{(A,0), (A,0)\}) = \rightarrow$

(A, 0), (B,0), (A, 1)

acción = asp

$f(\{(A,0), (B,0), (A,1)\}) = \text{asp}$

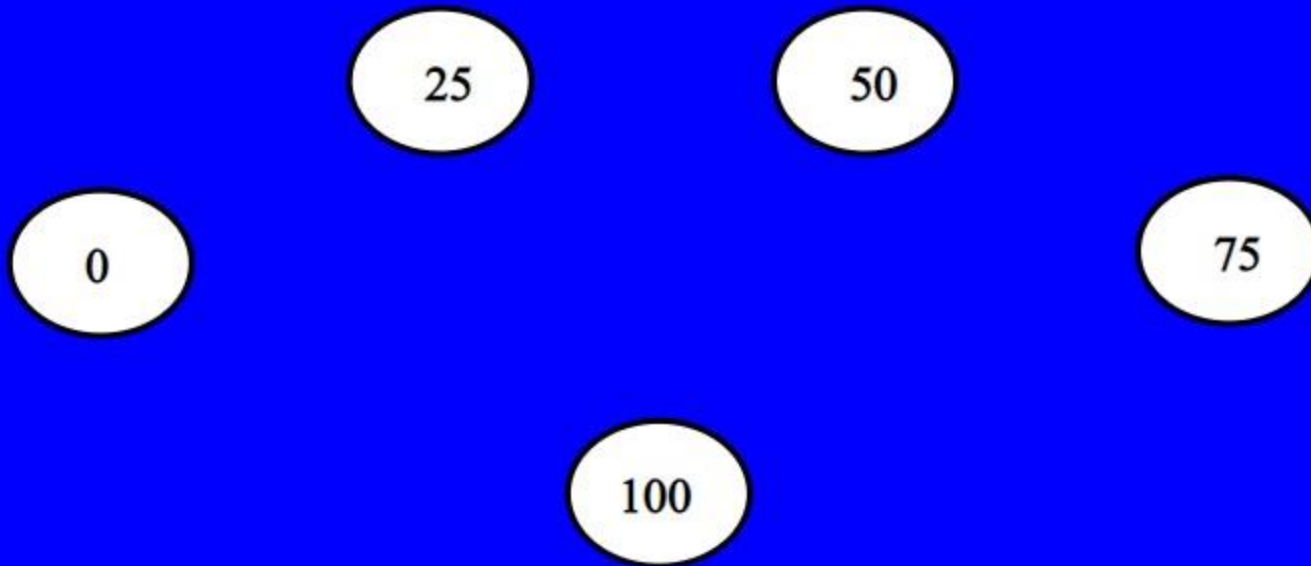
Caso particular: Autómatas

Ejemplo: Una máquina de vender sodas.

- Sólo acepta monedas de 1Q y de 25 centavos.
- No recibe más de 1Q.
- La soda cuesta 75 centavos (eso fue en los 80s)
- Posibles acciones:
 - El usuario deposita 25 centavos (25)
 - El usuario deposita 1Q (100)
 - Oprimir el botón para recibir la soda (soda)
 - Oprimir el botón para recibir el cambio (ret)

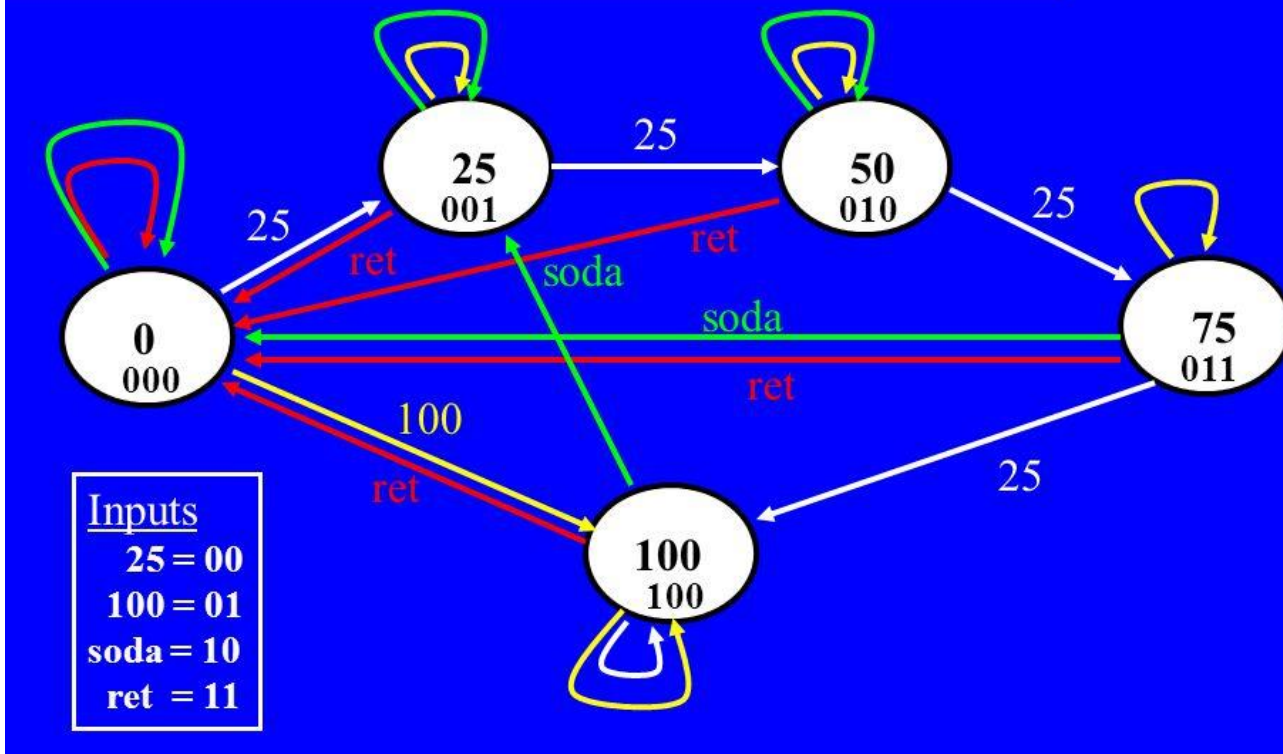
Caso particular: Autómatas

Example: Vending Machine



Caso particular: Autómatas

Example: Vending Machine



Caso particular: Autómatas

Example: Vending Machine

state			input		new state		
s2	s1	s0	i0	i1	s2	s1	s0

0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	0
0	1	0	0	0	0	1	1
0	1	1	0	0	1	0	0
1	0	0	0	0	1	0	0

0	0	0	0	1	1	0	0
0	0	1	0	1	0	0	1
0	1	0	0	1	0	1	0
0	1	1	0	1	0	1	1
1	0	0	0	1	1	0	0

state			input		new state		
s2	s1	s0	i0	i1	s2	s1	s0

0	0	0	1	0	0	0	0
0	0	1	1	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	1	0	0	0	0
1	0	0	1	0	0	0	1

0	0	0	1	1	0	0	0
0	0	1	1	1	0	0	0
0	1	0	1	1	0	0	0
0	1	1	1	1	0	0	0
1	0	0	1	1	0	0	0

Agentes racionales

Agente racional = aquel que actúa o hace “lo correcto”.

Consecuencialismo: vamos a decir qué es “lo correcto” en función de las consecuencias. Esto se mide a través de métricas.

Métrica de desempeño: función que mide o evalúa secuencias de estados ambiente.

Ejemplos:

- máquina de limpiar (estar limpio es mejor que estar sucio).
- algoritmo de regresión (reducir el error de los datos)

Racionalidad: depende de cuatro factores:

la secuencia de percepción

las acciones posibles

el conocimiento previo del agente

la métrica de desempeño

Def: Un **agente racional** es aquel que a cada secuencia de percepción le asigna una acción que maximiza la métrica de desempeño (esperada).

Entorno de tareas

Describimos un ambiente o entorno de tareas (*Task environment*) mediante

PEAS (*performance, environment, actuators, sensors*)

- *P = métricas de desempeño*
- *E = estados del ambiente*
- *A = actuadores*
- *S = sensores*

Ejercicio en grupos

Entorno de tareas

Propiedades que describen las características de cada entorno.

Completamente observable o **parcialmente observable**: si los sensores dan acceso al conjunto completo de estados SE del ambiente, en todo momento.

Uni-agente o **multi-agente**: si existe solo un agente actuando en el ambiente o hay más de uno. Caso multiagente: **competitivo** o **cooperativo**.

Determinista o **no-determinista**: Determinista = si el siguiente estado está determinado por el estado actual y la acción que se realiza.

Estocástico = si intervienen probabilidades, **no-determinista**: no hay prob.

Episódico o **secuencial**: Episódico si la experiencia se divide en episodios, pero el siguiente episodio no depende de los anteriores.

Estático o **dinámico**: Dinámico es cuando el ambiente cambia mientras el agente decide las acciones.

Continuo o **discreto**: Según el conjunto de estados del ambiente.

Conocido o **desconocido**: Según el agente conozca el estado del ambiente.

Entorno de tareas

Ejercicio en grupos

Agentes

Programa del agente: implementa la función de actuación

Arquitectura del agente: dispositivo físico o computacional

$$\textit{agente} = \textit{arquitectura} + \textit{programa}$$

Enfoque de Tabla: representar la función de actuación por una tabla.

- Este enfoque es imposible para tareas complejas!

Ejemplo: (Ajedrez), una tabla de 10^{150} estados o escenarios distintos.

$$f: (SA \times SE)^{IN} \rightarrow A \quad \text{función de actuación}$$

Ejemplo: (Taxi autónomo).

Un taxi = 8 cámaras. Una sola cámara recibe información a 70 Mb por seg.
(30 frames por seg. por 1080 x 720 píxeles, por 24 bits cada píxel). Estamos
Hablando de una tabla del orden de $10^{600,000,000,000}$.

Tipos de Agentes

Este enfoque de función-tabla sólo es útil es escenarios pequeños.

Ejemplo: (Máquina de limpiar).

```
function REFLEX-VACUUM-AGENT([location,status]) returns an action
    if status = Dirty then return Suck
    else if location = A then return Right
    else if location = B then return Left
```

Figure 2.8 The agent program for a simple reflex agent in the two-location vacuum environment. This program implements the agent function tabulated in Figure 2.3.

El objetivo de la IA es construir agentes o programas capaces de producir un comportamiento razonable, a partir de una representación simple.

Hay 4 tipos de agentes

- Agentes de simple reflejo (simple reflex)
- Agentes basados en modelos (model-based reflex)
- Agentes basados en objetivos (objective-based)
- Agentes basados en utilidad (utility-based)

Tipos de Agentes

Agentes de simple reflejo:

- Son los más sencillos
- Seleccionan acciones con base en el percepto actual (no el histórico)
- Operan en base a reglas condición-acción: *if [condition] then [action]*

Ejemplos: (Máquina de limpiar). Si [sucio] entonces[aspirar].

(Taxi autónomo). Si [carro del frente está frenando] entonces [frenar]

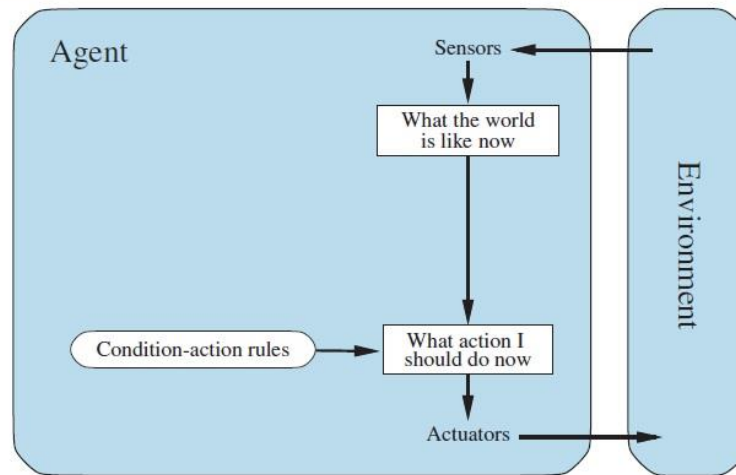


Figure 2.9 Schematic diagram of a simple reflex agent. We use rectangles to denote the current internal state of the agent's decision process, and ovals to represent the background information used in the process.

Tipos de Agentes

Agentes de reflejo basado en modelos:

- Útiles para lidiar con observabilidad parcial (estado interno)
- Requiere información de cómo cambia el ambiente (modelo de transición)
- Requiere información de cómo el ambiente se refleja en los perceptos (modelo sensorial)

Ejemplo: Si [carro del frente está frenando] entonces vamos a observar varias luces rojas.

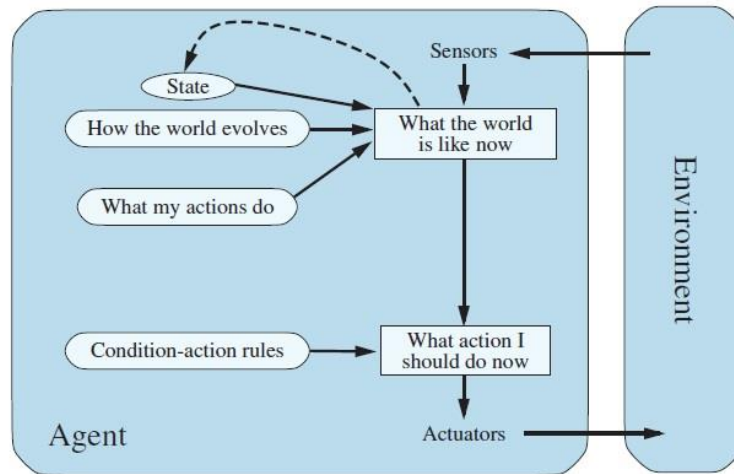


Figure 2.11 A model-based reflex agent.

Tipos de Agentes

Agentes basados en objetivos:

- En ocasiones sólo conocer el estado del ambiente no es suficiente
- Se necesita además cierta información de los objetivos
- Búsqueda y planeamiento (a futuro):
¿qué va a ocurrir si hago esto?

Ejemplo: (Taxi autónomo). Decisiones en una intersección de caminos.

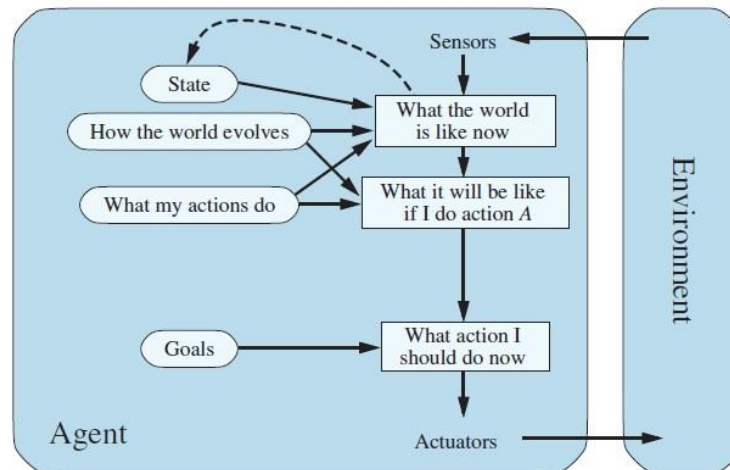


Figure 2.13 A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

Tipos de Agentes

Agentes basados en utilidad:

- Similares a los basados en objetivos
- Incorporan una “función de utilidad” o de “utilidad esperada” (es una internalización de las métricas de desempeño)

Ejemplos: (Máquina de limpiar). Se espera que el ambiente quede limpio.
(Taxi autónomo). Llegar al objetivo, minimizando distancia o tiempo.

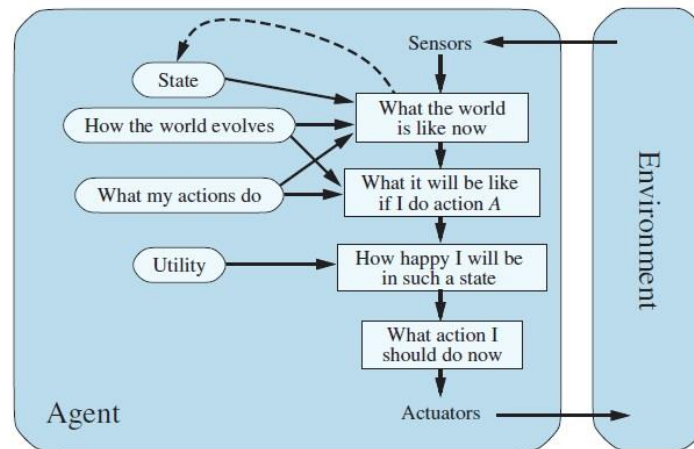


Figure 2.14 A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.

Tipos de Agentes

Aprendizaje en agentes:

- Un **elemento de aprendizaje** (learning element) es responsable de inducir mejoras en el agente.
- Durante su diseño, se incorporan **generador de problemas** (problem generator), responsable de sugerir acciones que conduzcan a nuevas experiencias informativas, que generen mejoras.

Ejemplo: Creador automático de billetes falsos vs. detector de billetes falsos.

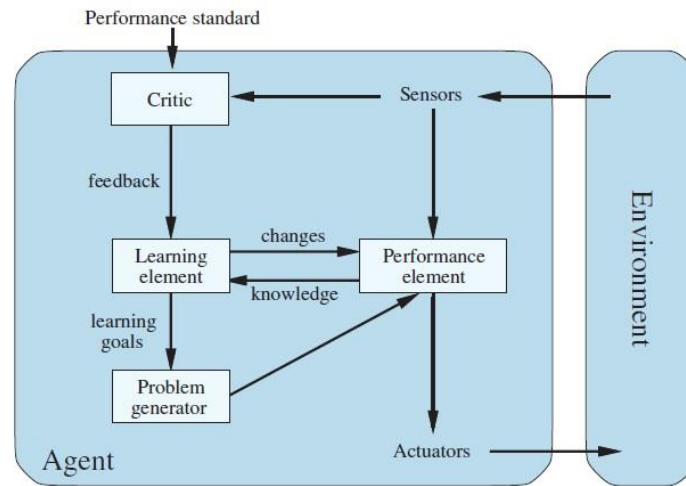


Figure 2.15 A general learning agent. The “performance element” box represents what we have previously considered to be the whole agent program. Now, the “learning element” box gets to modify that program to improve its performance.