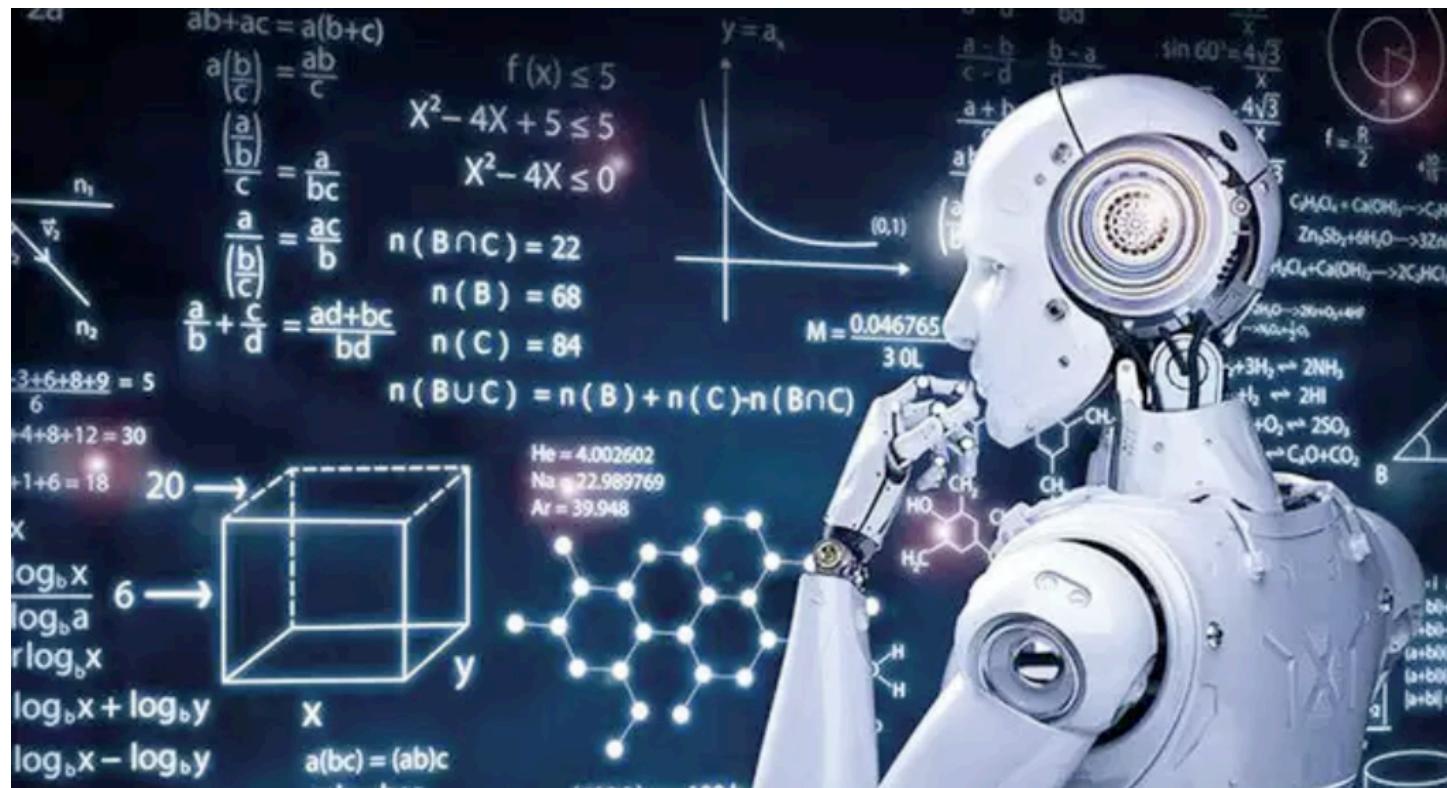


Inteligencia Artificial

Agentes Inteligentes



Agentes y Entornos

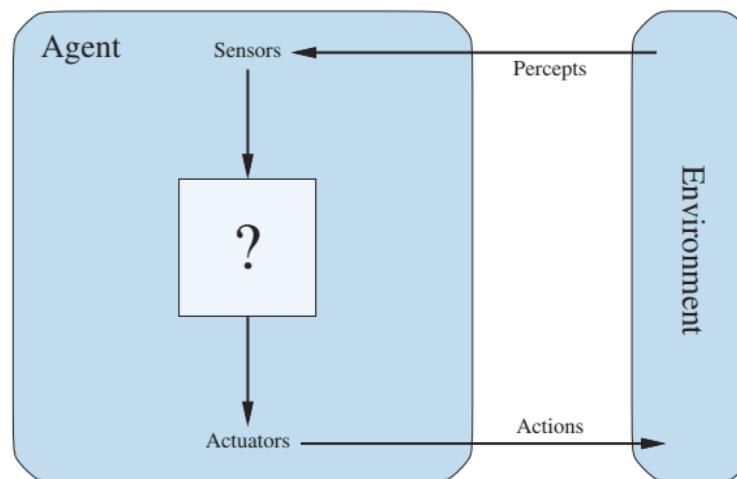
- Lectura: Capítulo 2 : *Intelligent Agents, “IA a modern approach.”*

- El concepto Agente Racional es central en AI como hemos visto en las sesiones anteriores.
- Puede aplicarse a multitud de entornos, cuya dificultad pondrá en jaque el éxito del agente
- Desarrollaremos un pequeño conjunto de principios de diseño que nos permitirán construir agentes exitosos, a los que podemos considerar **inteligentes**.

Un agente es una entidad que percibe su entorno a través de sus sensores, y actúa sobre él a través de sus actuadores

Un agente inteligente es un algoritmo que puede automáticamente llevar a cabo numerosas tareas basando sus decisiones en el entorno y sus experiencias.

- La noción de agente pretende ser una **herramienta de análisis de sistemas**, no una caracterización absoluta que divide al mundo en agentes y no agentes.

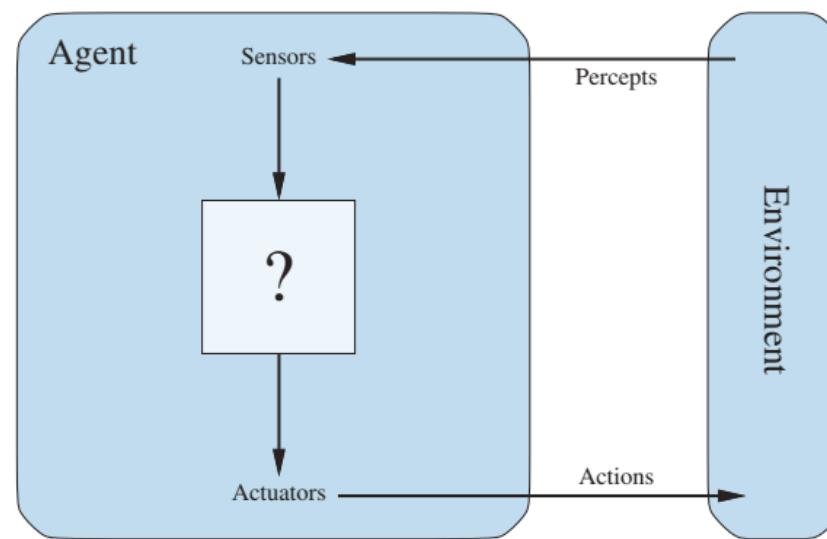


El término **percepción** se refiere a las entradas recibidas en un instante dado.

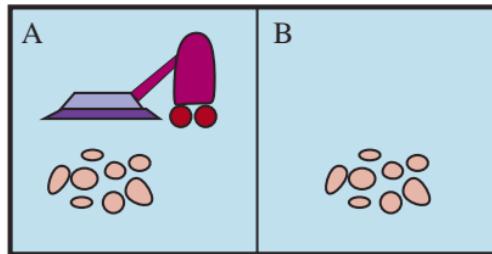
- Una **secuencia de percepción** de un agente es la **historia completa** de todo lo que el agente ha percibido alguna vez.

La elección de acción de un agente en un instante dado puede depender de toda la secuencia de percepción observada hasta la fecha, pero no de nada que no haya percibido.

- Definimos totalmente un **agente** al especificar la elección de su acción para cada posible secuencia de percepción
- Matemáticamente hablando, se dice que el comportamiento de un agente es descrita por la **función de agente** que mapea cualquier secuencia de percepciones dado a una acción. Esto es una **caracterización externa**. La función agente podría ser una tabla... como veremos en la siguiente transparencia.
- La función de agente se implementa en el agente por medio de un **programa de agente**



Mundo de la aspiradora

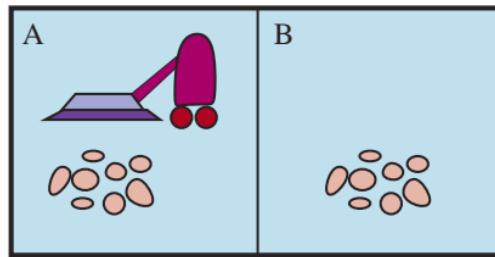


Función de agente

Percept sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
[A, Clean], [A, Clean]	Right
[A, Clean], [A, Dirty]	Suck
:	:
[A, Clean], [A, Clean], [A, Clean]	Right
[A, Clean], [A, Clean], [A, Dirty]	Suck
:	:

Depende del
objetivo

Relación entre ambas
dos



Programa de agente

implementación de función agente.

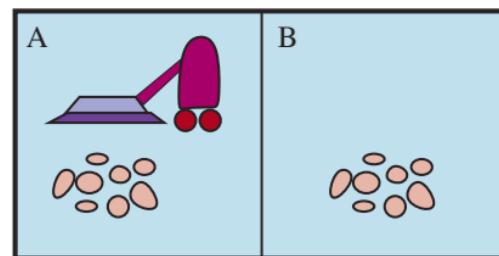
```
function REFLEX-VACUUM-AGENT([location,status]) returns an action
    if status = Dirty then return Suck
    else if location = A then return Right
    else if location = B then return Left
```

Buen comportamiento: El concepto de racionalidad

Un **agente racional** es el que hace las cosas bien:

- Selecciona la acción que produce más éxito
- ¿Cómo medimos el éxito? en base a **consecuencias de sus acciones**
- Una secuencia de acciones hace que el entorno evoluciones a distintos estados
- Existen varias secuencias deseables posibles, y **medir su desempeño** (Performance measure) puede ser subjetivo y/o complicado. Además se puede generar comportamientos no deseados como el que una aspiradora recoja suciedad y vuelva a volcarla para **maximizar la cantidad de basura recogida**.
- Es mejor que tenga objetivos **relativos a como se debe comportar**. En este caso una de las opciones para tener un comportamiento más racional sería que el desempeño fuese medido por **tener el suelo limpio**.
- Cuidado con los objetivos que especificamos, Hay que recordar a Norbert Wiener: “**El objetivo que le ponemos a la máquina es el que realmente deseamos**”

Como regla general, **es mejor diseñar medidas de rendimiento de acuerdo con lo que uno realmente quiere en el entorno, que de acuerdo con cómo cree que debería comportarse el agente.**

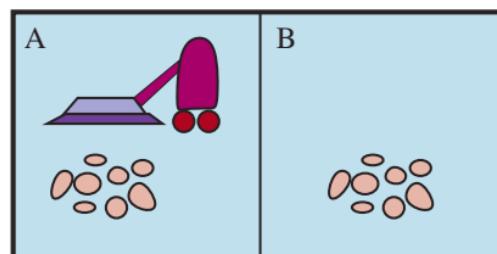


- Zonas limpias por cada unidad de tiempo
- Electricidad usada
- Trabajo mediocre continuo vs eficiente con paradas

Ser racional depende de:

- La medida de desempeño que define el criterio de éxito.
 - El conocimiento previo del agente sobre el medio ambiente.
 - Las acciones que puede realizar el agente
 - La secuencia de percepción del agente hasta la fecha
- Definición de agente racional:

Para cada posible secuencia de percepción, un agente racional debe seleccionar una acción que se espera que maximice su medida de desempeño, dada la evidencia proporcionada por la secuencia de percepción y cualquier conocimiento incorporado que tenga el agente.



Si está sucio, limpia. Si no, se mueve a la otra posición

¿Sería racional en otras circunstancias?

- Cuadrados limpios todo el tiempo
- Entorno desconocido
- ...

Tal y como está definida la función agente el comportamiento sería racional si asumimos que:

- Zonas limpios por cada unidad de tiempo: 1 punto
- Entorno conocido (pero no la posición inicial ni la suciedad)
- 3 acciones: izquierda, derecha, limpia
- Percepciones correctas

↓
Te crees drios

No confundir ser racional con omnisciente:

- ! • La racionalidad maximiza el rendimiento esperado, mientras que la perfección maximiza el rendimiento real. El agente conoce a priori el resultado de su acción.
- ! • No necesitamos omnisciencia, la definición de racionalidad contempla decisiones basadas en percepciones recibidas hasta momento
- Racionalidad necesita de **Recopilación de información** - realizar acciones para modificar percepciones futuras.
 - Exploración
 - Aprendizaje de lo que se percibe.
 - Con el **conocimiento previo solo, el agente carece de autonomía** (ni aprende ni percibe)
 - El aprendizaje permite a un agente racional independizarse del conocimiento previo



La naturaleza de los entornos

Entorno de la tarea = Medida de desempeño + entorno + sensores + actuadores

Los **entornos de la tarea (task environments)** son los problemas para los cuales un agente inteligente es la solución

La descripción **PEAS** describe de manera completa un entorno de tarea y agrupa todos los criterios que necesitamos para tener un agente que actúe de forma racional.

Performance Environment Actuators Sensors**PEAS para el caso de conductor de taxi**

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Propiedades de los entornos de tarea

Completamente observable vs parcialmente observable.

- Un entorno es **completamente observable** si los sensores del agente te dan toda la información relevante para tomar decisiones. No necesita estado interno.

• Un entorno es parcialmente observable si los sensores del agente no te dan el estado completo o preciso del entorno.

• Entornos inobservables

Agente único vs multi agente.

• Ejemplos:

- Agente único -> Resolviendo puzzles.
- Multi agente -> Ajedrez

• Multi agente:

- Competitivo (Ajedrez - minimizar rendimiento del otro jugador) y cooperativo (Taxi - Evitar colisiones)
- Comunicación (emerge como comportamiento racional multi agente)
- Comportamiento aleatorio (para evitar ser predecibles)

Determinista vs No determinista/estocástico (se conocen las probabilidades).

- Determinista: Siguiente estado = Estado actual + Acción (ajedrez)
- Entornos complejos y no observables pueden ser tratados como no deterministas (taxi)

Un entorno es incierto si:

- no es completamente observable, o
- no es determinista

Un entorno es no determinista si

- Una acción se caracteriza por su posible resultado
- No probabilidades



Propiedades de los entornos de tarea

Episódicos vs secuencial.

- Es episódico si el próximo episodio no depende de otro
- Es secuencial si las decisiones actuales afectan a las decisiones futuras
- Ejemplos:
 - Ajedrez, Taxi -> Secuencial
 - Robot que detecta y retira partes defectuosas -> episódico

• Estático vs dinámico.

- Es estático si el entorno no cambia mientras el agente delibera
- Es dinámico si se requiere actuaciones constantes (puede ser no hacer nada)
- Semidinámicos, cuando son estáticos, pero su medida de rendimiento varía

• Discreto vs continuo.

- se aplica al estado del entorno, a la forma en que se maneja el tiempo y a las percepciones y acciones del agente

• Conocido vs Desconocido.

- Se refiere al conocimiento del agente de los efectos de sus acciones (Física del entorno)
- En un entorno conocido, se dan los resultados (o probabilidades de resultado si el entorno es estocástico) para todas las acciones.



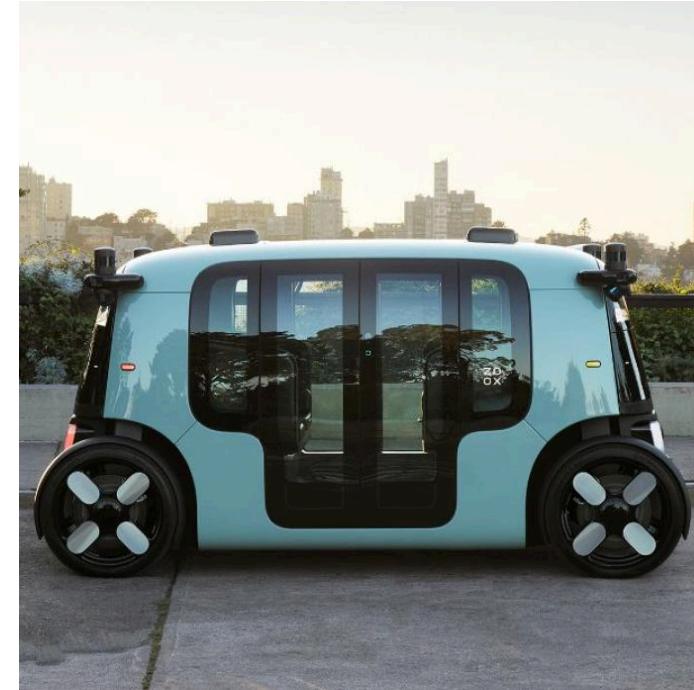
Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

La estructura de los agentes

El **objetivo** de la AI es **diseñar un programa agente** que implemente la función de agente — mapeado de percepciones a acciones

- Este programa se ejecuta en un **dispositivo de computación** con sensores y actuadores adecuados (**arquitectura**)

agente = arquitectura + programa



- Todos los agentes tienen el mismo patrón : Toman una percepción y generan una acción
- Un función de agente, en cambio, usa todas las percepciones anteriores
 - Un programa agente debería guardar el histórico si lo necesita para decidir

```
function TABLE-DRIVEN-AGENT(percept) returns an action
  persistent: percepts, a sequence, initially empty
              table, a table of actions, indexed by percept sequences, initially fully specified
  append percept to the end of percepts
  action  $\leftarrow$  LOOKUP(percepts, table)
  return action
```

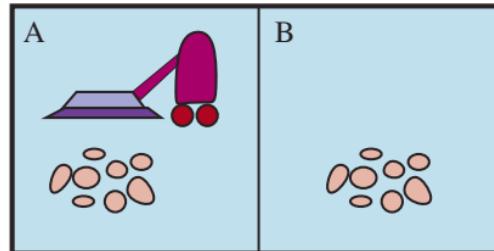
Tablas de raíces cuadradas vs método de Newton

- ¿Puede la IA hacer lo mismo para comportamientos inteligentes generales?

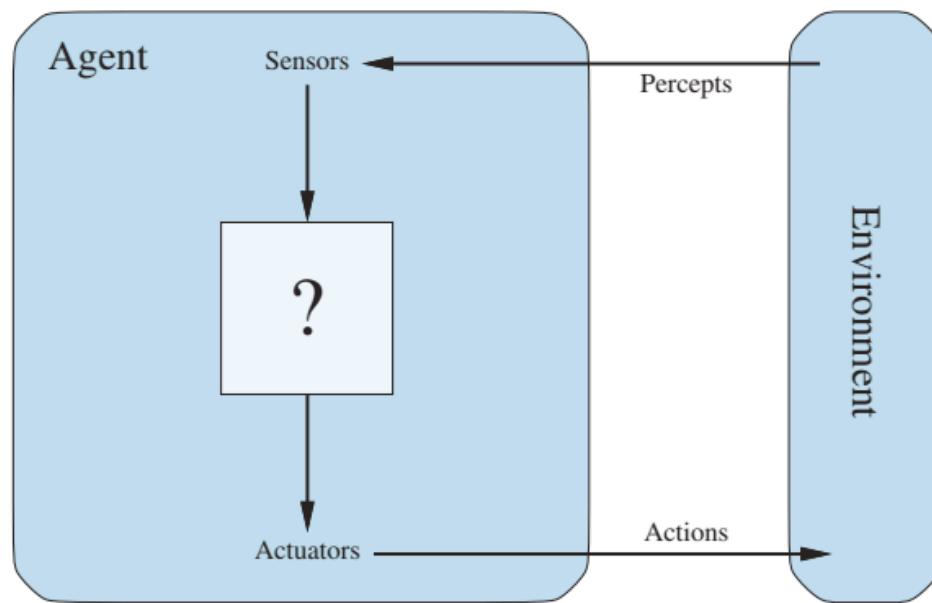
El desafío clave para la IA es descubrir cómo escribir programas que, en la medida de lo posible, produzcan un comportamiento racional a partir de un programa pequeño en lugar de una gran tabla.

- Veremos:
- Agentes reactivos simples;
- Agentes reactivos basados en modelos;
- Agentes basados en objetivos
- Agentes basados en función de Utilidad.

```
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
```



```
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
```



```
function SIMPLE-REFLEX-AGENT(percept) returns an action
  persistent: rules, a set of condition-action rules

  state  $\leftarrow$  INTERPRET-INPUT(percept)
  rule  $\leftarrow$  RULE-MATCH(state, rules)
  action  $\leftarrow$  rule.ACTION
  return action
```

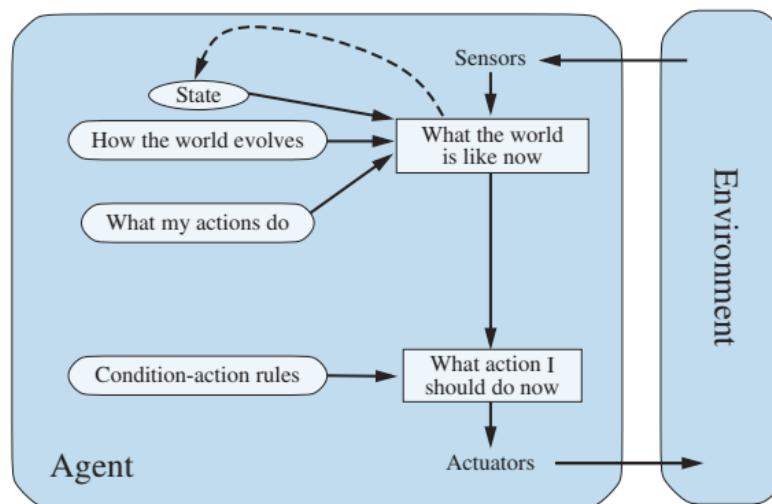
- Son simples, de limitada inteligencia
- Si el entorno no es completamente observable, estamos en problemas
 - Bucles infinitos -> aleatorización

Un agente puede tomar decisiones correctas en un entorno parcialmente observable si almacena el estado de la parte del mundo que no se puede percibir al tomar la decisión.

1. Cómo es mundo evoluciona independientemente del agente
2. Como las acciones del agente modifican el mundo

A esto llamamos **Modelo**,

y los agentes que lo usando son **agentes basados en modelo**



Un agente puede tomar decisiones correctas en un entorno parcialmente observable si almacena el estado de la parte del mundo que no se puede percibir al tomar la decisión.

1. Cómo es mundo evoluciona independientemente del agente
2. Como las acciones del agente modifican el mundo

A esto llamamos **Modelo**,

y los agentes que lo usando son **agentes basados en modelo**

```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
    persistent: state, the agent's current conception of the world state
                model, a description of how the next state depends on current state and action
                rules, a set of condition-action rules
                action, the most recent action, initially none

    state  $\leftarrow$  UPDATE-STATE(state, action, percept, model)
    rule  $\leftarrow$  RULE-MATCH(state, rules)
    action  $\leftarrow$  rule.ACTION
    return action
```

Tener un objetivo puede no ser suficiente para generar comportamientos de alta calidad en la mayor parte de los entornos

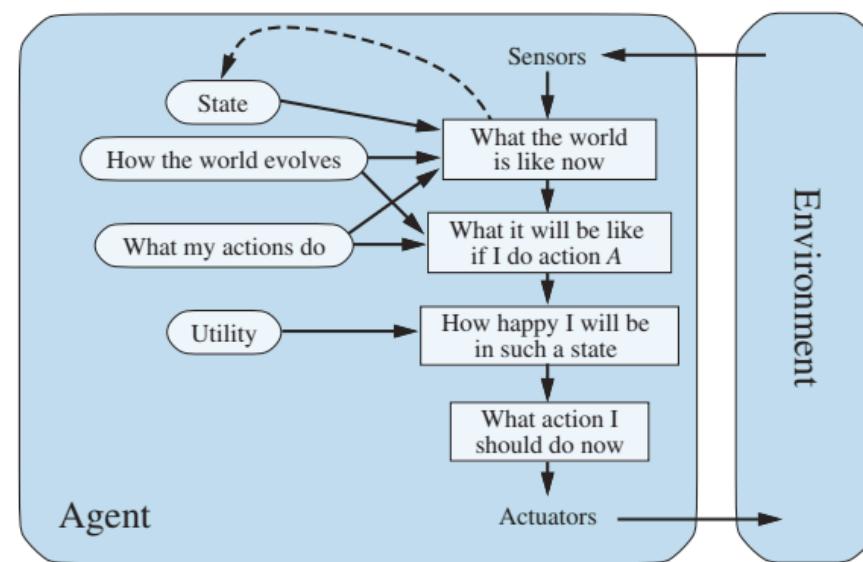
- La Utilidad es una medida de rendimiento que permite comparar diferentes conjuntos de estados
- Función de utilidad

```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
    persistent: state, the agent's current conception of the world state
                model, a description of how the next state depends on current state and action
                rules, a set of condition-action rules
                action, the most recent action, initially none

    state  $\leftarrow$  UPDATE-STATE(state, action, percept, model)
    rule  $\leftarrow$  RULE-MATCH(state, rules)
    action  $\leftarrow$  rule.ACTION
    return action
```

Tener un objetivo puede no ser suficiente para generar comportamientos de alta calidad en la mayor parte de los entornos

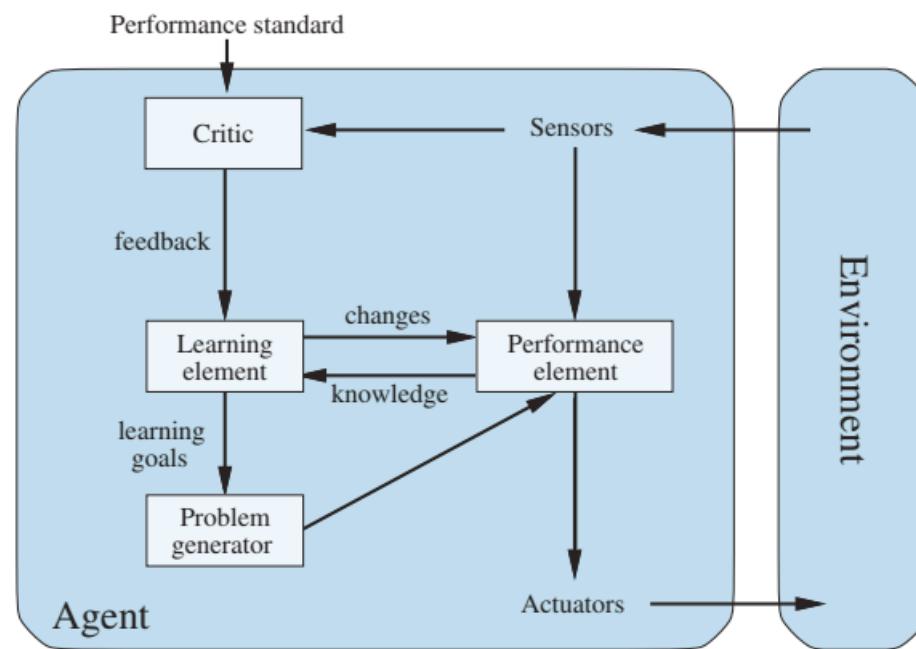
- La Utilidad es una medida de rendimiento que permite comparar diferentes conjuntos de estados
- Función de utilidad



Tienen la ventaja de poder desarrollarse en entornos desconocidos

- Elementos:

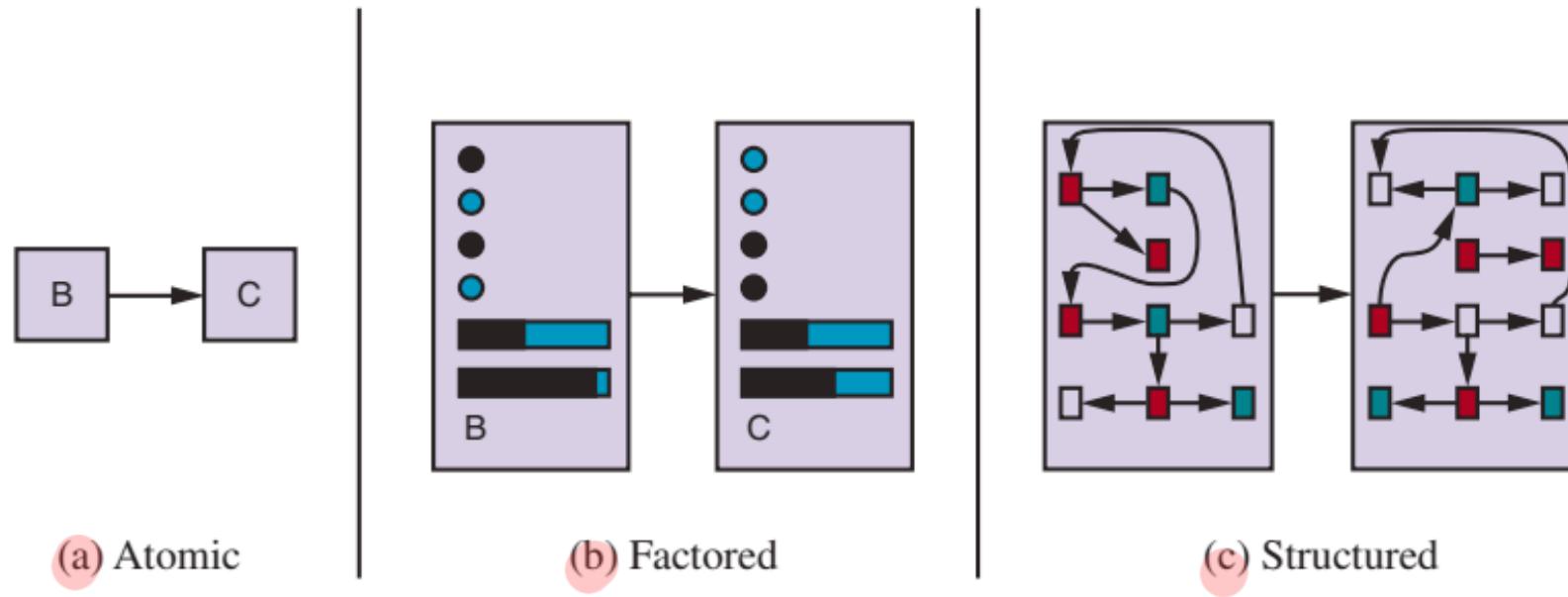
- Learning element: Es el responsable de hacer mejoras en el sistema
- Performance element: Selecciona la acción. Sería un agente como los anteriores.
- Critic: Determina cómo el agente está funcionando y como el performance element podría mejorar
- Problem generator: sugiere acciones que llevan a nuevas e informativas experiencias



¿Cómo representa el agente el estado del entorno?

- Representación de los estados

- a) • atómica: Los estados son indivisibles y no tienen estructura interna (Search, HMM, MDP)
- b) • factorizado: Cada estado tiene variables o atributos (Planning, Lógica proposicional, Machine Learning, BN)
- c) • Estructurado: Cada estado describe objetos y las relaciones entre ellos (FOL, lenguaje natural)



Ejercicios

1. Para cada una de las siguientes afirmaciones, diga si es verdadera o falsa y respalde su respuesta con ejemplos o contraejemplos cuando corresponda.

Un agente que sólo detecta información parcial sobre el estado no puede ser perfectamente racional.

Existen entornos de tareas en los que ningún agente reactivo puro puede comportarse razonadamente.

Existe un entorno de tareas en el que cada agente es racional.

La entrada a un programa de agente es la misma que la entrada a la función de agente.

Cada función de agente se puede implementar mediante alguna combinación de programa / máquina.

Suponga que un agente selecciona su acción de manera uniforme al azar del conjunto de acciones posibles.

Existe un entorno de tareas determinista en el que este agente es racional.

Es posible que un agente dado sea perfectamente racional en dos entornos de tareas distintos. Todo agente es racional en un entorno no observable.

Un agente de póquer perfectamente racional nunca pierde.

2. Para cada una de las siguientes actividades, proporcione una descripción PEAS del entorno de la tarea y caratterícela en términos de las propiedades enumeradas en la trasparencia 13:

Jugando futbol.

Explorando los océanos subsuperficiales de Titán.

Comprar libros usados de IA en Internet.

Jugar un partido de tenis.

Practicando tenis contra una pared.

Realización de un salto de altura.

Tejer un suéter.

Pujar por un artículo en una subasta.

3. Escriba programas de agente de pseudocódigo para los agentes basados en objetivos y basados en utilidades.