

Imaginemos un entorno determinista, estático y observable (puedo ver mi estado completo), discreto

Determinista	Estático	Observable
$s = f(a)$ estado	acción	Deja pasar esa pista... $s = (s_1 \dots s_n) \in D_1 \times \dots \times D_n = S$

$s$  pertenece a un conjunto finito de estados.  
 $a$  pertenece a un conjunto de acciones  
 $a \in A$ , pero muchas veces  $a \in f(s)$ , acciones legales en  $s$  (un estado legal).

Tom

$$\begin{array}{ccc} \text{II} & \text{II} & \text{II} \\ A & B & C \end{array} \quad a = \{AB, AC, BA, BC, CA, CB\}$$

(No necesitáis poner conjuntos más pequeños abajo de los demás)

En este caso al ser un entorno estático las acciones pertenecen al solo conjunto.

Det, Dinámico, Observable, Discreto  
determinista

$s_{k+1} = f(s_k, a_k)$  se calcula el siguiente estado a partir del anterior.  
 Este es como misioneros y canibales.  
 O n-puzzle. O torres de Hanoi.  
 Si se conoce la función es ...

14 de enero 2026

Det., dinámico, Parcialmente observable, el control

$$S_{k+1} = f(S_k, a_k) \quad \text{Dominios.}$$

$$P_k = g(S_k)$$

$$g(S_k) = S_{k+1} \quad (\text{observable})$$

Si  $g(S_k)$  es subconjunto es P. observable.

Juego donde la gente no puede ver todo el estado. (P. observable)

Det. dinámico, Parcial. Obs., continuo.

Asumimos que la var que cambia es el tiempo.

$$x_t = f(x_t, a_t) \quad > \text{ECS. diferenciales.}$$

$$P_t = g(x_t) \quad \text{en modelo.}$$

> Teoría del control. > Si fuera espacio E de probabilidad

Típicamente si son continuas se modelan con ecuaciones diferenciales. >  $P_t = g(x_t)$

observable

Estocástico

Det. Estático, Obs., Discreto.

$$S = \{S^{(1)}, \dots, S^{(m)}\} \quad m = \text{card}(S)$$

- No sabemos que pasará al hacer una acción.

$$\text{Prob. } [S/a] = \left\{ \begin{array}{l} \Pr[S = S^{(1)}] a \\ \Pr[S = S^{(2)}] a \\ \vdots \\ \Pr[S = S^{(m)}] a \end{array} \right\}_{S=1}^{m+1}$$

- Por miedo: No se exactamente que pasará tras cierta acción. Modela su probabilidad.
- Por flojo: El modelo es tan complejo que se opta por prob.

Estocástico, Dinámico, Obs., Discreto

Juegos como Risk. Tiras el dado, no sabes que va a pasar. Sabes donde estabas, pero no adónde vas.

$$\Pr[S_{k+1} | S_k, a_k] = \left[ \begin{array}{l} \Pr[S_{k+1} = s^{(1)} | S_k, a_k] \\ \Pr[S_{k+1} = s^{(2)} | S_k, a_k] \\ \vdots \\ \Pr[S_{k+1} = s^{(m)} | S_k, a_k] \end{array} \right] \quad \{ \delta = 1 \}$$

Los entornos también pueden ser episódicos, problema-resolución. Cuando un problema es episódico puede ocurrir aprendizaje.

Modelo conocido: - Poker, se conocen las reglas.

Modelo desconocido: - Manejar un carro autónomo.

Para estos modelos se requieren datos extensos y obtenidos de experiencia. Aprendizaje.

Primero se tiene que saber el PEAS.

P es una función, usualmente de recompensa o de ganancia / pérdida. Minimizar pérdidas es más importante.

$$-\min(-a, -b) = \max(a, b)$$

$$\arg \max_x f(x) = \arg \min_x -f(x)$$

$$x_{k+1} = f(x_k, a_k)$$

$$p_k = g(x_k)$$

$$a_{k+1} \in A(s_{k+1})$$

$$a_k \in A(s_k)$$

$$p_0, p_1, \dots, p_k$$

procesar  $u_1, \dots, u_k \rightarrow f_A \xrightarrow{a_k}$  proceso

$f_A$  - función de agente, da  $a_k$

$a_k$  - acción de agente

$p_k$  - percepciones

$$f_A: P^* \rightarrow A$$

Menos  
abstracto

Subímbolico

LLM

Más  
abstracto

Simbólico

Todo se convierte  
a símbolos, no simbol.

# Intelligent Agents Chapter II

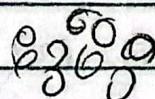
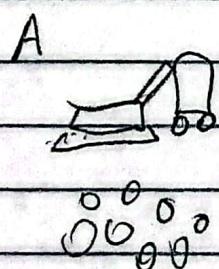
## AI as agents

15 de enero 2026

$\text{act} = \text{AgentFn}(\text{percept})$   
sensors, agent fn, actuators

Hacer inteligencia artificial es resolver la función.

Vacuum-cleaner world



Actions: left, right,  
suck, No op

$$\begin{aligned} S &= (A, B / R) \\ A &\in \{L, S\}^2 \\ B &\in \{L, S\}^2 \\ R &\in \{A, B\}^2 \\ &8 \text{ estados} \end{aligned}$$

AB · Cuarto +  
R · Robot

$$A = \{\leftarrow, \rightarrow, \text{aspira}, \text{nada}\}$$

s - situación

$$\text{Percepciones } p = (R, S[R])$$

$$o = (R, S[\text{val}(R)])$$

valor de R y la situación en valor de R (A o B)  
(donde está)

IS enero 2006

(Sucio, Sucio, 'A')

function REPLEX-LAUM-AGENT,

(location, status) returns an action.

if status = Dirty then return suck.

else if location

El problema con un agente reflex es que no tiene memoria

$$a_t = f(p_t)$$

$$a_t = f(p_t, x_t)$$

$$\text{Init } x_t = (s, s, a)$$

def  $f(p_t, x_t)$  Donde esta mi robot?

#actualizar  $x_t$

$$> p_t = (R, s[\text{val}[n]])$$

$$x_t[2] = p_t[0]$$

$$\text{if } p_t[0] = 'A'$$

$$x_t[0] = p_t[0]$$

else

$$x_t[1] = p_t[1]$$

El estado interno no es el estado del entorno.

if  $x_t[0] == x_t[1] == 'L'$

a = 'nada'

elif  $p_t[1] == 'S'$

a = 'aspirar'

15 Enero 2020

clif  $p[0] == 'A'$

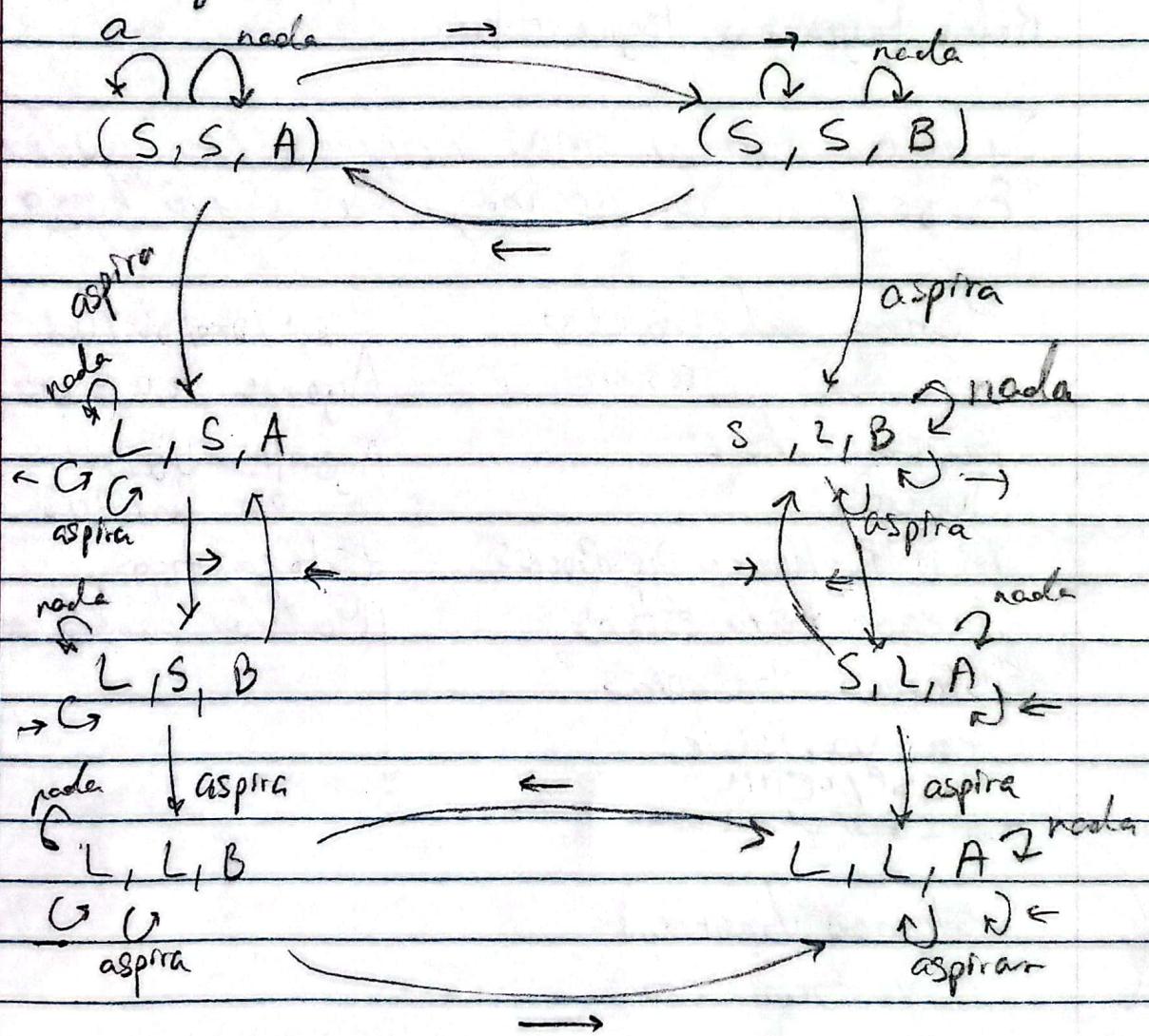
$a = ' \rightarrow '$

else

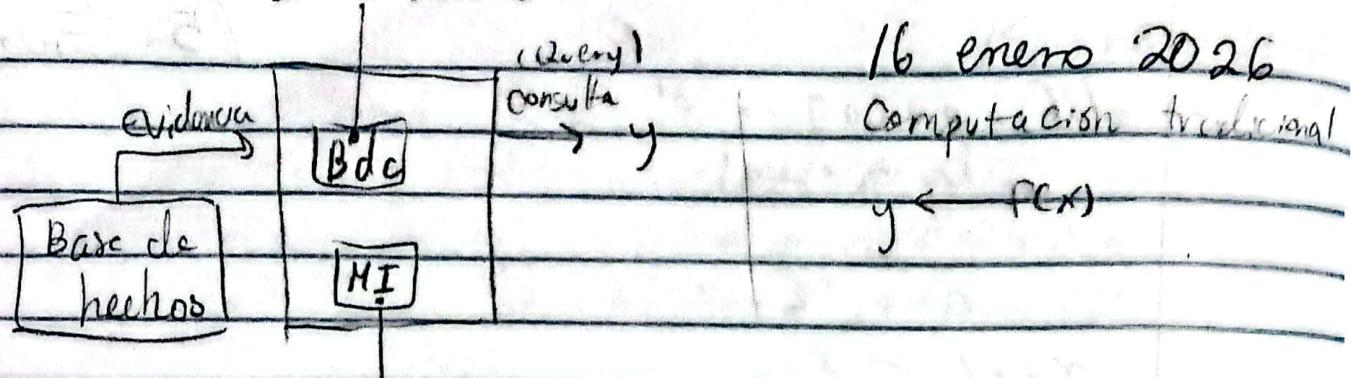
$a = ' \leftarrow '$

$x_t+1 \leftarrow x_t$

regresa a.



Base de conocimiento



16 enero 2026

Computación tradicional

$$y \leftarrow f(x)$$

Motor de Inferencia

Redes bayesianas, lógica difusa, etc.

Su uso para sistemas reactivos con objetos  
"Estos son los hechos, dime, ¿qué hago?"

Menos abstracto

est/dec/obs/act  
descendido  
Agente reactivo

Búsquedas

Agentes basados en disc  
en metas obs/act  
descendido

Juegos

MDP: Markov Dec. Process

estocásticos

Juegos estocásticos

Modelos basados en probabil

Sistemas basados

en variables

CSP/OPTIM

restricciones/optimización

Sistemas basados  
en conocimiento

Más abstracto

\* constraint satisfaction problem