

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

Determinista

Estático

Observable

$$S = f(a)$$

estado                      acción

Deja pasar esa pieza...

$$S = (s_1 \dots s_n) \in D_1 \times \dots \times D_n = S$$

$S$  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 dado).

Tan

$\Pi$        $\Pi$        $\Pi$   
 A      B      C

$a = \{AB, AC, BA, BC, CA, CB\}$   
 (No puedes poner anillos más  
 pequeños abajo de grandes)

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

Determinista, Dinámico, Observable, Discreto

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



14 de enero 2026

Det., dinámico, Parcialmente observable, discreto

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

$$P_k = g(S_k)$$

$$g(S_k) = s_k \quad (\text{Observable})$$

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 ECS. discreto.

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

Observable

Estocástico

Det. Estático, Obs., Discreto.

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

- No sabemos que pasará a hacer una acción.

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

- Por menos: No se exactamente que pasará a hacer una acción. Modelo su probabilidad.

- Por flojo: El modelo es tan complejo que se opte por prob.



Estocástico, Dinámico, Obs., Discreto

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

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

Dos 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 obtenidos de experiencia. Aprendizaje.

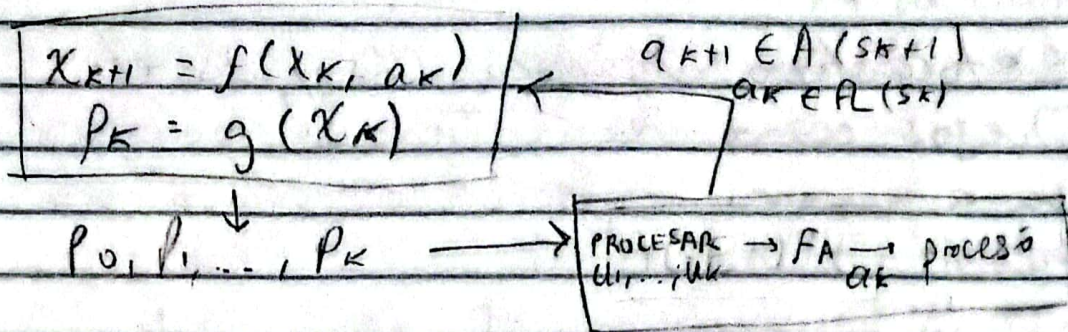
Primero se tiene que saber el PEAS.

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

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

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





$f_A$ - función de agente, de $a_k$ $a_k$ - acción de agente $P_k$ - Percepciones	$f_A : P^k \rightarrow A$
---	---------------------------

Menos abstracto

↑  
subsimbolico  
LLM

Más abstracto

↑  
simbolico  
Todo se convierte a símbolos, números.



# Intelligent Agents Chapter II

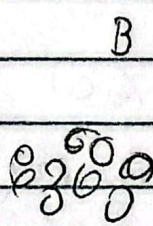
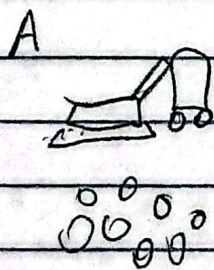
## AI as agents

15 de enero 2026

act = AgentFn(percept)  
sensors, agent fn, actuators

Hacer inteligencia artificial es resolver la función,

Vacuum-cleaner world



Actions: left, right,  
suck, No op

$S = (A, B, R)$

$A \in \{L, S\}$  2

$B \in \{L, S\}$  2

$R \in \{A, B\}$  2

8 estados

AB - Cuarto

R - Robot

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

s - situación

Percepciones  $p = (R, S[R])$

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

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



15 enero 2026

(Sucio, Sucio, A)

function REPLEX-VACUUM-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.

$$a_t = f(p_t)$$

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

$$\text{init } x_t = \overset{(0)}{S}, \overset{(1)}{S}, \overset{(2)}{A}$$

def  $f(p_t, x_t)$

Donde está mi robot?

# actualizar  $x_t$

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

$x_t[2] = p_t[0]$

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

$a_t = 'nada'$

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

$a = 'aspirar'$



15 Enero 2020

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

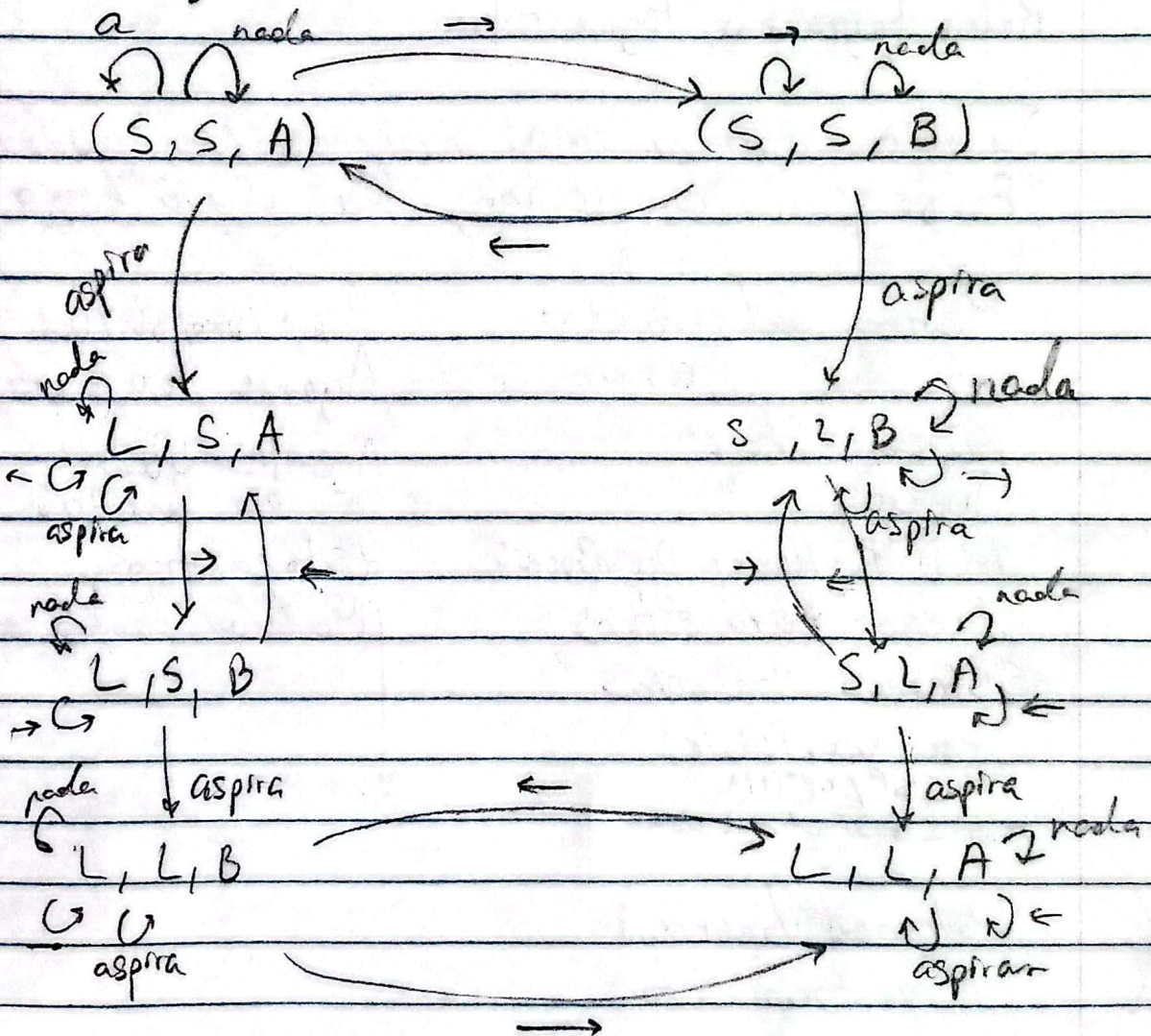
$a = ' \rightarrow '$

else

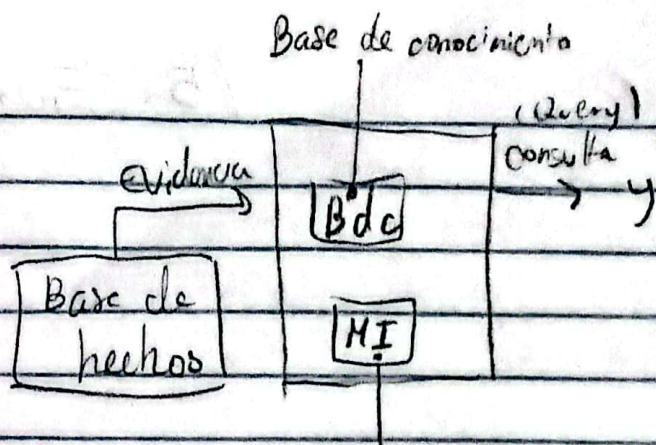
$a = ' \leftarrow '$

$x_{t+1} \leftarrow x_t$

regresa a.







16 enero 2026  
Computación tradicional  
 $y \leftarrow f(x)$

Motor de Inferencia  
Redes bayesianas, lógica difusa, etc.

Se usa para sistemas reactivos con estados  
"Estos son los hechos, dame, ¿qué hago?"

Menos abstracto

Búsquedas  
Juegos  
MDP: Markov Dec. Process  
Juegos estocásticos  
Sistemas basados  
en variables  
CSP/OPTIM  
restricciones/optimización

Est./Disc./Obs./Det desconocido  
Agente reactivo

Agentes basados en metas  
Disc./Disc./Obs./Det conocido

Estocásticos,  
Modelos basados en utilidad

Sistemas basados  
en conocimiento

Más abstracto  
constraint satisfaction problem