

# Aprendizaje por refuerzo

Procesos de decisión de Markov

---

Antonio Manjavacas Lucas

[manjavacas@ugr.es](mailto:manjavacas@ugr.es)

# Índice

1. Búsqueda asociativa
2. Procesos de decisión de Markov
3. Problemas episódicos y continuados
4. Objetivos y recompensas
5. Políticas
6. Funciones de valor y optimalidad
7. Trabajo propuesto

Hasta ahora, nos hemos centrado en **problemas no asociativos**:

- Problemas en los que *no asociamos* diferentes acciones a diferentes situaciones.
- El objetivo es encontrar la mejor acción si el problema es **estacionario**, o buscarla directamente si es **no-estacionario**.

No obstante, en problemas de RL más complejos, pueden darse diferentes situaciones donde **el valor de una acción varía dependiendo del estado actual del agente**.

Es lo que denominamos un problema de **búsqueda asociativa**.

# Búsqueda asociativa

---

# Búsqueda asociativa

## Problema de búsqueda asociativa

- **Búsqueda** basada en prueba y error para encontrar las mejores acciones.
- **Asociativa**, porque trata de asociar a cada situación (estado) la mejor acción disponible.

# Búsqueda asociativa

## Problema de búsqueda asociativa

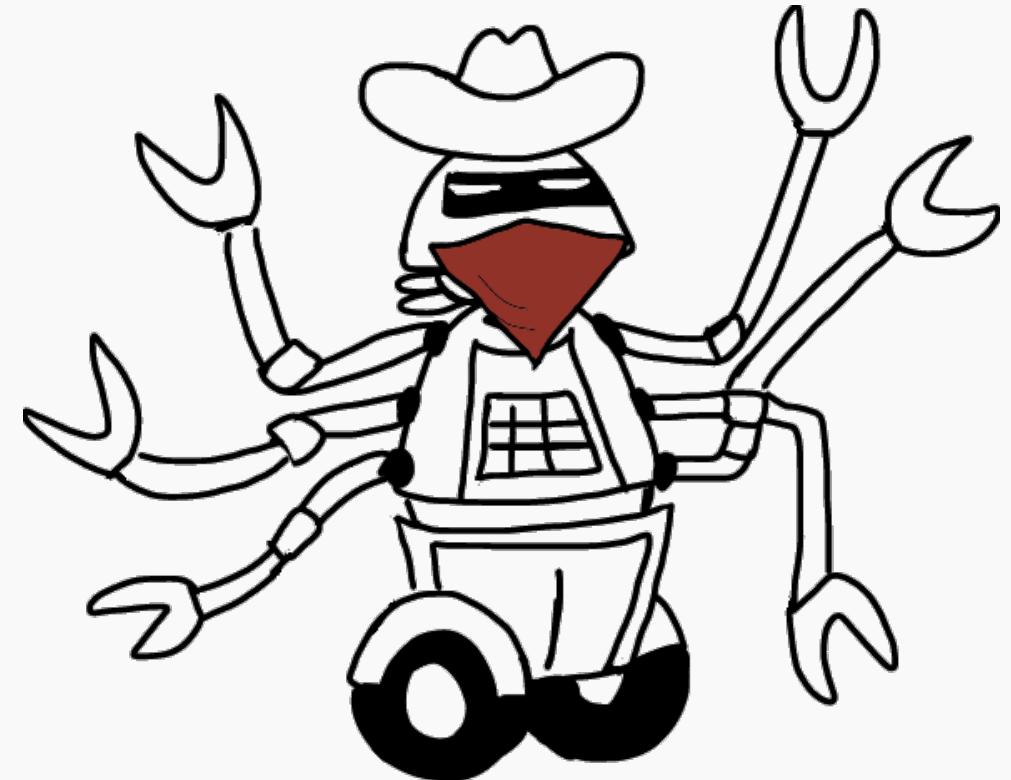
- **Búsqueda** basada en prueba y error para encontrar las mejores acciones.
- **Asociativa**, porque trata de asociar a cada situación (estado) la mejor acción disponible.

*Elegir la mejor acción para cada estado, a partir de prueba y error.*

# Contextual bandits

A los problemas de búsqueda asociativa también se les denomina ***contextual bandits*** (*bandits* con contexto).

- El objetivo es aprender una **política** de comportamiento que mapee cada **situación** con la mejor **acción** posible.
- Los problemas de búsqueda asociativa / *contextual bandits* se encuentran a medio camino entre *K-armed bandits* y los problemas de RL completos.



# *K-armed bandits vs. Contextual bandits vs. RL completo*

Problema	Características
<i>K-armed bandits</i>	<ul style="list-style-type: none"><li>• Elegimos una acción en cada instante de tiempo.</li><li>• Buscamos maximizar la recompensa acumulada a lo largo del tiempo.</li><li>• El valor de las acciones puede ser siempre el mismo (problema estacionario) o variar a lo largo del tiempo (problema no-estacionario).</li></ul>
<i>Contextual bandits</i>	<ul style="list-style-type: none"><li>• Podemos encontrarnos en diferentes situaciones/contextos que harán variar el valor de cada acción.</li><li>• Buscamos aprender a tomar la mejor decisión para cada situación, maximizando las recompensas a largo plazo.</li></ul>
<i>RL completo</i>	<ul style="list-style-type: none"><li>• Buscamos maximizar la recompensa en un ambiente desconocido.</li><li>• Las acciones afectan, no sólo a las recompensas inmediatas, sino también al estado del entorno, que a su vez repercute en las recompensas futuras.</li><li>• Generalmente entornos estocásticos, no estacionarios y con grandes espacios de estados y acciones.</li></ul>

# Ejemplo: tienda online

Problema	Características
<i>K-armed bandits</i>	<ul style="list-style-type: none"><li>• Cada producto a anunciar constituye una acción.</li><li>• En cada instante de tiempo, el agente selecciona el producto a publicitar, y recibe una recompensa positiva si el usuario hace <i>click</i>.</li></ul>
<i>Contextual bandits</i>	<ul style="list-style-type: none"><li>• Añadimos contexto: información sobre el usuario, edad, género, historial de búsqueda, etc.</li><li>• Las recompensas también varían dependiendo del usuario que haga <i>click</i> sobre el anuncio.</li></ul>
<i>RL completo</i>	<ul style="list-style-type: none"><li>• Las acciones ahora pueden repercutir en el entorno y recompensas futuras.</li><li>• Por ejemplo, mostrar un anuncio de forma repetitiva puede molestar al usuario y hacer que abandone el sitio web, evitando que vuelva a hacer <i>click</i> en cualquier otro anuncio.</li></ul>

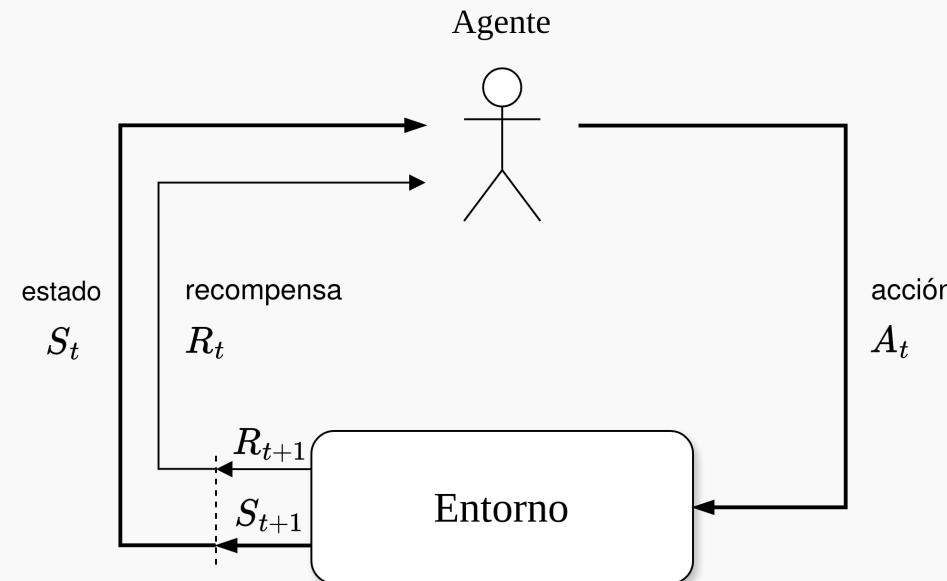
# Procesos de decisión de Markov

---

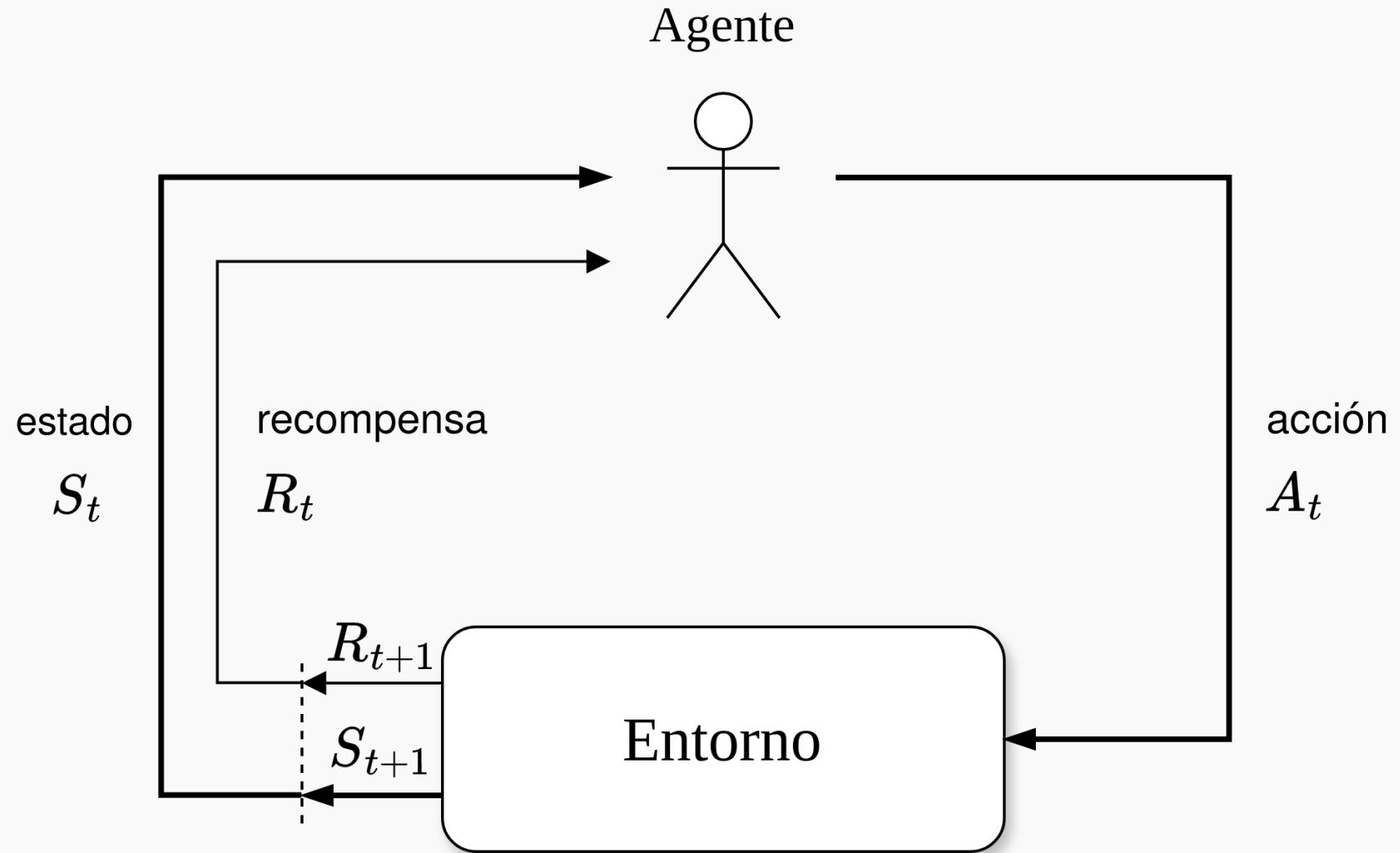
# Procesos de decisión de Markov

El marco formal empleado para definir problemas de aprendizaje por refuerzo es el de **proceso de decisión de Markov (MDP)**.

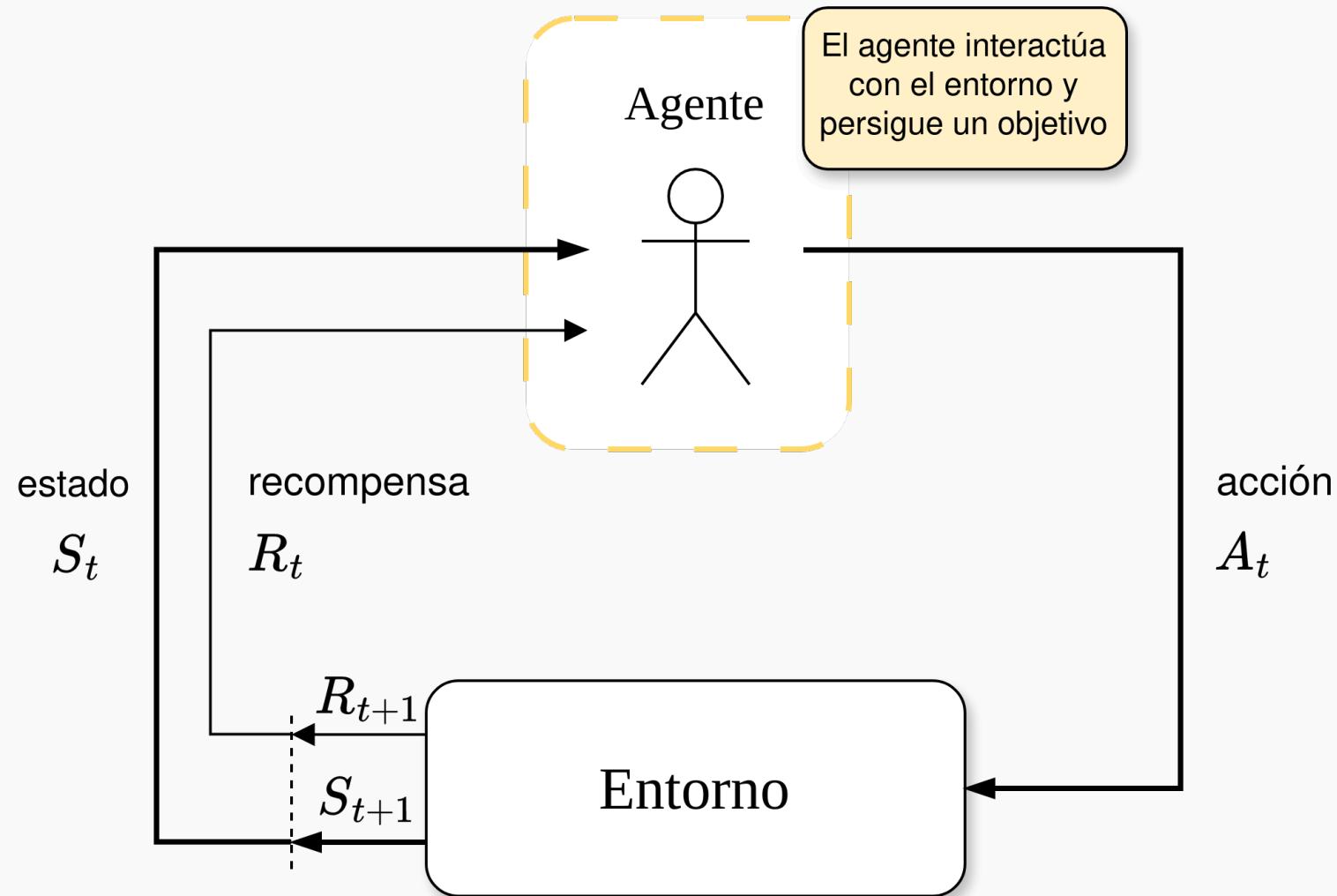
La interacción **agente-entorno** en un problema de RL puede representarse como un MDP finito de la siguiente manera:



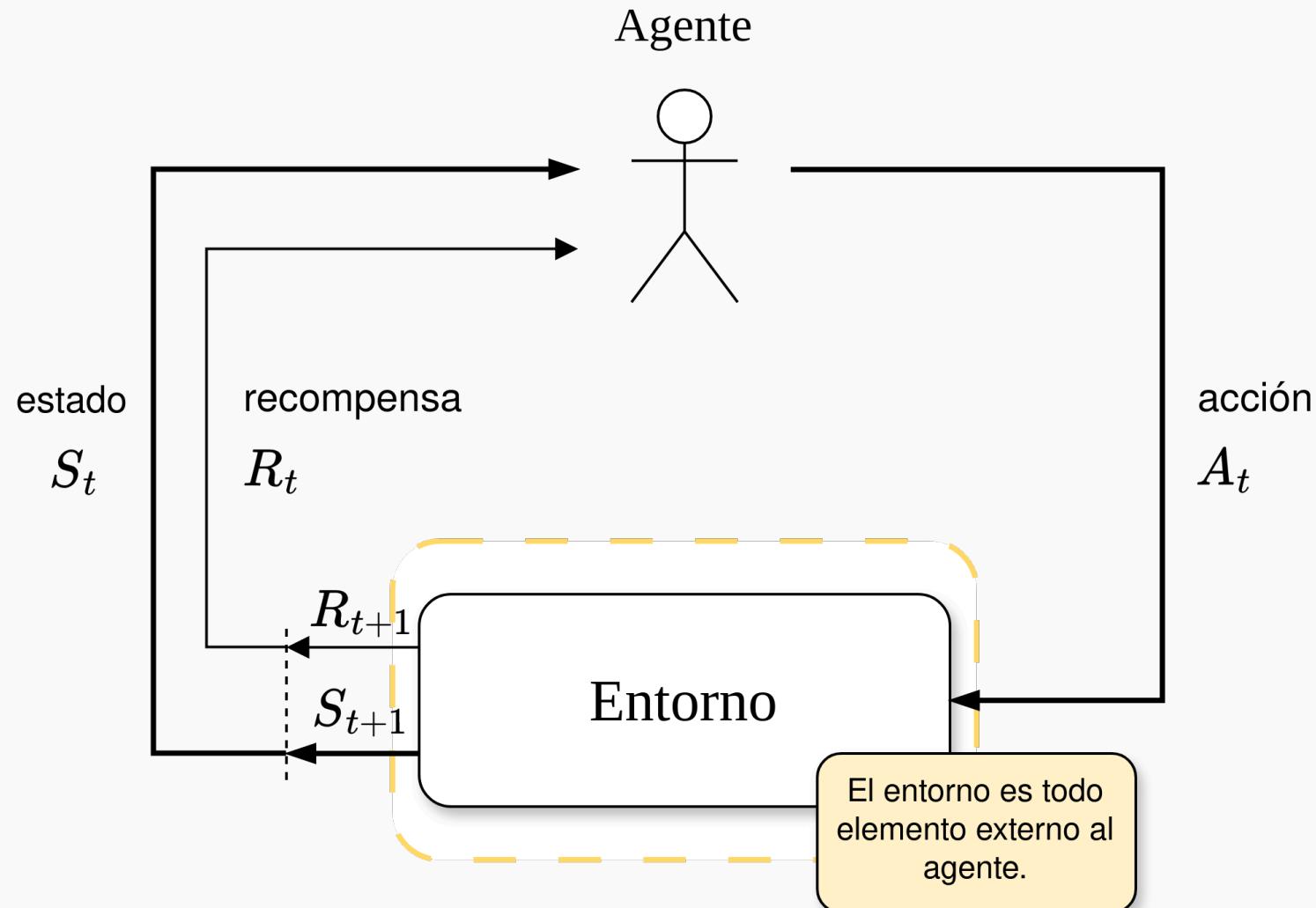
# Proceso de decisión de Markov (MDP)



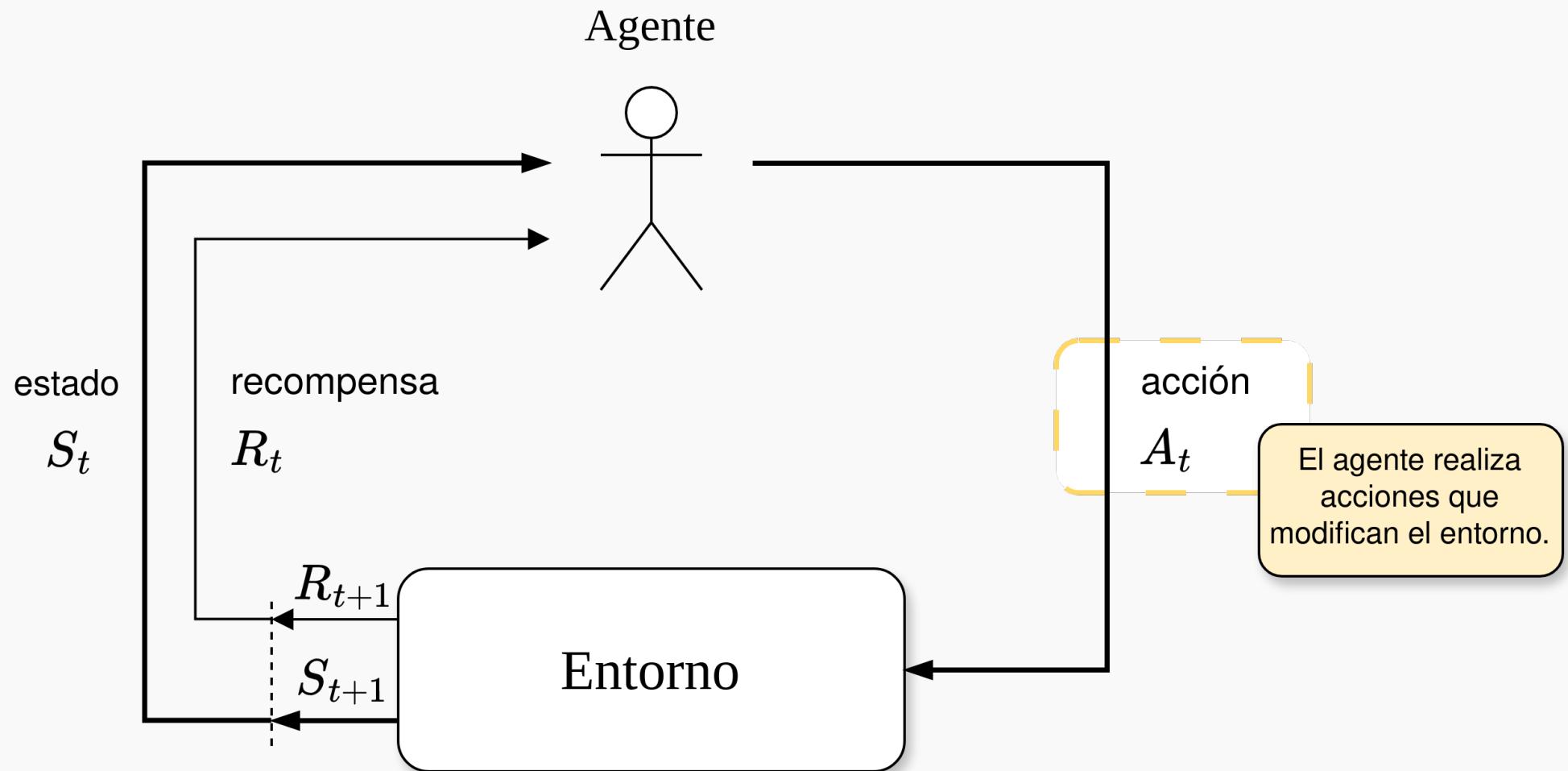
# MDP: agente



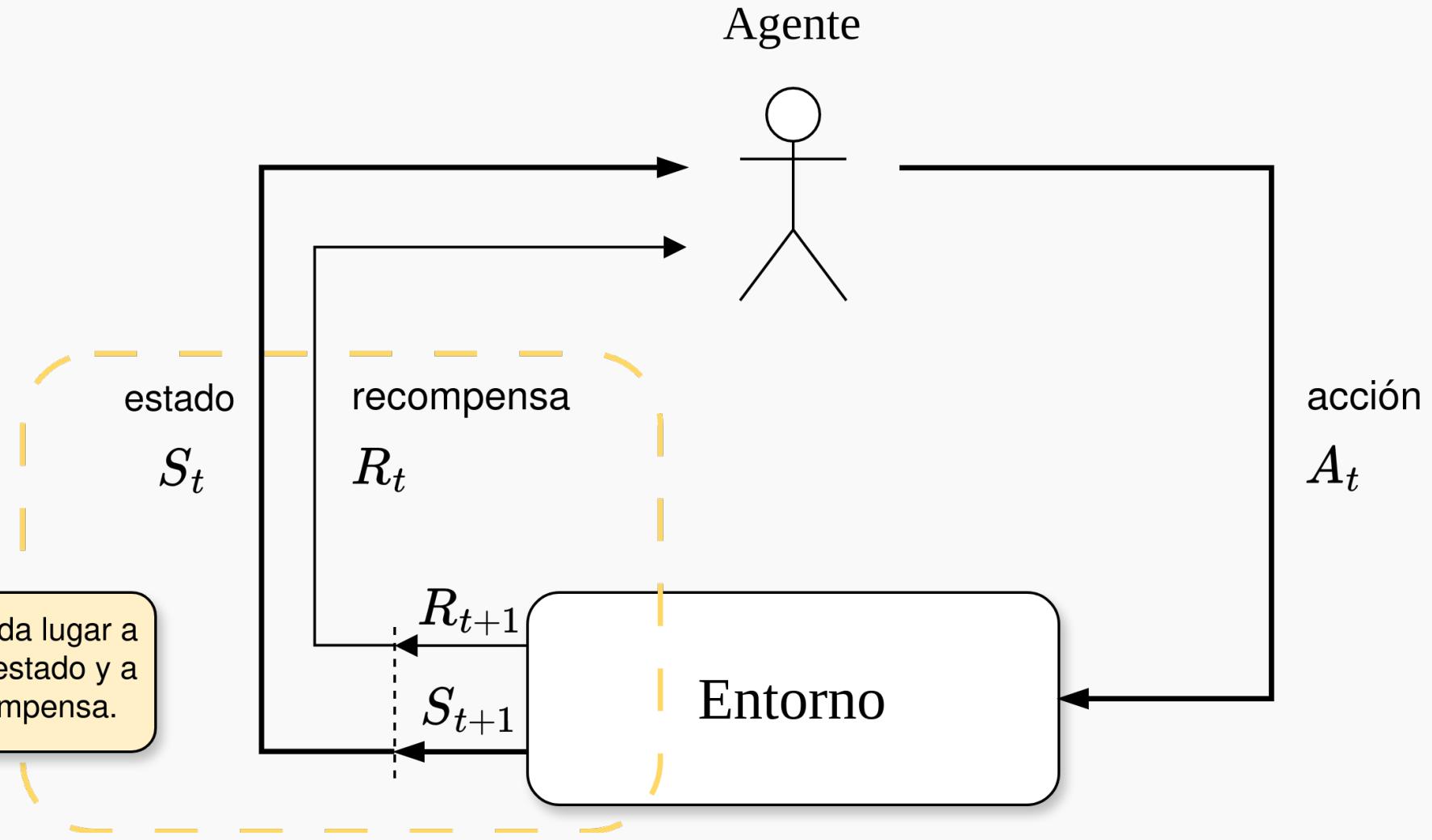
# MDP: entorno



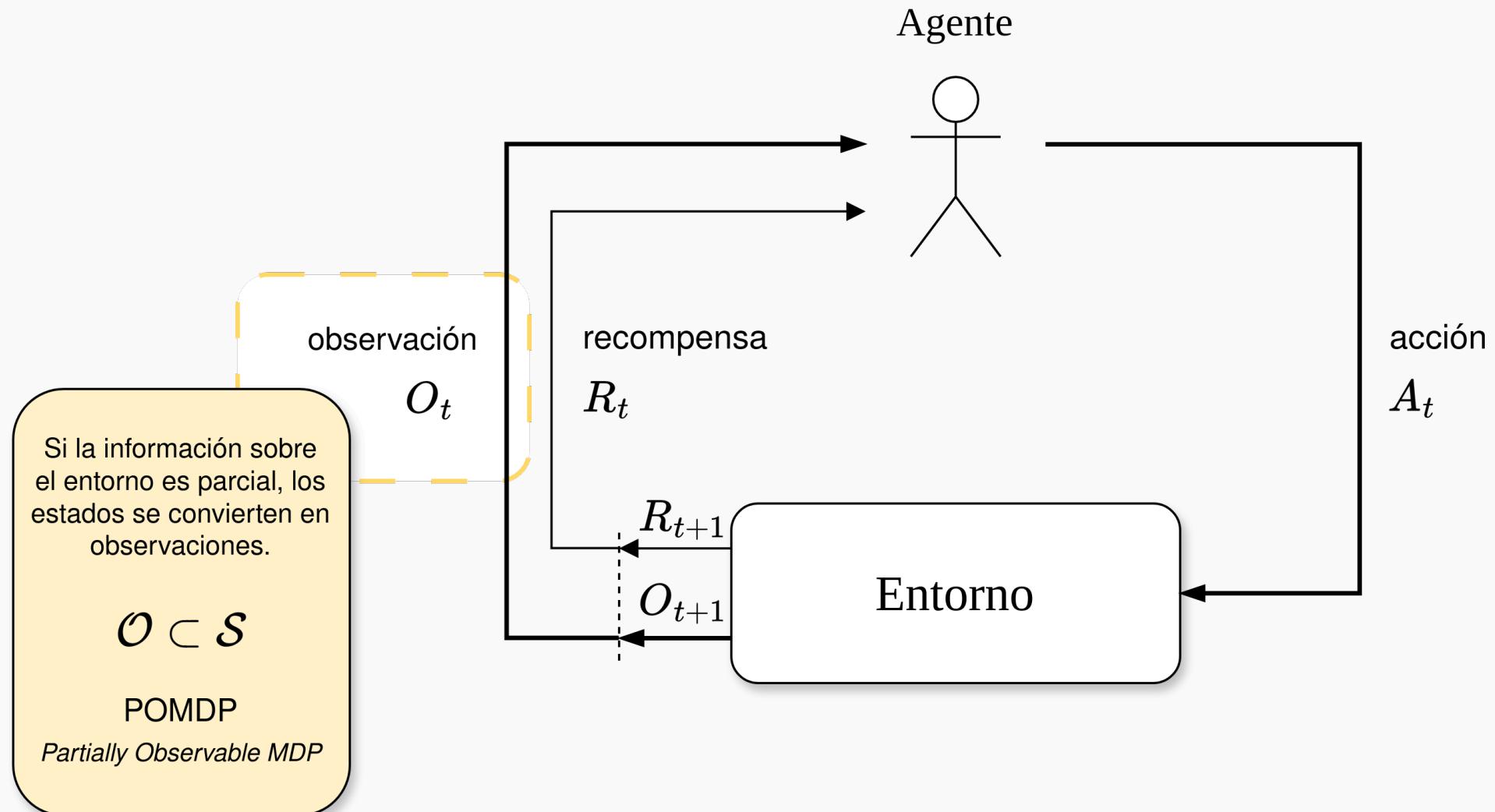
# MDP: acción



# MDP: estado y recompensa



# MDPs parcialmente observables (POMDPs)



# Interacción agente-entorno

El agente interactúa con el entorno a lo largo de una secuencia de pasos o **timesteps**.

En cada instante de tiempo:

1. El agente percibe el **estado actual**  $S_t$  y realiza una **acción**  $A_t$ .
2. El entorno se ve modificado por dicha acción.
3. El agente percibe el **nuevo estado** del entorno  $S_{t+1}$  y recibe una **recompensa**  $R_{t+1}$ .

Esta interacción da lugar a una secuencia de estados, acciones y recompensas denominada **trayectoria**:

$$\tau = \{S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_{T-1}, A_{T-1}, R_T, S_T\}$$

# Probabilidad de transición

En un **MDP finito**, los conjuntos  $\mathcal{S}$  (estados),  $\mathcal{A}$  (acciones) y  $\mathcal{R}$  (recompensas) son finitos.

$R_t$  y  $S_t$  son variables aleatorias con distribuciones de probabilidad bien definidas, que solamente dependen del estado anterior  $S_{t-1}$  y de la acción realizada  $A_t$ .

Es decir, para todo  $s' \in \mathcal{S}, r \in \mathcal{R}$ , existe la probabilidad de que estos valores se den en un instante  $t$  dados unos valores particulares para  $s \in \mathcal{S}$  y  $a \in \mathcal{A}$ :

$$p(s', r | s, a) = \Pr\{S_t = s', R_T = r | S_{t-1} = s, A_{t-1} = a\}$$

Esta función define las **dinámicas del MDP**:

$$p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$$

# Propiedad de Markov

Se cumple que:

$$\sum_{s' \in S, r \in \mathcal{R}} p(s', r | s, a) = 1, \forall s \in S, a \in \mathcal{A}(s)$$

- En un MDP, las probabilidades de transición dependen únicamente del estado y acción inmediatamente previos ( $S_{t-1}, A_{t-1}$ ).
- Es decir, un estado  $S_t$  codifica *toda* la información referente a la interacción agente-entorno previa, y es la **única** información necesaria para elegir la acción a realizar.

Es lo que definimos como **PROPIEDAD DE MARKOV**.

# Propiedad de Markov

## Propiedad de Markov

El estado actual  $S_t$  en un MDP contiene toda la información relevante de los estados pasados  $S_{t-1}, S_{t-2}, \dots, S_0$ .

Por tanto, la transición a un nuevo estado  $S_{t+1}$  no requiere de información sobre los estados previos al estado actual:

$$\Pr[S_{t+1} | S_t] = \Pr[S_{t+1} | S_0, S_1, \dots, S_t]$$

*El futuro es independiente del pasado, dado el presente.*

# Transición entre estados

La siguiente fórmula representa la **regla de transición** entre estados en un MDP:

$$p(s' | s, a) = \Pr\{S_t = s' | S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in \mathcal{R}} p(s', r | s, a)$$

Proporciona la probabilidad de transicionar a  $s'$  partiendo de  $s$  y ejecutando  $a$ .

# Transición entre estados

*¿Por qué necesitamos saber las probabilidades de transición?*

En problemas de RL **deterministas**, una acción  $a$  desde un estado  $s$  siempre conduce al mismo estado  $s'$ .

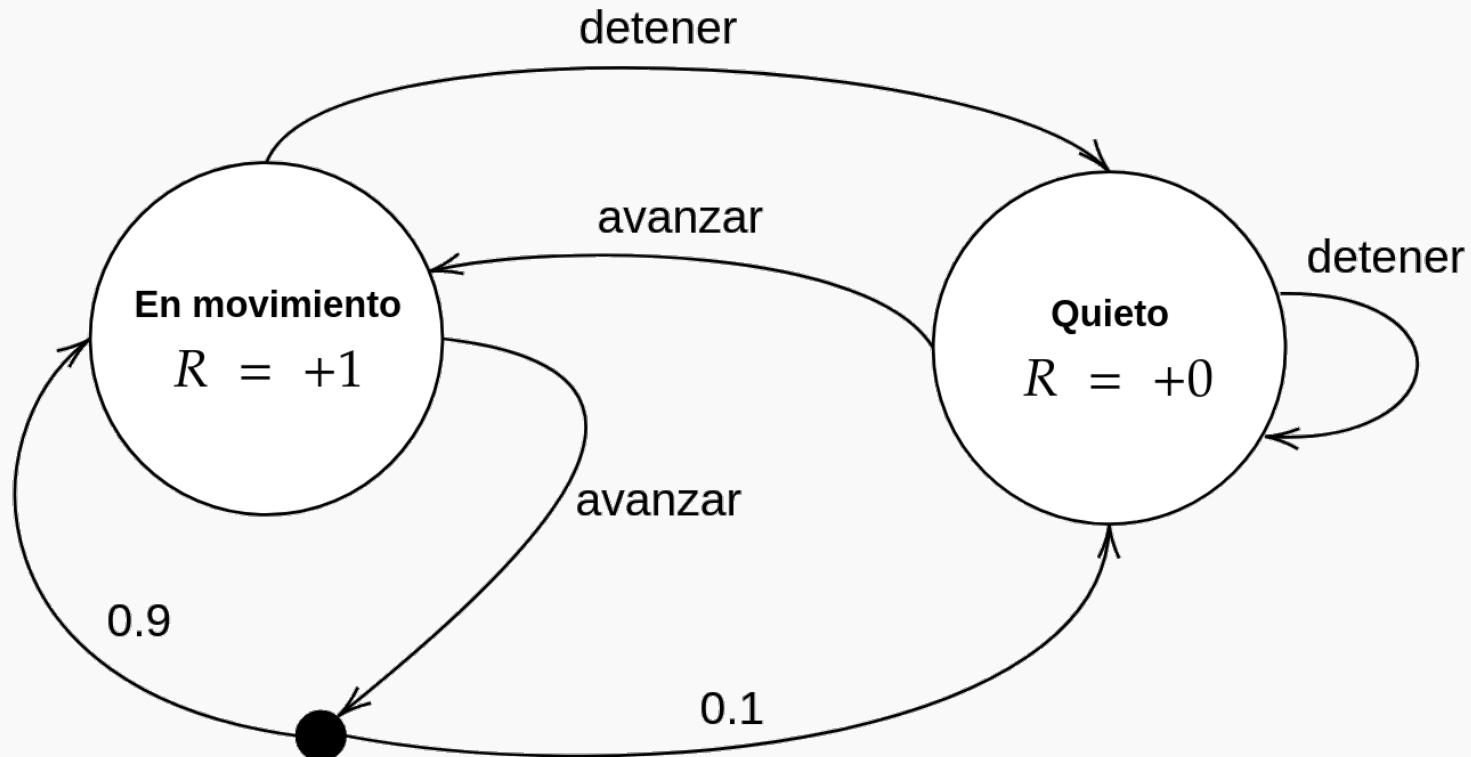
- Ej. ajedrez → reglas fijas.

Pero en problemas de RL **estocásticos**, la misma acción  $a$  puede llevar a diferentes estados  $s'$ .

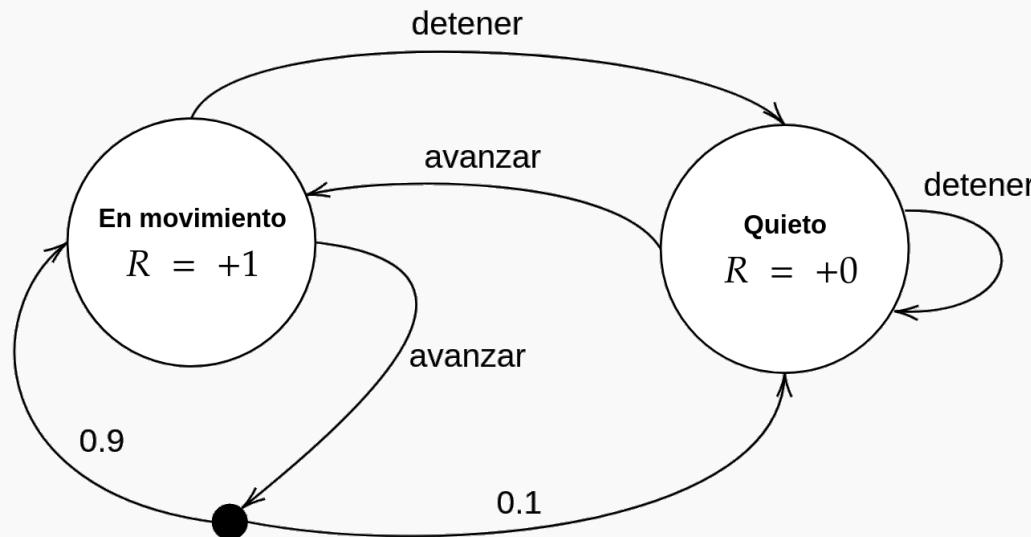
- Ej. controlar la trayectoria de un dron → viento.

*Veamos un ejemplo...*

# Ejemplo



# Ejemplo



$$p(s_1, 0 \mid s_0, a_1) =$$

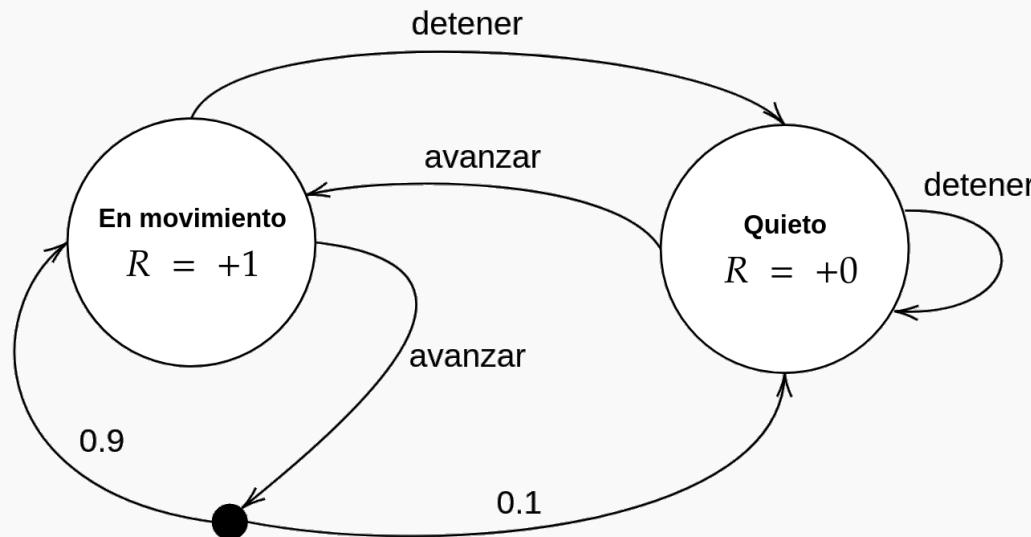
$$\mathcal{S} = \{s_0, s_1\}$$

- $s_0$  : en movimiento
- $s_1$  : quieto

$$\mathcal{A} = \{a_0, a_1\}$$

- $a_0$  : avanzar
- $a_1$  : detener

# Ejemplo



$$\mathcal{S} = \{s_0, s_1\}$$

- $s_0$  : en movimiento
- $s_1$  : quieto

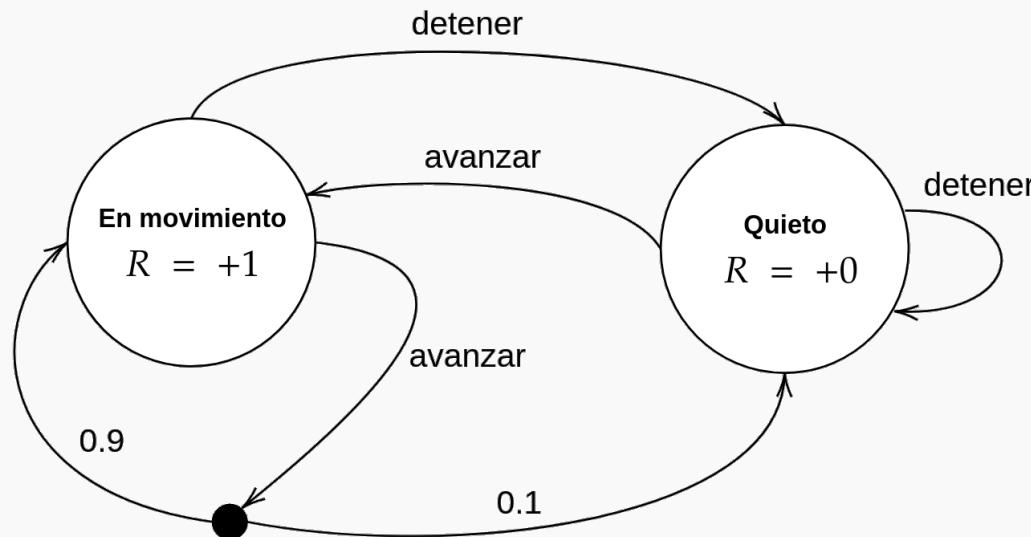
$$\mathcal{A} = \{a_0, a_1\}$$

- $a_0$  : avanzar
- $a_1$  : detener

$$p(s_1, 0 | s_0, a_1) = 1$$

$$p(s_0 | s_0, a_0) =$$

# Ejemplo



$$\mathcal{S} = \{s_0, s_1\}$$

- $s_0$  : en movimiento
- $s_1$  : quieto

$$\mathcal{A} = \{a_0, a_1\}$$

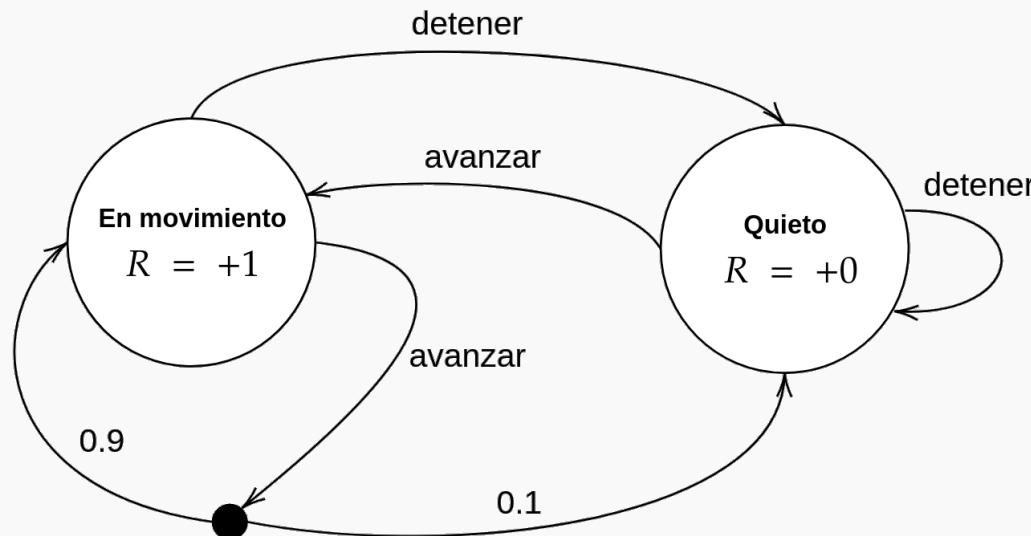
- $a_0$  : avanzar
- $a_1$  : detener

$$p(s_1, 0 | s_0, a_1) = 1$$

$$p(s_0 | s_0, a_0) = 0.9$$

$$p(s_1 | s_0, a_0) =$$

# Ejemplo



$$\mathcal{S} = \{s_0, s_1\}$$

- $s_0$  : en movimiento
- $s_1$  : quieto

$$\mathcal{A} = \{a_0, a_1\}$$

- $a_0$  : avanzar
- $a_1$  : detener

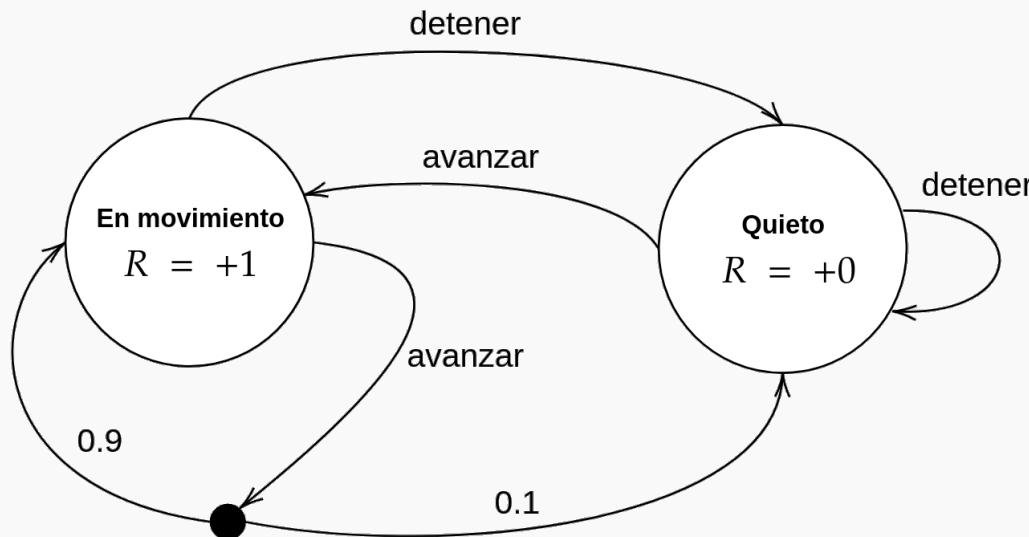
$$p(s_1, 0 | s_0, a_1) = 1$$

$$p(s_0 | s_0, a_0) = 0.9$$

$$p(s_1 | s_0, a_0) = 0.1$$

$$p(s_1, 1 | s_1, a_0) =$$

# Ejemplo



$$\mathcal{S} = \{s_0, s_1\}$$

- $s_0$  : en movimiento
- $s_1$  : quieto

$$\mathcal{A} = \{a_0, a_1\}$$

- $a_0$  : avanzar
- $a_1$  : detener

$$p(s_1, 0 | s_0, a_1) = 1$$

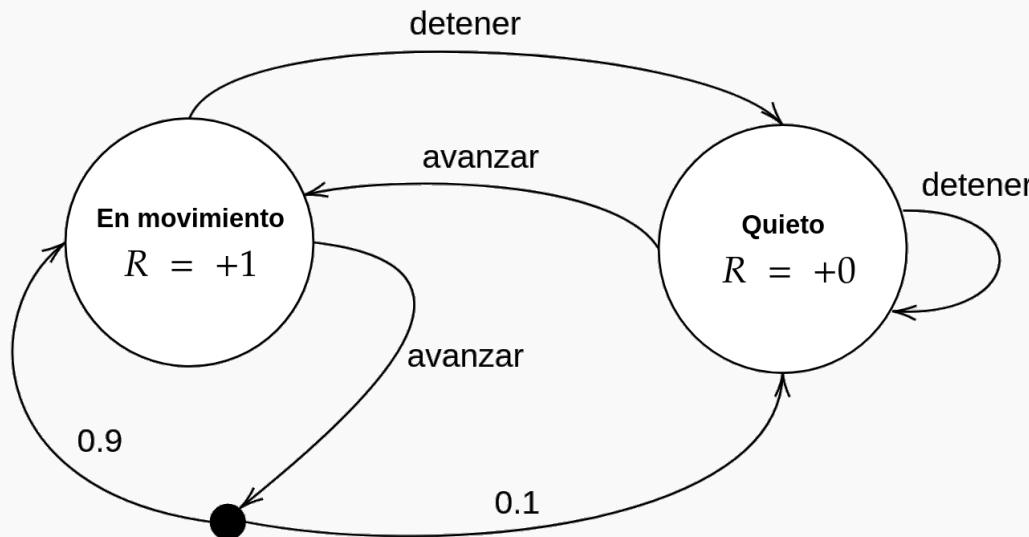
$$p(s_0 | s_0, a_0) = 0.9$$

$$p(s_1 | s_0, a_0) = 0.1$$

$$p(s_1, 1 | s_1, a_0) = 0$$

$$p(s_0, 1 | s_1, a_1) =$$

# Ejemplo



$$\mathcal{S} = \{s_0, s_1\}$$

- $s_0$  : en movimiento
- $s_1$  : quieto

$$\mathcal{A} = \{a_0, a_1\}$$

- $a_0$  : avanzar
- $a_1$  : detener

$$p(s_1, 0 | s_0, a_1) = 1$$

$$p(s_0 | s_0, a_0) = 0.9$$

$$p(s_1 | s_0, a_0) = 0.1$$

$$p(s_1, 1 | s_1, a_0) = 0$$

$$p(s_0, 1 | s_1, a_1) = 0$$

**Propiedad de Markov:** las transiciones sólo dependen del **estado actual** (y la **acción realizada**).

# Recompensa esperada

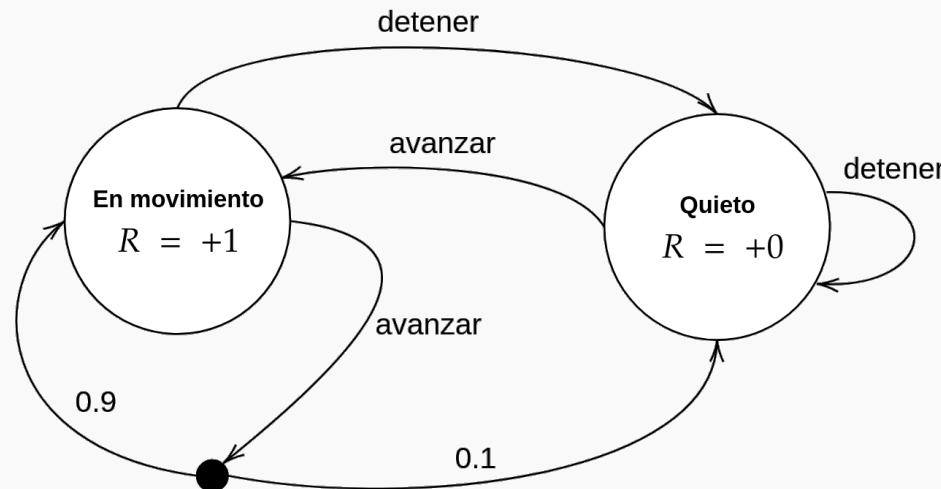
¿Qué recompensa podemos esperar de un par acción-estado?

$$r(s, a) = \mathbb{E}[R_t \mid S_{t+1} = s, A_{t-1} = a] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r \mid s, a)$$

¿Qué recompensa podemos esperar de una tripleta estado-acción-estado?

$$r(s, a, s') = \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a, S_t = s'] = \sum_{r \in \mathcal{R}} r \frac{p(s', r \mid s, a)}{p(s' \mid s, a)}$$

# Ejemplo



$$r(s_0, a_1) =$$

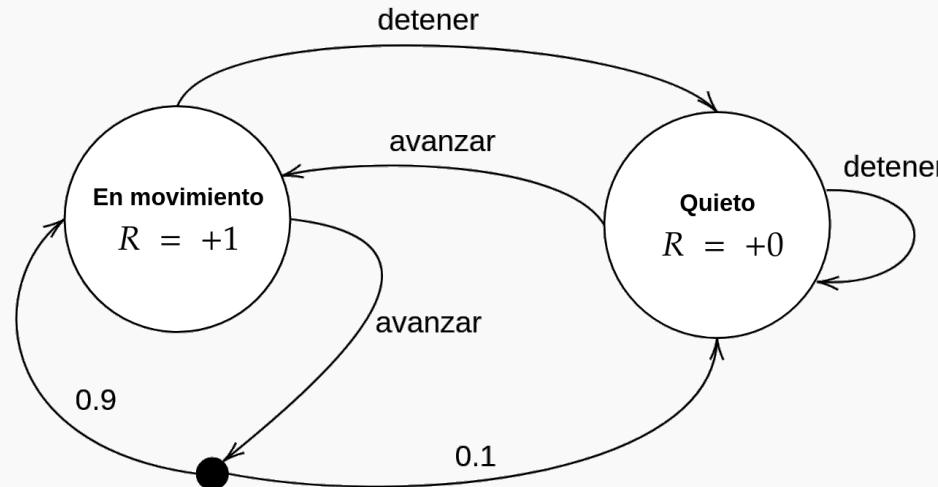
$$\mathcal{S} = \{s_0, s_1\}$$

- $s_0$  : en movimiento
- $s_1$  : quieto

$$\mathcal{A} = \{a_0, a_1\}$$

- $a_0$  : avanzar
- $a_1$  : detener

# Ejemplo



$$\mathcal{S} = \{s_0, s_1\}$$

- $s_0$  : en movimiento
- $s_1$  : quieto

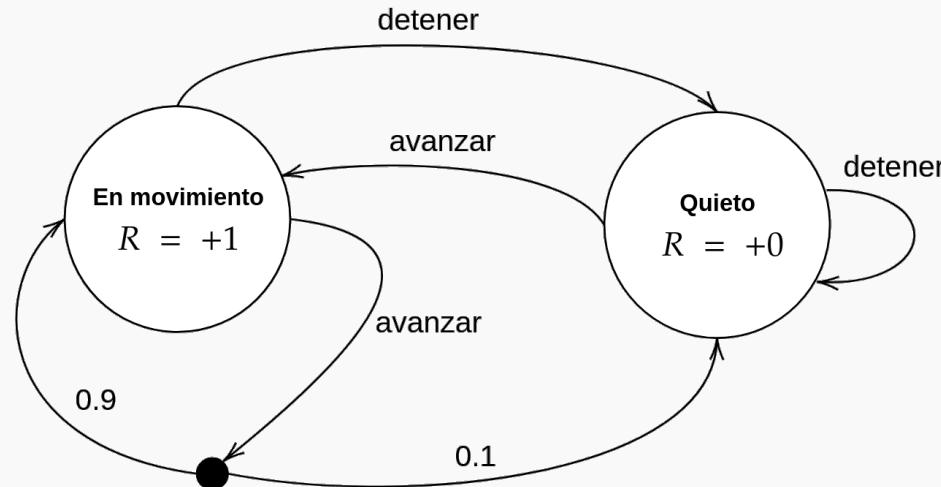
$$\mathcal{A} = \{a_0, a_1\}$$

- $a_0$  : avanzar
- $a_1$  : detener

$$r(s_0, a_1) = 0$$

$$r(s_1, a_0) =$$

# Ejemplo



$$\mathcal{S} = \{s_0, s_1\}$$

- $s_0$  : en movimiento
- $s_1$  : quieto

$$\mathcal{A} = \{a_0, a_1\}$$

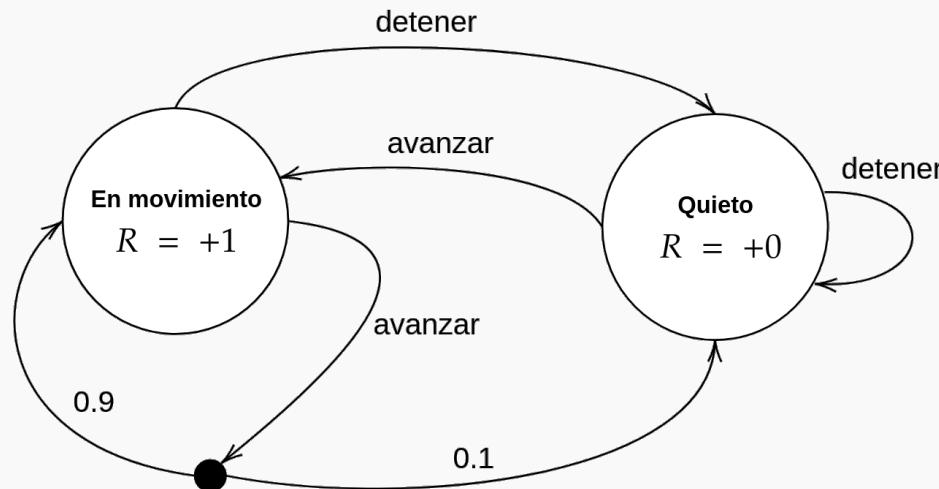
- $a_0$  : avanzar
- $a_1$  : detener

$$r(s_0, a_1) = 0$$

$$r(s_1, a_0) = 1$$

$$r(s_0, a_0) =$$

# Ejemplo



$$\mathcal{S} = \{s_0, s_1\}$$

- $s_0$  : en movimiento
- $s_1$  : quieto

$$\mathcal{A} = \{a_0, a_1\}$$

- $a_0$  : avanzar
- $a_1$  : detener

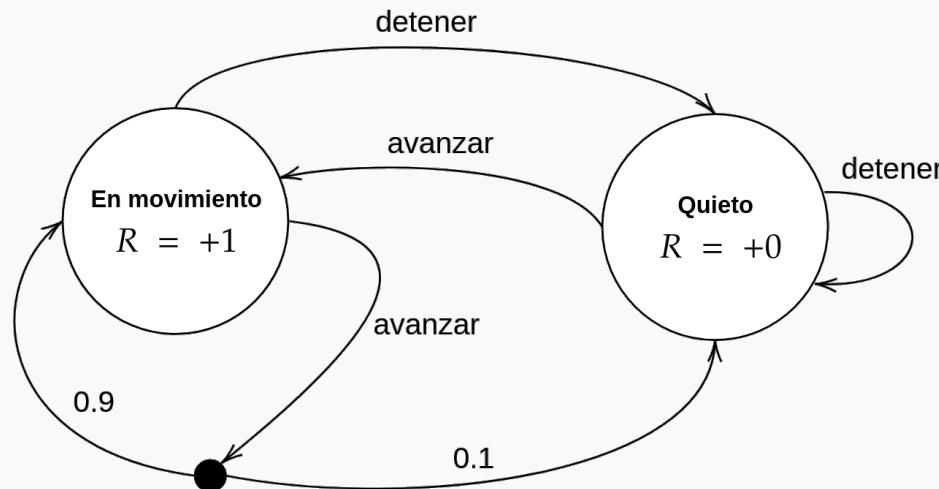
$$r(s_0, a_1) = 0$$

$$r(s_1, a_0) = 1$$

$$r(s_0, a_0) = (1 \cdot 0.9) + (0 \cdot 0.1) = 0.9$$

$$r(s_0, a_0, s_0) =$$

# Ejemplo



$$\mathcal{S} = \{s_0, s_1\}$$

- $s_0$  : en movimiento
- $s_1$  : quieto

$$\mathcal{A} = \{a_0, a_1\}$$

- $a_0$  : avanzar
- $a_1$  : detener

$$r(s_0, a_1) = 0$$

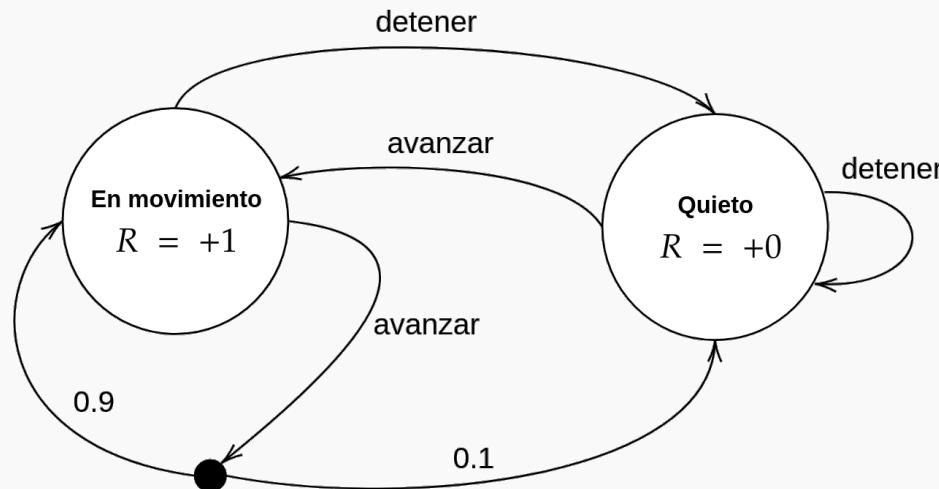
$$r(s_1, a_0) = 1$$

$$r(s_0, a_0) = (1 \cdot 0.9) + (0 \cdot 0.1) = 0.9$$

$$r(s_0, a_0, s_0) = 1$$

$$r(s_1, a_1, s_0) =$$

# Ejemplo



$$\mathcal{S} = \{s_0, s_1\}$$

- $s_0$  : en movimiento
- $s_1$  : quieto

$$\mathcal{A} = \{a_0, a_1\}$$

- $a_0$  : avanzar
- $a_1$  : detener

$$r(s_0, a_1) = 0$$

$$r(s_1, a_0) = 1$$

$$r(s_0, a_0) = (1 \cdot 0.9) + (0 \cdot 0.1) = 0.9$$

$$r(s_0, a_0, s_0) = 1$$

$$r(s_1, a_1, s_0) = ?$$

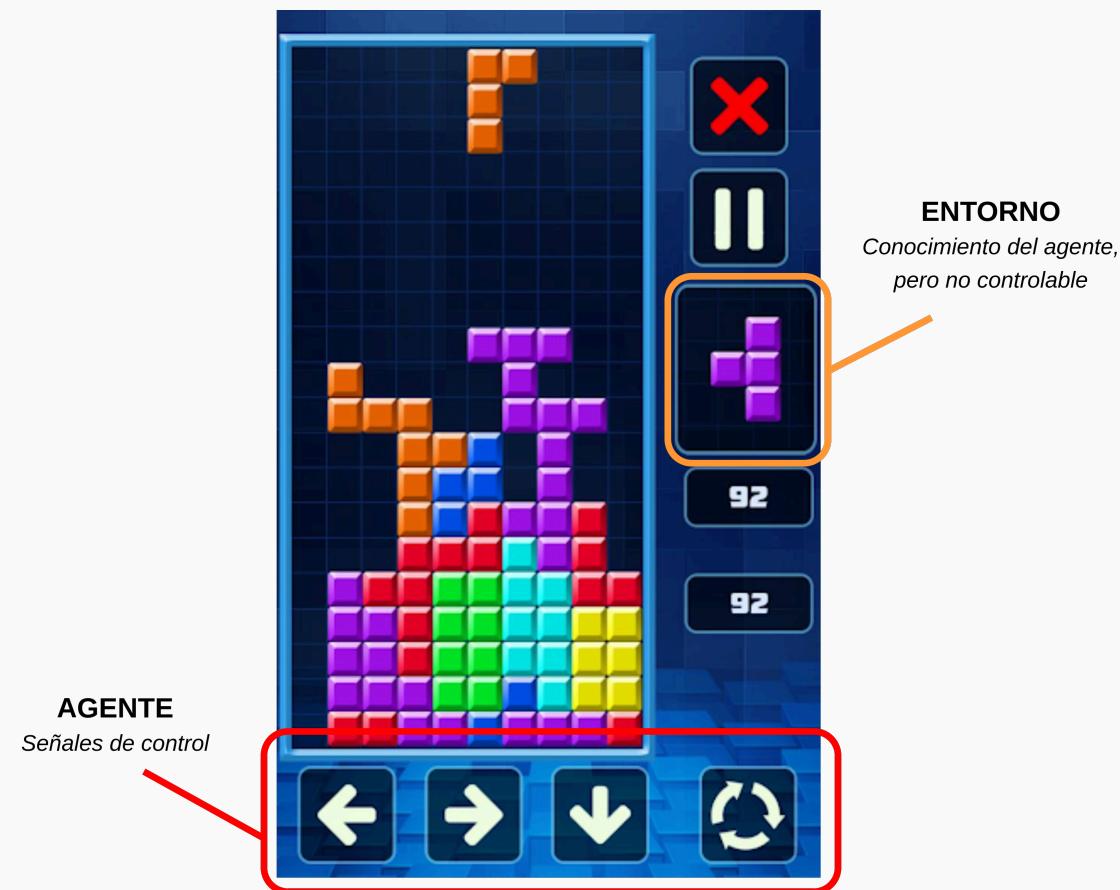
$$r(s, a, s') = \sum_{r \in \mathcal{R}} r \frac{p(s', r | s, a)}{\cancel{p(s' | s, a)}}$$

Algunas consideraciones...

# Frontera agente-entorno

Consideramos **entorno** a todo aquello sobre lo que el agente no tiene control.

Es decir, la **frontera agente-entorno** viene dada por las capacidades de control del agente, no por su conocimiento.



# Interacción agente-entorno

Podemos resumir el aprendizaje basado en interacción en 3 señales:

1. **Acciones**: elecciones, control del agente.
2. **Estados**: información en base a dichas elecciones.
3. **Recompensas**: adecuación al objetivo.

El nivel de abstracción / complejidad de estados y acciones dependerá del problema a tratar.



# Representaciones estructuradas

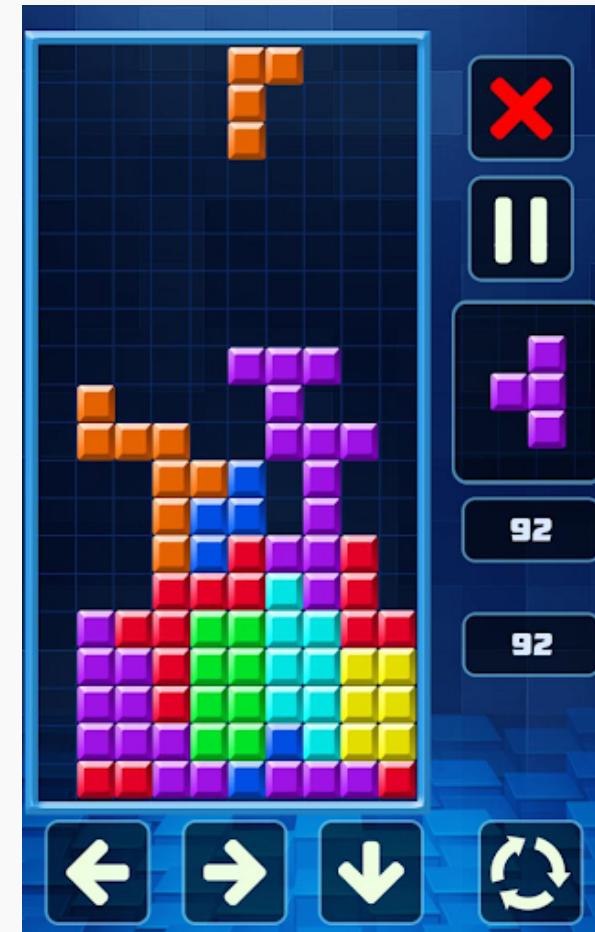
Es común contar con **representaciones estructuradas** de estados y acciones (ej. vectores de valores).

$$S_t : \{M_t, \text{type}, \text{pos}, \text{next}, \text{score}\}$$

- $M_t$ : matriz con posiciones libres (0) u ocupadas (1)
- $\text{type}$ : pieza actual
- $\text{pos}$ : posición de la pieza actual
- $\text{next}$ : próxima pieza
- $\text{score}$ : puntuación acumulada

$$\mathcal{A} : \{0 : \leftarrow, 1 : \rightarrow, 2 : \downarrow, 3 : \uparrow\}$$

$$A_t \in \{0, 1, 2, 3\}$$



# Estados no-markovianos

En un proceso de decisión de Markov, el futuro depende únicamente del estado presente y no de los estados anteriores (**propiedad de Markov**).

Sin embargo, podemos encontrarnos ante problemas con **estados no-markovianos**, donde el estado actual no contiene toda la información relevante para predecir el futuro.

## Estado no-markoviano

Un estado no-markoviano es aquel en el que la información necesaria para tomar una decisión óptima no está completamente representada por el estado actual del sistema.

# Ejemplo

*¿La bola va hacia la niña o hacia el hombre?*



# Ejemplo

- En este ejemplo, no podemos predecir el movimiento de la bola a partir de una sola imagen.
  - Una imagen instantánea no nos ofrece la suficiente información.
  - **Estado no-markoviano.**
- Una posible solución sería **concatenar múltiples fotogramas consecutivos.**



# Temporalidad

Los *time steps* pueden no darse en intervalos de tiempo fijos, sino estar condicionados por eventos.

Por ejemplo:

- Movimiento de piezas en una partida de ajedrez.
- Cambio de valor en el precio de un acción.
- Acceso a la web de venta de un producto.
- ...



# Problemas episódicos y continuados

---

# Problemas episódicos

El **estado inicial** de un problema de RL es el estado en el cual comienza la interacción del agente con el entorno.

Alcanzar un **estado terminal** supone el fin de esta interacción.

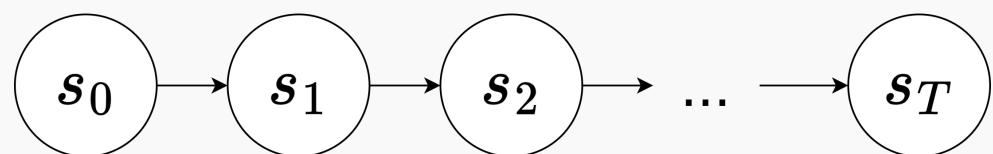
Un problema de RL puede contar con múltiples estados iniciales y finales.

- Estados no terminales:  $S$ .
- Estados terminales y no terminales:  $S^+$ .

Definimos así el concepto de **episodio**.

## Episodio

Secuencia de *time steps* desde un estado inicial hasta un estado terminal.



La longitud  $T$  de un episodio no tiene por qué ser fija, y puede variar entre episodios.

# Problemas episódicos

## Problema episódico

Problema dividido en una secuencia finita de estados, desde un estado inicial hasta un estado terminal.

¿Y si el problema no tiene fin?

# Problemas continuados

En los **problemas continuados** (vs. episódicos), no existen episodios que finalicen en un estado terminal.

- Es decir, **no hay estados terminales**.
- Por tanto, el problema no finaliza en un *time step*  $T$  concreto ( $T = \infty$ ).

## Problema continuado

Problema consistente en una secuencia **infinita** de estados, partiendo de un estado inicial.

Suelen ser problemas más cercanos a la realidad (control térmico, robótica, ...).

# Objetivos y recompensas

---

# Objetivos y recompensas

*¿Qué relación hay entre las señales de recompensa y los objetivos del agente?*

# Objetivos y recompensas

*¿Qué relación hay entre las señales de recompensa y los objetivos del agente?*

*¿Cómo se realiza el cálculo de las recompensas?*

# Objetivos y recompensas

*¿Qué relación hay entre las señales de recompensa y los objetivos del agente?*

*¿Cómo se realiza el cálculo de las recompensas?*

*¿Cómo condicionamos el comportamiento del agente en base a dichas recompensas?*

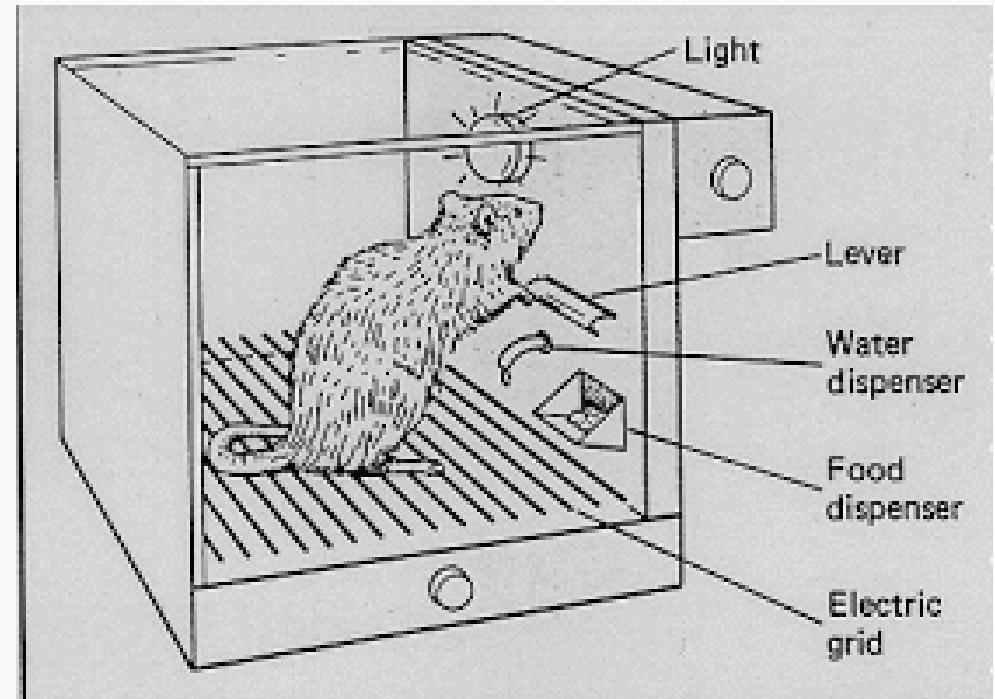
# Recompensa

Para guiar a un agente hacia su **objetivo**, empleamos **recompensas**.

Una señal de **recompensa** es un valor numérico que indica al agente si su comportamiento le acerca o no a su objetivo.

$$R_t \in \mathbb{R}$$

Buscamos maximizar la **recompensa acumulada** a largo plazo, no sólo las **recompensas inmediatas**.



# Reward hypotheses

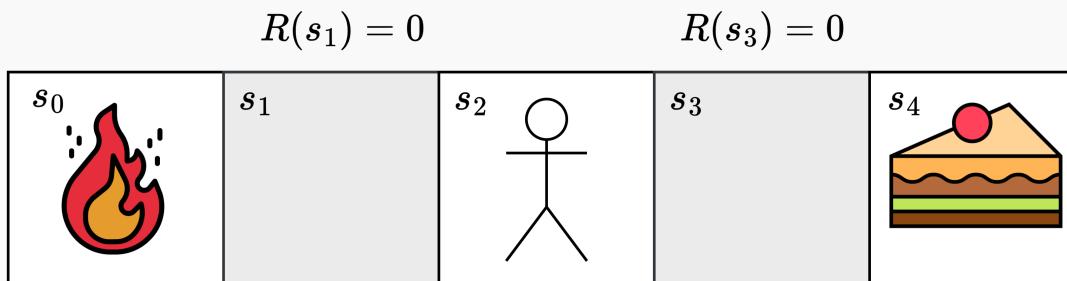
*Todo lo que entendemos como objetivo o propósito puede interpretarse como la maximización del valor esperado para una suma acumulada de una señal escalar (llamada recompensa).*

 Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT press. (p. 53).

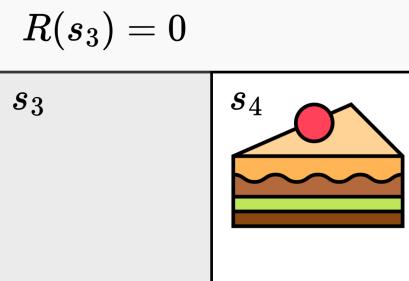
El uso de una señal de recompensa para formalizar la idea de **objetivo** es uno de los aspectos más distintivos del aprendizaje por refuerzo.

# Función de recompensa

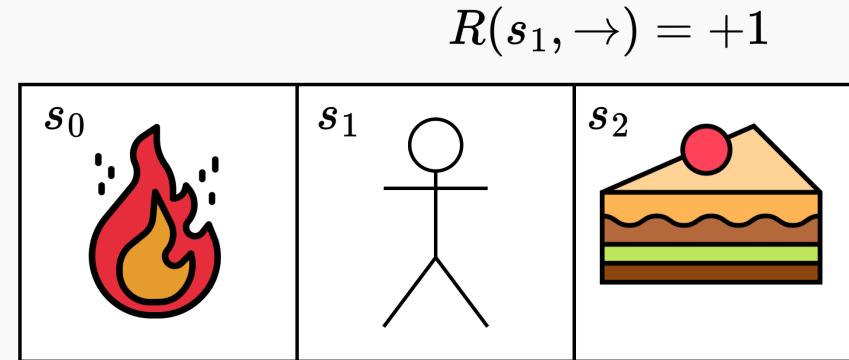
Las recompensas que el agente recibe vienen dadas por una **función de recompensa**, que generalmente depende del **estado alcanzado**, o de la **acción realizada**, esto es:  $r_t = R(S_t)$ , o bien:  $r_t = R(S_t, A_t)$ .



$$R(s_0) = -1$$



$$R(s_2) = 0$$



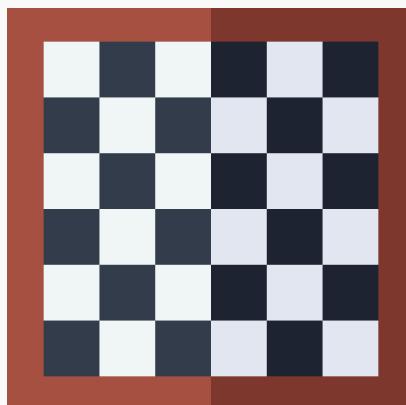
$$R(s_4) = +1$$

$$R(s_1, \rightarrow) = +1$$

$$R(s_1, \leftarrow) = -1$$

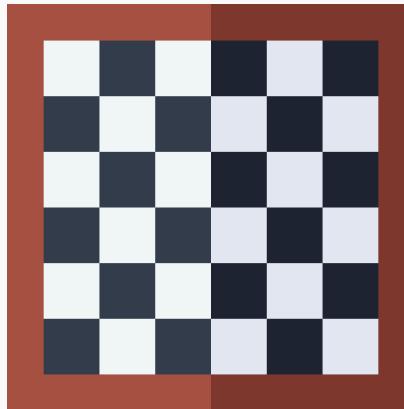
# Objetivos y subobjetivos

El **objetivo** de un agente de RL es **maximizar la recompensa acumulada (return)** a lo largo del tiempo.



- **Agente orientado a subobjetivos:** cada vez que se come una pieza,  $R = +1$  (o valor variable, dependiendo de la pieza comida).
- **Agente orientado a objetivos:** si se gana,  $R = +1$ , si se pierde:  $R = -1$ , si se empata (*tablas*):  $R = 0$ .

# Objetivos y subobjetivos



La recompensa no debe orientarse exclusivamente hacia el cumplimiento de **subobjetivos**, sino hacia el cumplimiento de un **objetivo final**.

- Ej. *comer piezas sin ningún criterio.*

Si, por ejemplo, queremos fomentar cierto comportamiento desde un principio, es mejor emplear otros recursos, como valores iniciales optimistas.

# Retorno

El **retorno**, o *recompensa acumulada*, es el valor que tratamos de maximizar.

Se define de la siguiente manera:

$$G_t = R_{t+1} + R_{t+2} + \dots + R_{T-1} + R_T$$

Siendo  $T$  el último *time step* del **episodio** (o de una ventana de tiempo determinada).

## Retorno

Suma de las recompensas a obtener desde el momento presente hasta el final de un episodio o ventana de tiempo determinada.

# Retorno

$$G_t = R_{t+1} + R_{t+2} + \dots + R_{T-1} + R_T$$

Esta formulación es válida para **problemas episódicos**, pero...

$$G_t = R_{t+1} + R_{t+2} + \dots + R_{T-1} + R_T$$

Esta formulación es válida para **problemas episódicos**, pero...

**¿Qué ocurre en los problemas continuados?**

- Si el *time step* final es  $T = \infty$ , la recompensa esperada es una suma infinita  $G_t = \infty$ .

Necesitamos reformular la definición de recompensa acumulada.

# Retorno descontado

Introducimos el concepto de **retorno descontado**:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Donde  $\gamma \in [0, 1]$  se denomina **factor de descuento** (*discount factor*).

# Factor de descuento

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- El factor de descuento determina el **valor presente asignado a recompensas futuras**.
- Una recompensa recibida en  $k$  *time steps* futuros tiene un valor  $\gamma^{k-1}$  veces lo que valdría en el *time step* actual.
- Si  $\gamma = 0$ , el agente solamente tendrá en cuenta las **recompensas inmediatas**. Se trata de un *agente miope*, que sólo tiene en cuenta  $R_{t+1}$  para elegir  $A_t$ .
- A medida que  $\gamma$  se approxima a 1, el agente tendrá más en cuenta aquellas acciones que maximicen las **recompensas futuras**.

# Definición recursiva

El retorno descontado puede definirse **de forma recursiva**:

$$\begin{aligned}G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\&= R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \dots) \\&= R_{t+1} + \gamma G_{t+1}\end{aligned}$$

# Definición recursiva

El retorno descontado puede definirse **de forma recursiva**:

$$\begin{aligned}G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \dots \\&= R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \dots) \\&= \underbrace{R_{t+1}}_{\substack{\text{Recompensa} \\ \text{inmediata}}} + \underbrace{\gamma G_{t+1}}_{\substack{\text{Return} \\ \text{descontado} \\ \text{desde } t+1}}\end{aligned}$$

# Definición recursiva

El retorno descontado puede definirse **de forma recursiva**:

$$G_t = R_{t+1} + \gamma G_{t+1}$$

Esta formulación será importante para la teoría y algoritmos de RL que veremos más adelante.

## Recompensa constante

$$G_t = R_{t+1} + \gamma G_{t+1}$$

La definición recursiva de  $G_t$  es válida para todo *time step*  $t < T$ , incluso si la terminación ocurre en  $t + 1$ , siempre que definamos  $G_T = 0$ .

Por otro lado, aunque  $G_t$  sea una suma infinita, se convierte en finita si la recompensa es siempre  $> 0$  y constante (con  $\gamma < 1$ ).

- Por ejemplo, si la recompensa es siempre  $+1$ , el retorno será:

$$G_t = \sum_{k=0}^{\infty} \gamma^k = \frac{1}{1 - \gamma}$$

# ¿Cómo definir la recompensa?

Existen diferentes formas de guiar el aprendizaje mediante una funciones de recompensa.

El aprendizaje puede basarse en **refuerzos positivos** o **negativos**. Por ejemplo:

**Recompensa basada en objetivos:**

- +1 si el agente alcanza un objetivo
- +0 en cualquier otro caso

**Recompensa basada en penalizaciones:**

- 1 por *time step* empleado
- +0 cuando se alcanza el objetivo.

Cualquier representación de la función de recompensa tiene sus *pros* y sus *contras*, por lo que la elección de una u otra dependerá del problema que tratemos de abordar.

# ¿Cómo definir la recompensa?

- ¿Qué definición de recompensa emplearías para entrenar a un robot a salir de un laberinto?

# ¿Cómo definir la recompensa?

- ¿Qué definición de recompensa emplearías para entrenar a un robot a salir de un laberinto?
- ¿Y para conducir un coche autónomo?

# *Inverse reinforcement learning*

Existen formas alternativas de aplicar las funciones de recompensa.

Por ejemplo, el *inverse reinforcement learning* (RL inverso) consiste en plantear un ejemplo de comportamiento óptimo y hacer que el agente adivine la recompensa a maximizar que se asocia con este.

**Comportamiento → Recompensa**

vs.

**Recompensa → Comportamiento**

## En resumen...

Un MDP se define por la tupla:

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$$

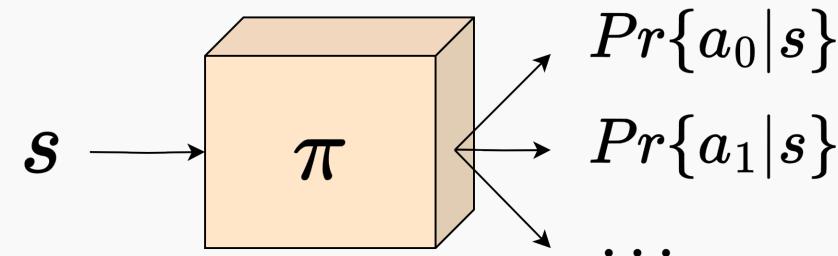
- Un conjunto de **estados**  $\mathcal{S}$ .
- Un conjunto de **acciones**  $\mathcal{A}$ .
- Una **función de transición**  $\mathcal{P}$ .
- Una función de **recompensa**  $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$
- Un **factor de descuento**  $\gamma \in [0, 1]$ .

# Políticas

---

Una **política** es una función que refleja la probabilidad de emplear una determinada acción  $a \in \mathcal{A}(s)$  a partir de un estado  $s \in \mathcal{S}$ .

- La política rige el comportamiento del agente, representando su preferencia por unas acciones u otras ante diferentes estados.



Diferenciamos entre políticas **deterministas** y **estocásticas**.

# Tipos de políticas

## Política determinista

La probabilidad de tomar una acción es 0 ó 1:

$$a_t = \mu(s_t)$$

## Política estocástica

Distribución de probabilidades de todas las acciones posibles:

$$a_t \sim \pi(\cdot | s_t)$$

En ambos casos se cumple que:

$$\sum_{a \in \mathcal{A}(s)} \pi(s|a) = 1$$

# Tipos de políticas

## Política determinista

La probabilidad de tomar una acción es 0 ó 1:

$$a_t = \mu(s_t)$$

## Política estocástica

Distribución de probabilidades de todas las acciones posibles:

$$a_t \sim \pi(\cdot | s_t)$$

En ambos casos se cumple que:

$$\sum_{a \in \mathcal{A}(s)} \pi(s|a) = 1$$

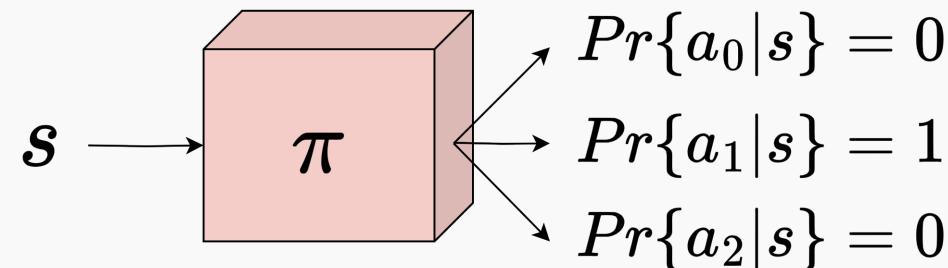
Por simplicidad, emplearemos indistintamente  $\pi$  para ambos tipos de políticas.

# Tipos de políticas

## Política determinista

La probabilidad de tomar una acción es 0 ó 1:

$$\pi(a | s) \in \{0, 1\}$$

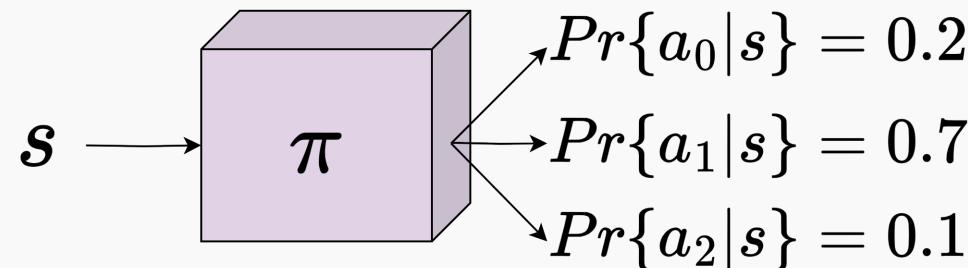


# Tipos de políticas

## Política estocástica

Distribución de probabilidades de todas las acciones posibles:

$$\pi(a | s) \in [0, 1]$$

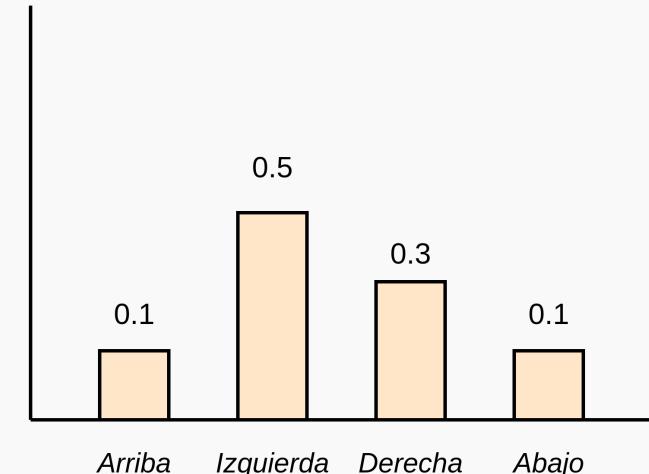


# Políticas estocásticas

## Política categórica

Selecciona acciones de una **distribución categórica**.

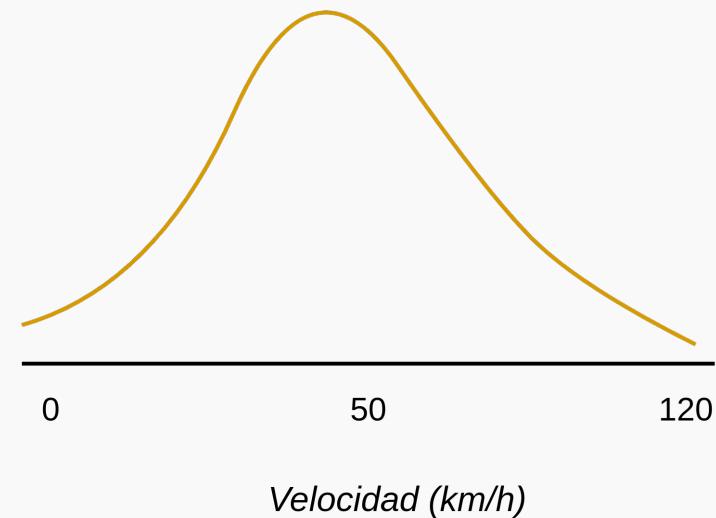
- Se emplea en **espacios de acciones discretos**.



## Política gaussiana

Muestrea acciones de una **distribución gaussiana**.

- Se emplea en **espacios de acciones continuos**.



## Pregunta...

*¿Alternar entre acciones (ej.  $a_0, a_1, a_2, a_0, a_1, a_2, \dots$ ) sería una política válida?*

# Pregunta...

*¿Alternar entre acciones (ej.  $a_0, a_1, a_2, a_0, a_1, a_2, \dots$ ) sería una política válida?*



**NO**, porque se incumple la propiedad de Markov.

# Aprendizaje de la política óptima

Nuestro objetivo es hacer que el agente aprenda una **política de comportamiento óptima** que le permita alcanzar sus objetivos ...

... y, por tanto, **maximizar la recompensa acumulada** (retorno).

Esto se traduce en asignar una mayor probabilidad a aquellas acciones que conduzcan a una mayor recompensa a largo plazo.

Para guiar al agente en el proceso de aprendizaje empleamos **funciones de valor**.

# Funciones de valor y optimalidad

---

# Funciones de valor

Utilizamos **funciones de valor** para evaluar la *calidad* de **estados** y **acciones**.

## FUNCIÓN ESTADO-VALOR

$$v_{\pi}(s) = \mathbb{E}[G_t | S_t = s]$$

*Retorno esperado al visitar el estado  $s$  y seguir una política  $\pi$ .*

## FUNCIÓN ACCIÓN-VALOR

$$q_{\pi}(s, a) = \mathbb{E}[G_t | S_t = s, A_t = a]$$

*Retorno esperado al realizar la acción  $a$  desde el estado  $s$  y seguir una política  $\pi$ .*

# Funciones de valor

Si desarrollamos estas fórmulas tenemos:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t | S_t = s] \\ &= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right] \end{aligned}$$

$$\begin{aligned} q_\pi(s, a) &= \mathbb{E}_\pi[G_t | S_t = s, A_t = a] \\ &= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\right] \end{aligned}$$

- Suma de las recompensas descontadas desde  $t$  en adelante.

# Pregunta...

*¿Cuál es la diferencia entre valor y recompensa?*

# Pregunta...

*¿Cuál es la diferencia entre **valor** y **recompensa**?*

- **Recompensa** → es una señal **inmediata** que el agente recibe después de realizar una acción o transicionar a un estado.
- **Valor** → es una estimación de las **recompensas** a obtener a largo plazo (*retorno*).

# Ecuaciones de Bellman

# Ecuación de Bellman para $v_\pi$

La ecuación de Bellman para  $v_\pi$  es la *definición recursiva* de la función estado-valor:

$$\begin{aligned}v_\pi(s) &= \mathbb{E}_\pi[G_t | S_t = s] \\&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s] \\&= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma v_\pi(s')]\end{aligned}$$

# Ecuación de Bellman para $v_\pi$

La ecuación de Bellman para  $v_\pi$  es la *definición recursiva* de la función estado-valor:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t | S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a) [r + \gamma v_\pi(s')] \end{aligned}$$

Probabilidad  
de elegir  
cada acción    Probabilidad  
de transición    Recompensa  
inmediata +  
futura  
descontada

Si el problema es determinista,  $\sum_{s',r} p(s',r|s,a)$  se elimina de la ecuación de Bellman.

## Ecuación de Bellman para $v_\pi$

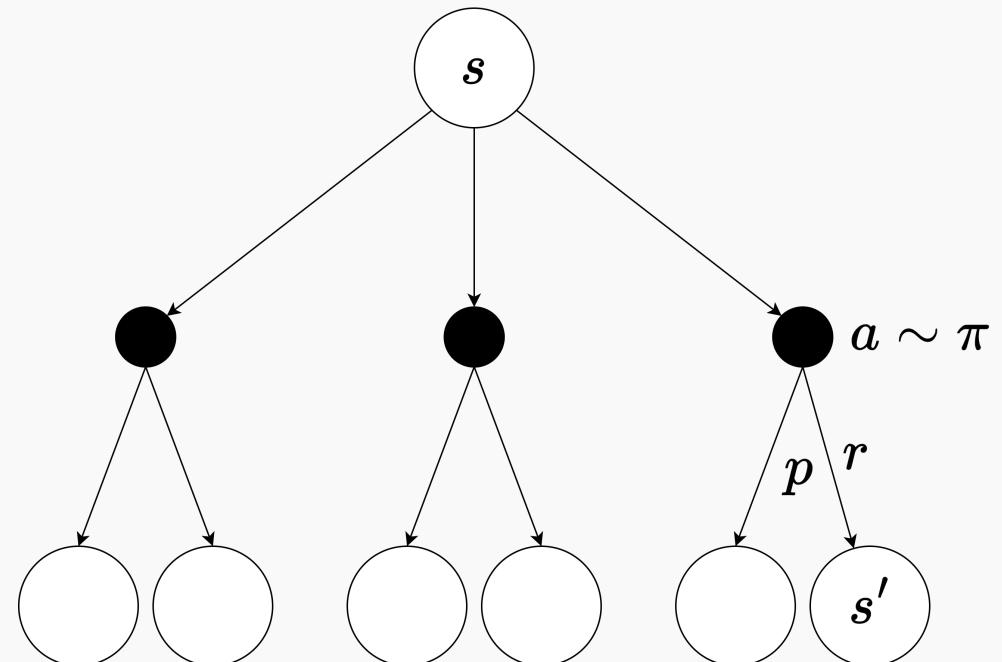
La **ecuación de Bellman** tiene en cuenta todas las probabilidades de transición, ponderando las **recompensas obtenibles** por su **probabilidad**.

$$\begin{aligned}v_{\pi(s)} &= \mathbb{E}_\pi[G_t | S_t = s] \\&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s] \\&= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a) [r + \gamma v_\pi(s')]\end{aligned}$$

# Diagrama *backup* para $v_\pi$

Los **diagramas *backup*** representan cómo se transfiere la información sobre los valores desde estados sucesores hasta el estado actual.

- Similar para pares acción-estado.



# Ecuación de Bellman para $q_{\pi}$

De forma análoga, esta es la **definición recursiva** de la **función acción-valor**:

$$\begin{aligned} q_{\pi}(s, a) &= \mathbb{E}_{\pi}[G_t | S_t = s, A_t = a] \\ &= \sum_{s', r} p(s', r | s, a) [r + \gamma \mathbb{E}_{\pi}[G_{t+1} | S_{t+1} = s']] \\ &= \sum_{s', r} p(s', r | s, a) \left[ r + \gamma \sum_{a'} \pi(a' | s') q_{\pi}(s', a') \right] \end{aligned}$$

# Ecuación de Bellman para $q_{\pi}$

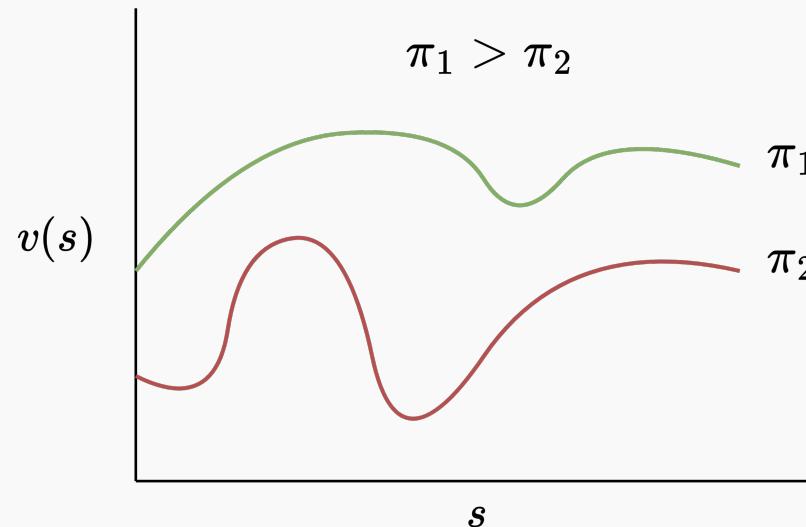
De forma análoga, esta es la **definición recursiva** de la **función acción-valor**:

$$\begin{aligned} q_{\pi}(s, a) &= \mathbb{E}_{\pi}[G_t | S_t = s, A_t = a] \\ &= \sum_{s', r} p(s', r | s, a) [r + \gamma \mathbb{E}_{\pi}[G_{t+1} | S_{t+1} = s']] \\ &= \sum_{s', r} p(s', r | s, a) \left[ \underbrace{\begin{array}{c} r \\ \text{Recompensa} \\ \text{inmediata} \end{array}}_{\substack{\text{Probabilidades de transición} \\ (\text{dependiente del entorno})}} + \underbrace{\gamma \sum_{a'} \pi(a' | s') q_{\pi}(s', a')}_{\text{Recompensa futura, ponderada por la prob. de cada acción}} \right] \end{aligned}$$

# Políticas y funciones de valor

Podemos comparar políticas y establecer un **orden** entre ellas empleando las funciones de valor:

$$\pi \geq \pi' \Leftrightarrow v_\pi(s) \geq v_{\pi'}(s), \quad \forall s \in \mathcal{S}$$



# Políticas óptimas

Siempre existirá, al menos, una política mejor que cualquier otra, denominada **política óptima**,  $\pi^*$ .

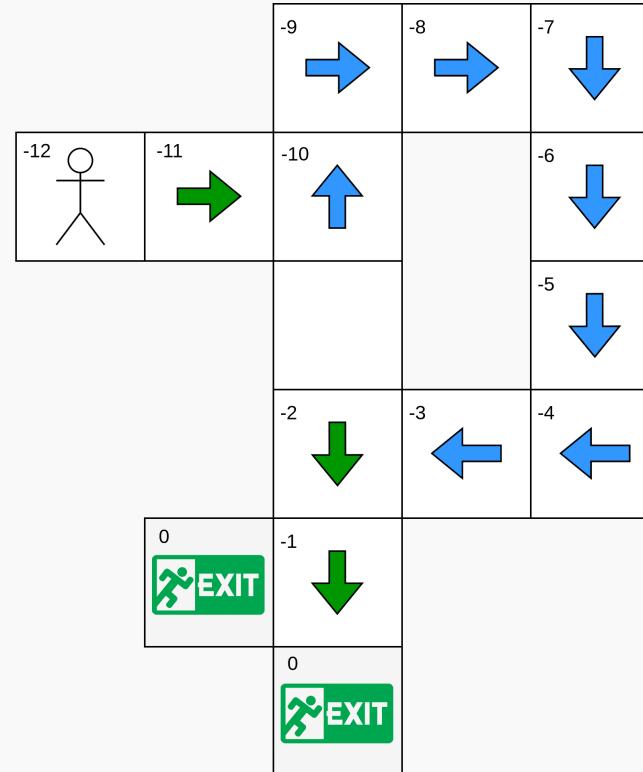
⚠ Puede haber más de una política óptima.

Las políticas óptimas comparten la misma **función estado-valor óptima**  $v^*$ :

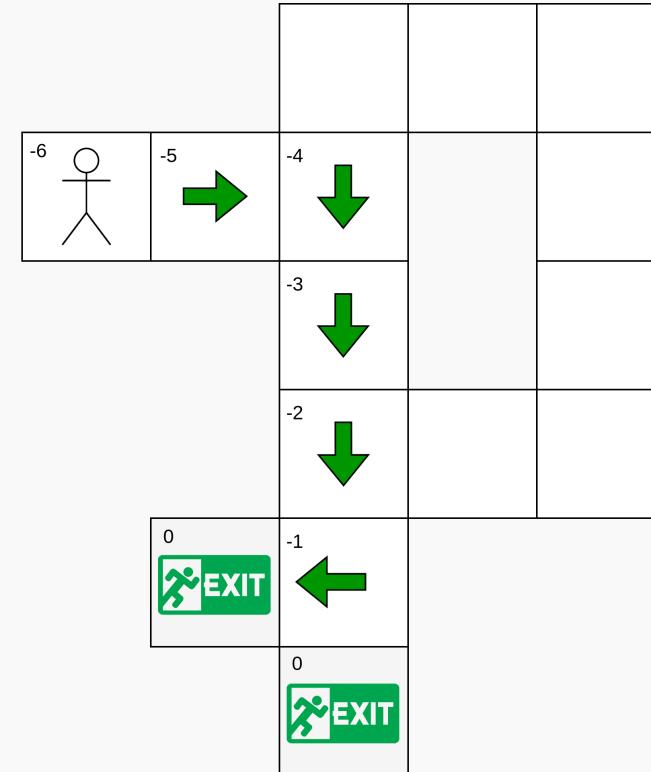
$$v^*(s) = \max_{\pi} v_{\pi}(s), \quad \forall s \in \mathcal{S}$$

- La función estado-valor óptima es la función estado-valor con el valor más alto entre todas las políticas.
- La función estado-valor óptima es **única**.

# Políticas óptimas



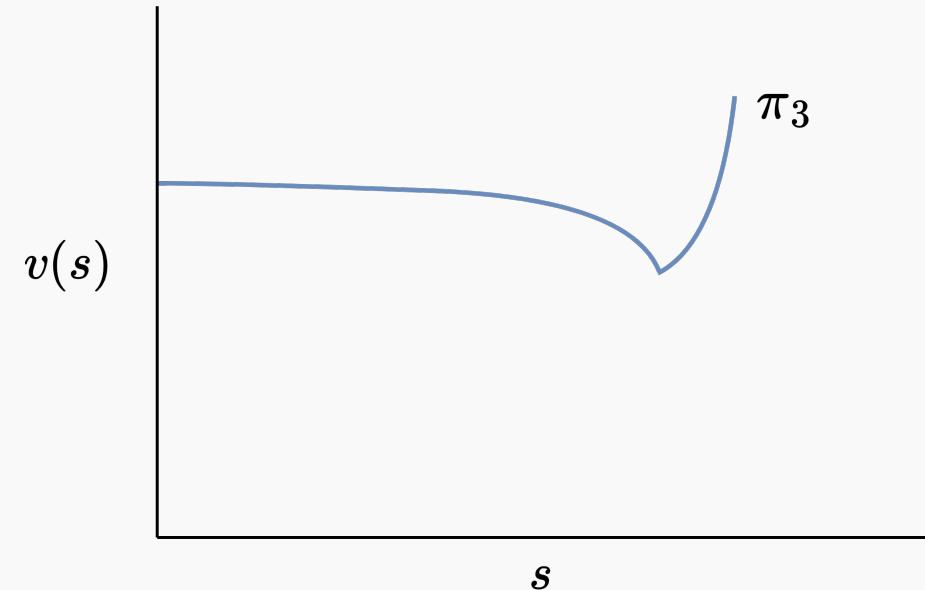
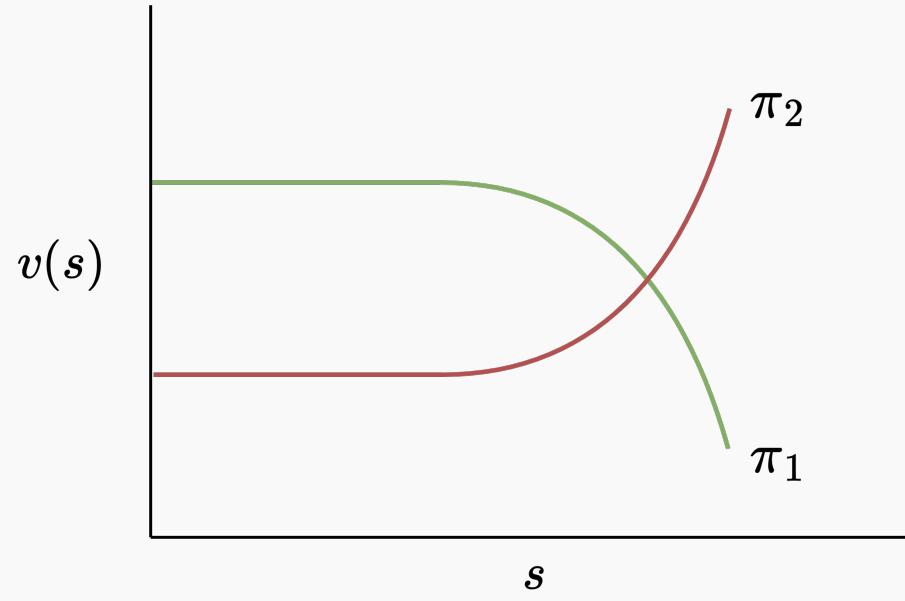
**Política subóptima** asociada a una función *estado-valor* que no es óptima para todos los estados.



Diferentes **políticas óptimas** asociadas a la misma función *estado-valor* óptima.

# Combinación de políticas

Podemos combinar políticas subóptimas para formar políticas mejores:



No es necesario hacer sacrificios en determinados estados tomando acciones subóptimas.

# Función acción-valor óptima

Las políticas óptimas también comparten la misma **función acción-valor óptima**:

$$q^*(s, a) = \max_{\pi} q_{\pi}(s, a), \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$$

También podemos definir  $q^*$  en términos de  $v^*$  tal que:

$$q^*(s, a) = \mathbb{E}[R_{t+1} + \gamma v^*(S_{t+1}) | S_t = s, A_t = a]$$

$q^*$  asocia a cada par acción-estado una recompensa esperada igual a la recompensa inmediata + recompensa (descontada) futura de acuerdo a la función de valor óptima  $v^*$ .

# Ecuaciones de optimalidad de Bellman

# Ecuación de optimalidad de Bellman para $v^*$

La **función estado-valor óptima**  $v^*$  puede definirse tal que:

$$\begin{aligned}v^*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi^*}(s, a) \\&= \max_a \mathbb{E}[R_{t+1} + \gamma v^*(S_{t+1}) \mid S_t = s, A_t = a] \\&= \max_a \sum_{s',r} p(s', r | s, a)[r + \gamma v^*(s')]\end{aligned}$$

*El valor óptimo de un estado será aquel asociado a seguir una acción óptima desde este en adelante.*

# Ecuación de optimalidad de Bellman para $v^*$

La **función estado-valor óptima**  $v^*$  puede definirse tal que:

$$\begin{aligned} v^*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi^*}(s, a) \\ &= \max_a \mathbb{E}[R_{t+1} + \gamma v^*(S_{t+1}) \mid S_t = s, A_t = a] \quad (\text{Eq. 1}) \end{aligned}$$

$$= \max_a \sum_{s',r} p(s', r | s, a)[r + \gamma v^*(s')] \quad (\text{Eq. 2})$$

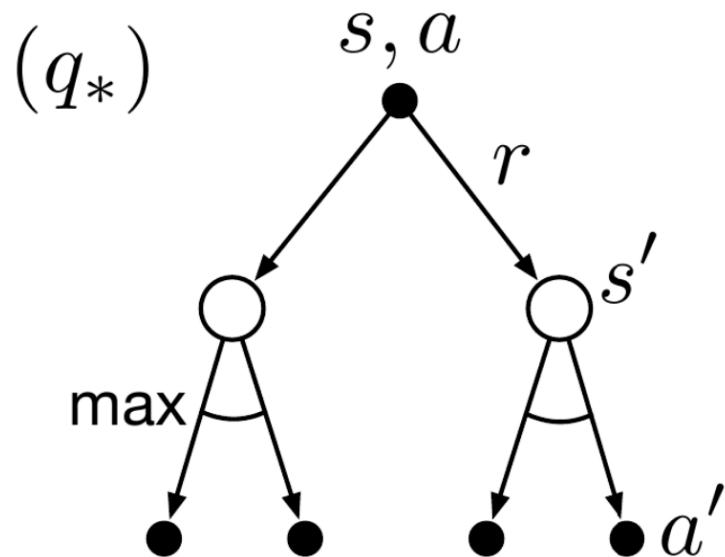
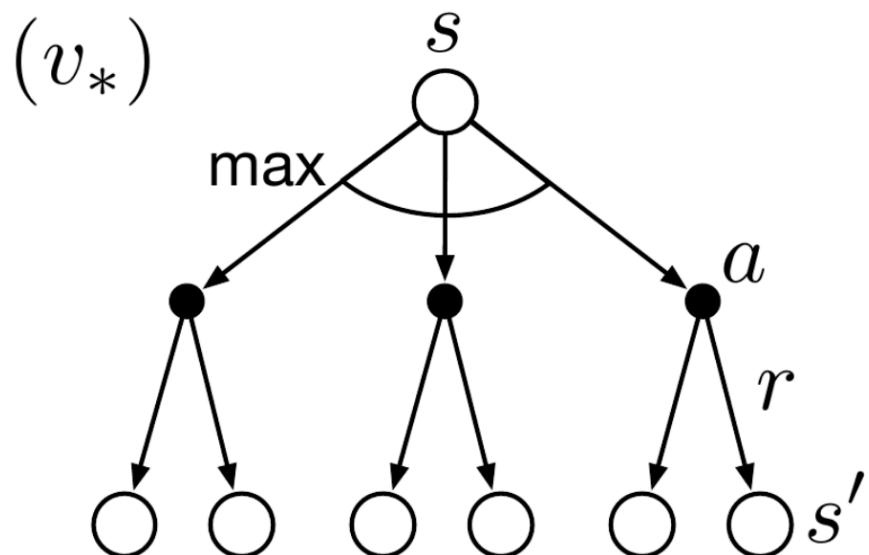
*El valor óptimo de un estado será aquel asociado a seguir una acción óptima desde este en adelante.*

# Ecuación de optimalidad de Bellman para $v^*$

La **función de acción-valor óptima**  $q^*$  puede definirse tal que:

$$\begin{aligned} q^*(s, a) &= \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q^*(S_{t+1}, a') \mid S_t = s, A_t = a\right] \\ &= \sum_{s', r} p(s', r | s, a) \left[ r + \gamma \max_{a'} q^*(s', a') \right] \end{aligned}$$

# Diagramas backup para ecuaciones de optimalidad



# Obtención de la política óptima

Conociendo  $v^*$  podemos extraer fácilmente la política óptima, ya que cualquier política que actúa de forma *greedy* con respecto a  $v^*$  es óptima:

1. Examinar el valor de los sucesores de  $s$ .
2. Transición al  $s'$  de mayor valor (asumiendo que  $v^*$  es óptima).

Conocer  $q^*$  hace que el proceso de obtención de la política óptima sea incluso más sencillo:

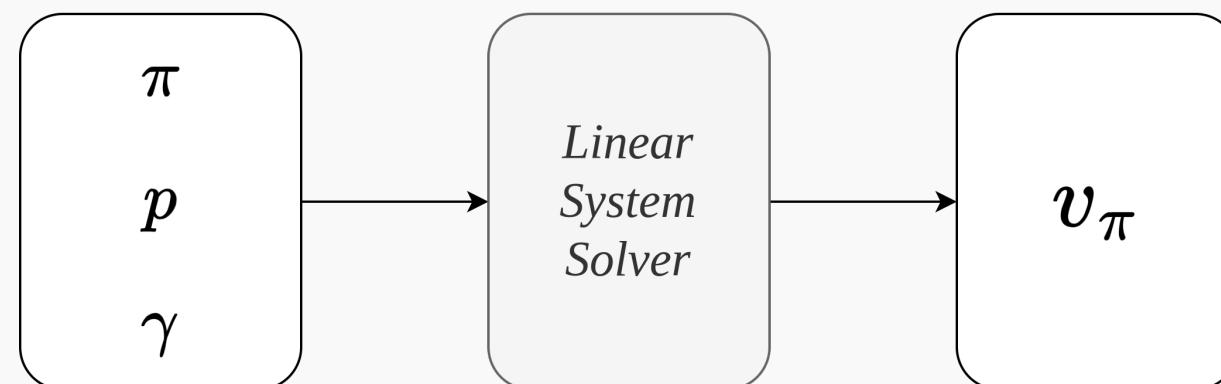
1. Para cualquier estado  $s$ , se emplea la acción que maximice  $q^*(s, a)$ .

# Resolución de las ecuaciones de optimalidad de Bellman

En MDPs finitos, las ecuaciones de optimalidad de Bellman tienen **soluciones únicas**.

- Definimos una ecuación por estado.
- Dados ***n* estados**, tenemos ***n* ecuaciones lineales** y, por tanto, ***n* incógnitas**.

La resolución del sistema de ecuaciones nos permite obtener la **política óptima**.



# Resolución de las ecuaciones de optimalidad de Bellman

No obstante, esto implica dar por supuestos tres aspectos fundamentales que rara vez se dan en la práctica:

1. Conocimiento preciso/completo de las **dinámicas del entorno**.
2. **Recursos computacionales** suficientes para calcular la solución.
3. El cumplimiento de la **propiedad de Markov**.

Esto motiva el uso de métodos alternativos basados en la **aproximación** de la solución.

# Resolución de las ecuaciones de optimalidad de Bellman

A pesar de que se cumpliesen las condiciones:

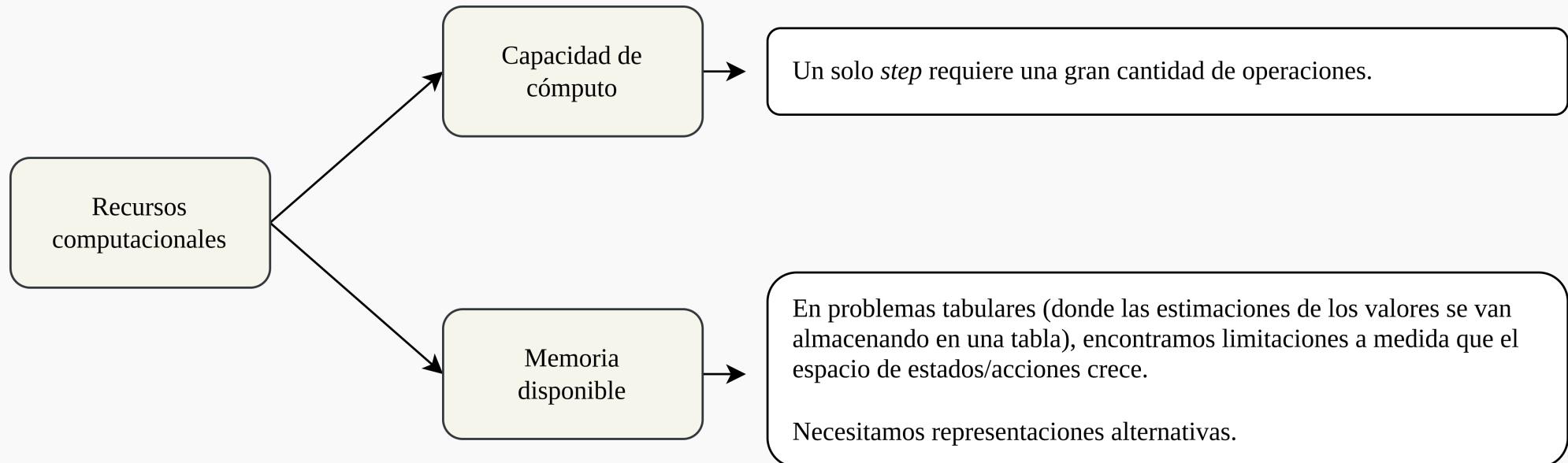
1. Conocimiento preciso/completo de las dinámicas del entorno.
3. El cumplimiento de la propiedad de Markov.

normalmente no es posible lidiar con:

2. Recursos computacionales suficientes para calcular la solución.

especialmente en problemas complejos con un gran número de estados/acciones.

# Resolución de las ecuaciones de optimidad de Bellman



Sería deseable que el agente sacrificase cierta precisión en situaciones poco frecuentes, y la mejorase en aquellos casos más comunes.

# Trabajo propuesto

---

# Trabajo propuesto

- Leer sobre otros tipos de **MDPs / POMDPs** y conocer sus diferencias.
- Familiarización con la API de **Gymnasium**:
  - <https://gymnasium.farama.org/>
  - Implementación de un **agente aleatorio** sobre un entorno de ejemplo.
  - Implementación de un **agente basado en reglas** sobre el mismo entorno.
- **Resolver** un MDP sencillo mediante un **sistema de ecuaciones** lineal.
- **Función de ventaja** (*advantage function*):
  - ¿Qué es?
  - ¿En qué se diferencia de las funciones de valor estudiadas?

## Bibliografía y vídeos:

- <https://youtu.be/lfHX2hHRMVQ?si=2jR4HI72ReErh7rb>
- [https://web.stanford.edu/class/cme241/lecture\\_slides/david\\_silver\\_slides/MDP.pdf](https://web.stanford.edu/class/cme241/lecture_slides/david_silver_slides/MDP.pdf)

# Procesos de decisión de Markov

Procesos de decisión de Markov

---

Antonio Manjavacas Lucas

[manjavacas@ugr.es](mailto:manjavacas@ugr.es)