Aprendizaje por refuerzo-Introducción

Fernando Lozano

Universidad de los Andes

24 de enero de 2023

Aprendizaje por refuerzo

Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.

Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.
- Agente busca lograr una meta, a través de una serie de acciones.

Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.
- Agente busca lograr una meta, a través de una serie de acciones.
- No existe un maestro o supervisor.

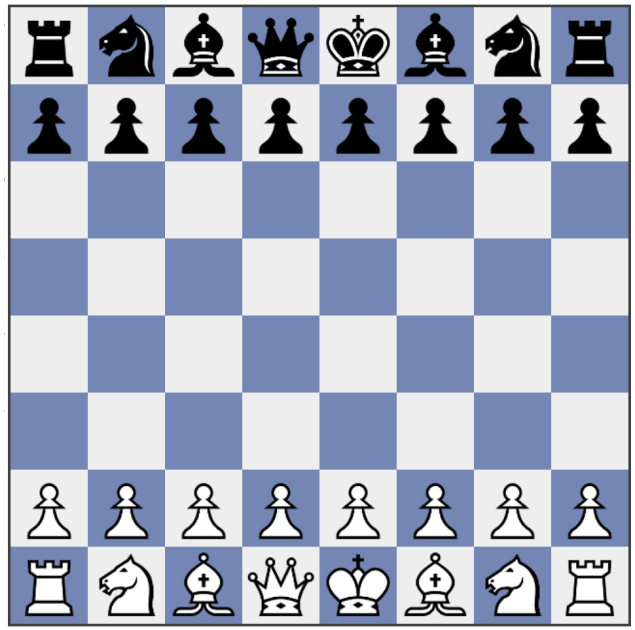
Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.
- Agente busca lograr una meta, a través de una serie de acciones.
- No existe un maestro o supervisor.
- Agente debe aprender de su experiencia.

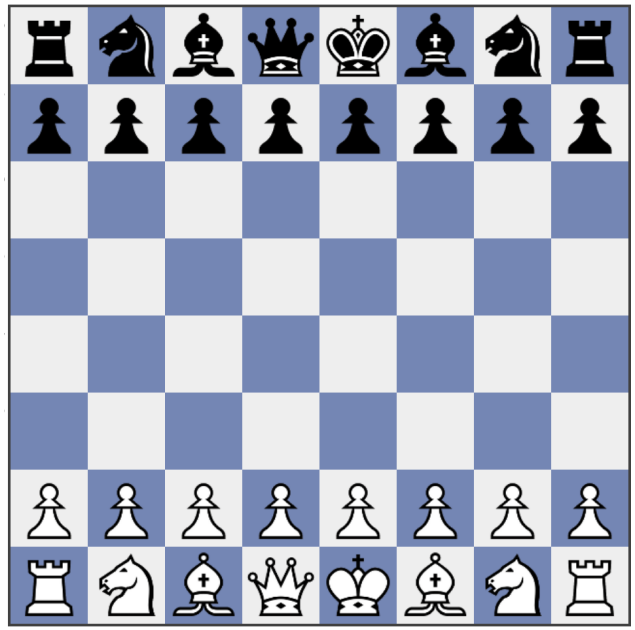
Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.
- Agente busca lograr una meta, a través de una serie de acciones.
- No existe un maestro o supervisor.
- Agente debe aprender de su experiencia.

Ejemplo

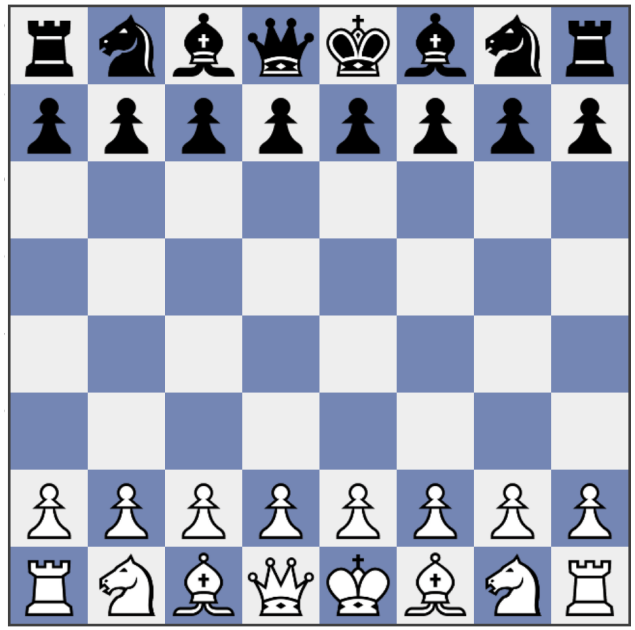


Ejemplo



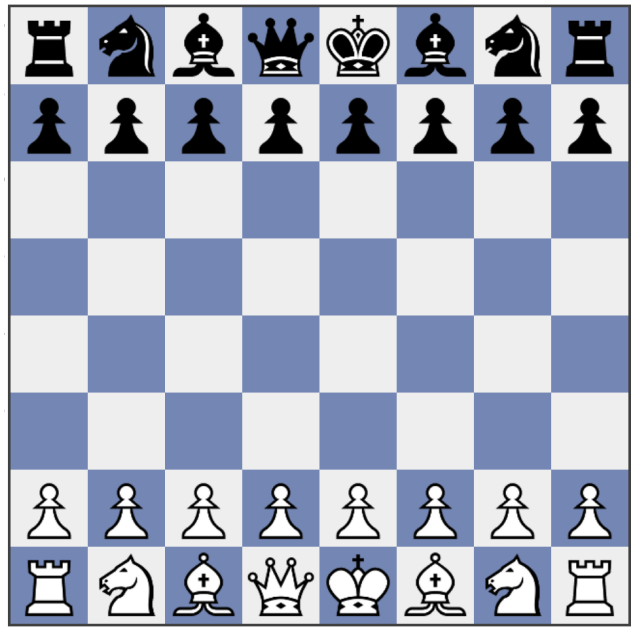
- Agente que aprende a jugar ajedrez.

Ejemplo



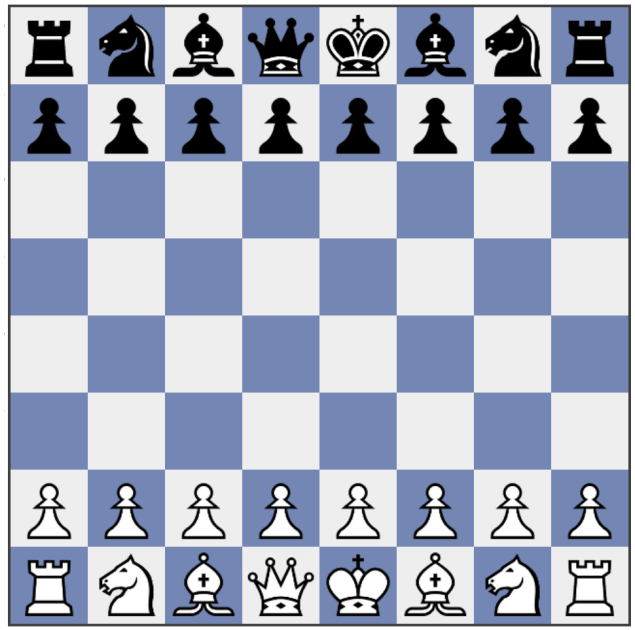
- Agente que aprende a jugar ajedrez.
- Ambiente:

Ejemplo



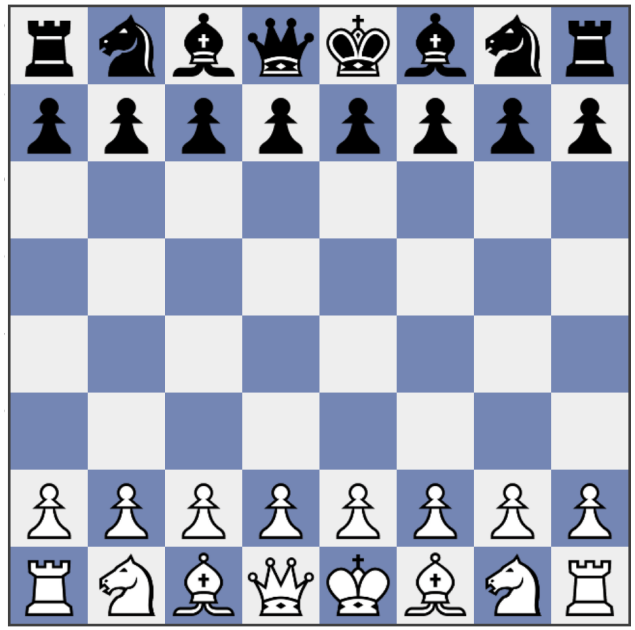
- Agente que aprende a jugar ajedrez.
- Ambiente: Tablero

Ejemplo



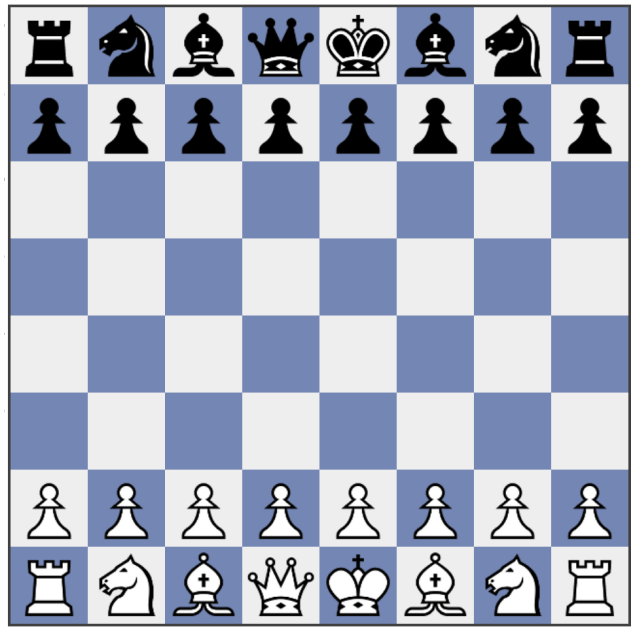
- Agente que aprende a jugar ajedrez.
- Ambiente: Tablero, oponente.

Ejemplo



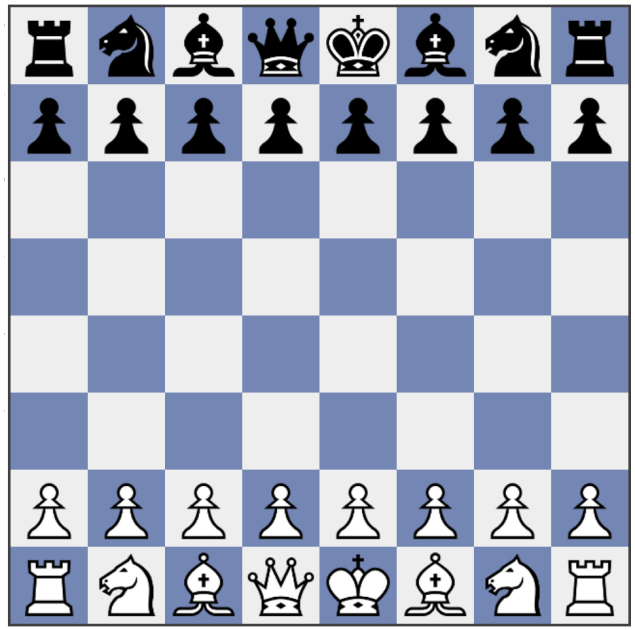
- Agente que aprende a jugar ajedrez.
- Ambiente: Tablero, oponente.
- Acciones:

Ejemplo



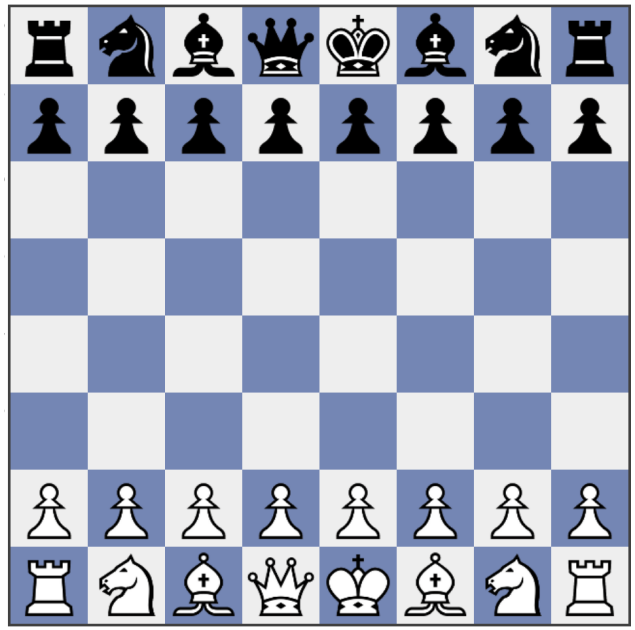
- Agente que aprende a jugar ajedrez.
- Ambiente: Tablero, oponente.
- Acciones: Jugadas válidas en cada configuración posible del tablero.
- Meta:

Ejemplo



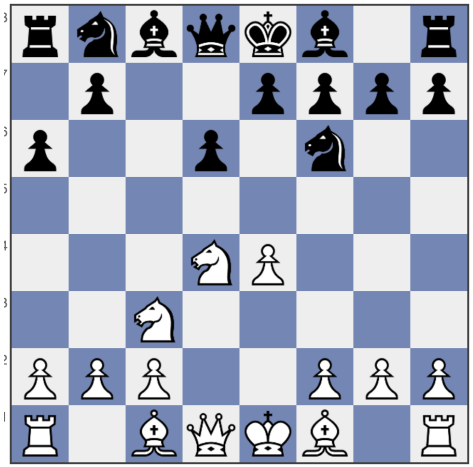
- Agente que aprende a jugar ajedrez.
- Ambiente: Tablero, oponente.
- Acciones: Jugadas válidas en cada configuración posible del tablero.
- Meta: Aprender política que tenga alta probabilidad de ganar.

Ejemplo



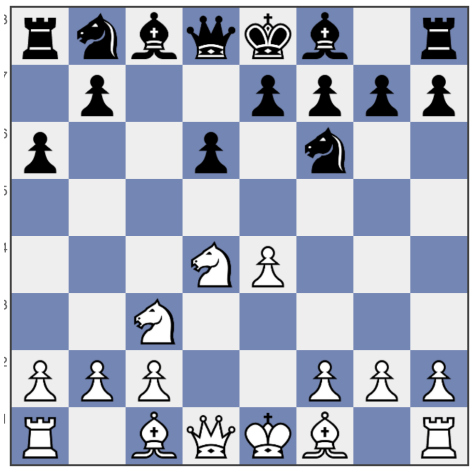
- Agente que aprende a jugar ajedrez.
- Ambiente: Tablero, oponente.
- Acciones: Jugadas válidas en cada configuración posible del tablero.
- Meta: Aprender política que tenga alta probabilidad de ganar.
  - Jugada a realizar en cada configuración posible del tablero.

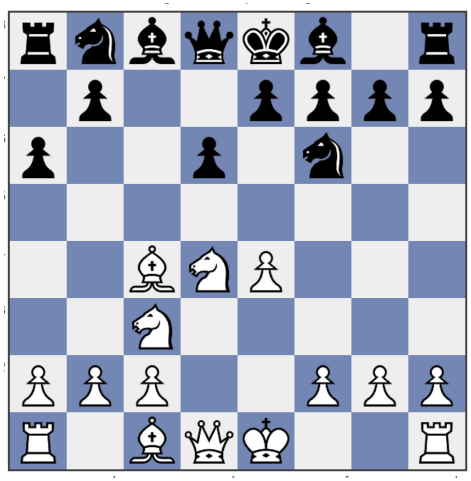
S_t



S_{t+1}

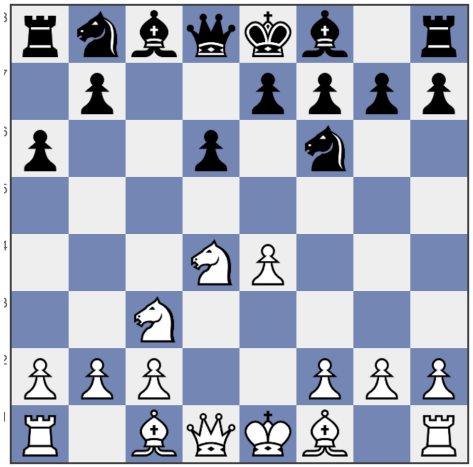
S_t

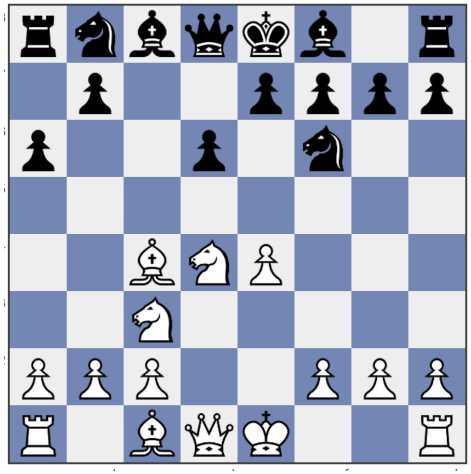




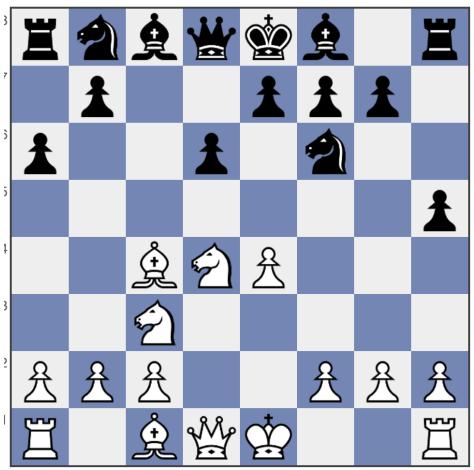
S_{t+1}

S_t

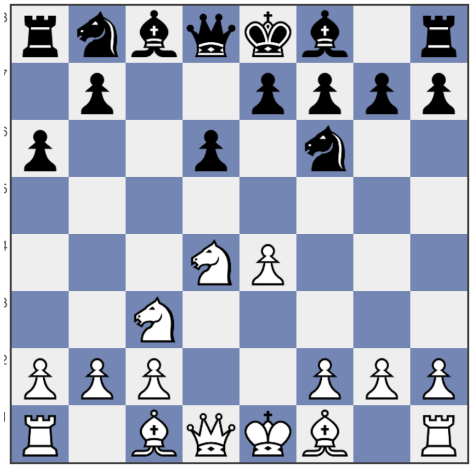


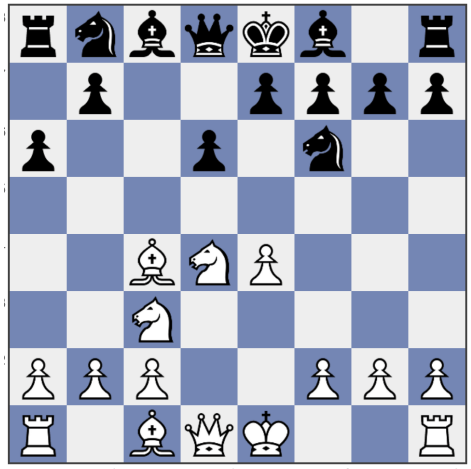


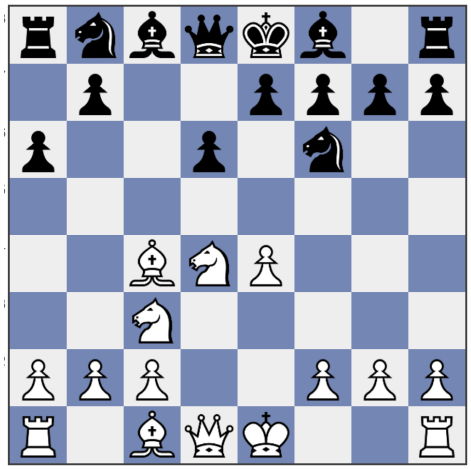
S_{t+1}



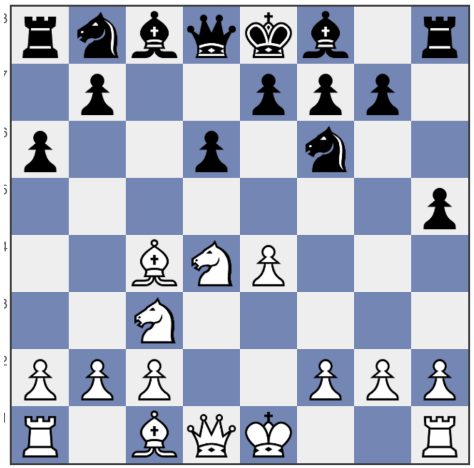
S_t



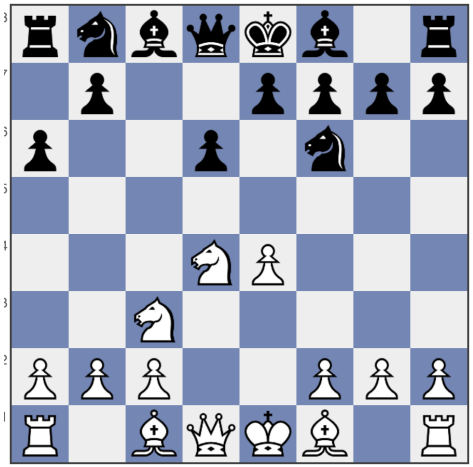




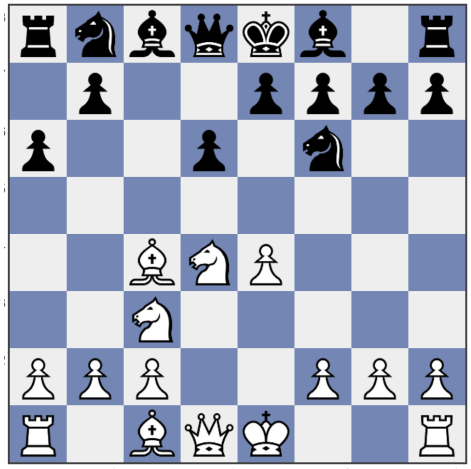
S_{t+1}

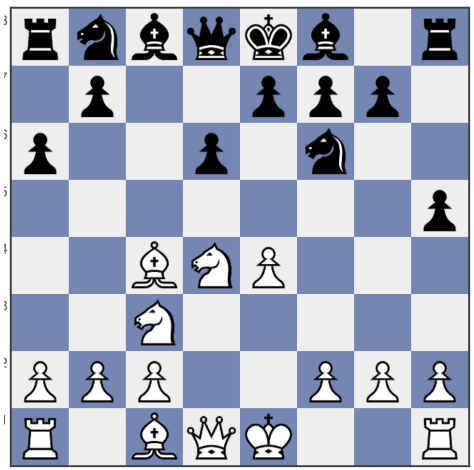


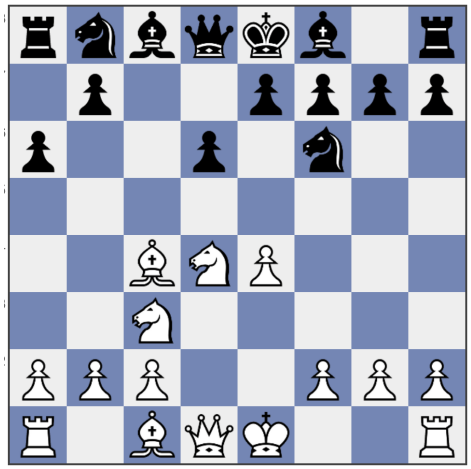
S_t

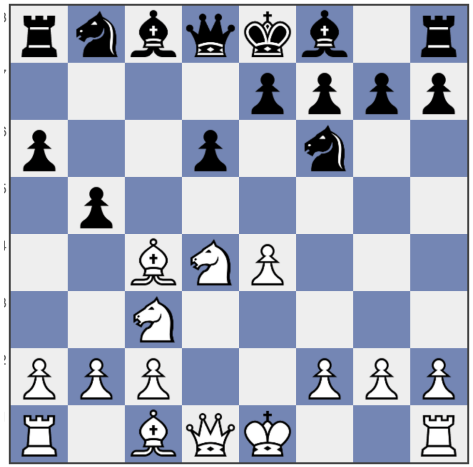


S_{t+1}

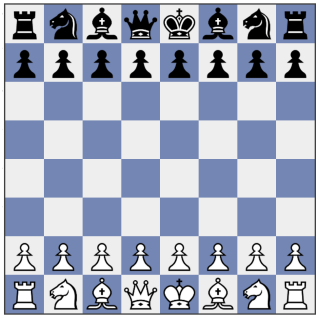




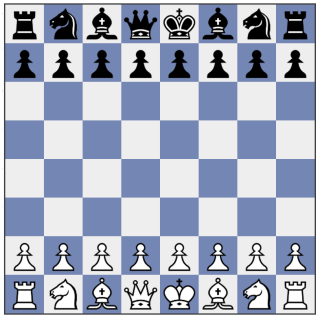




S_0 $\cdots$ S_t S_{t+1} $\cdots$ S_T

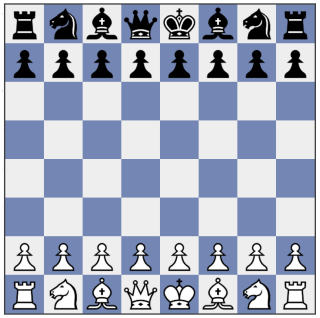


S_0 $\cdots$ S_t S_{t+1} $\cdots$ S_T

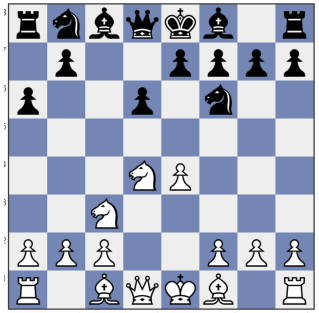


$\cdots$

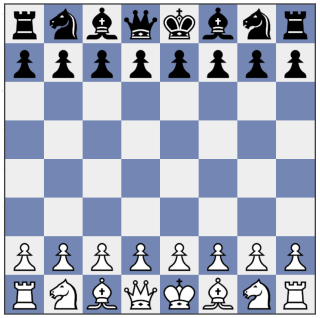
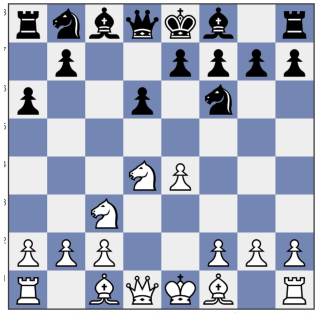
S_0 $\cdots$ S_t S_{t+1} $\cdots$ S_T



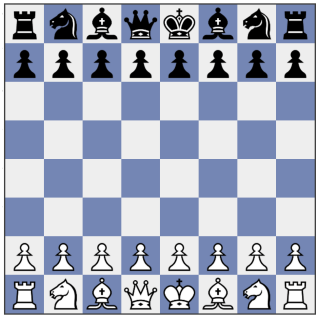
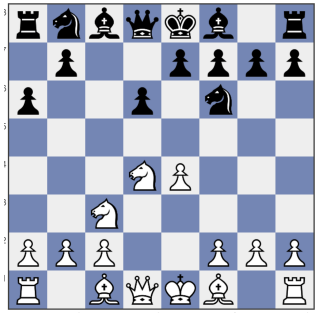
$\cdots$



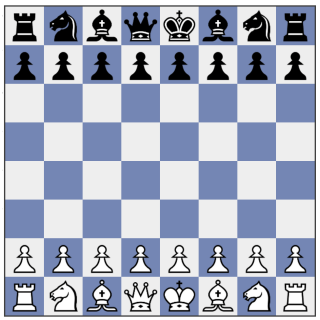
S_0 $\cdots$ S_t S_{t+1} $\cdots$ S_T

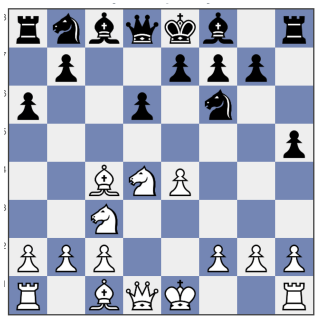
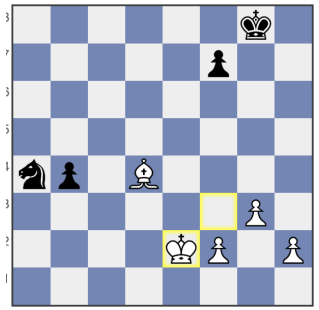




S_0 $\cdots$ S_t S_{t+1} $\cdots$ S_T

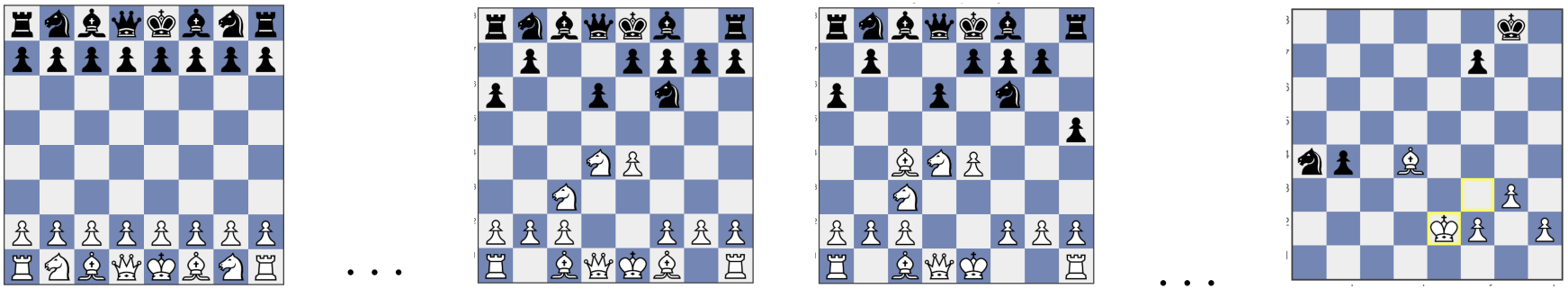




S_0 $\cdots$ S_t S_{t+1} $\cdots$ S_T



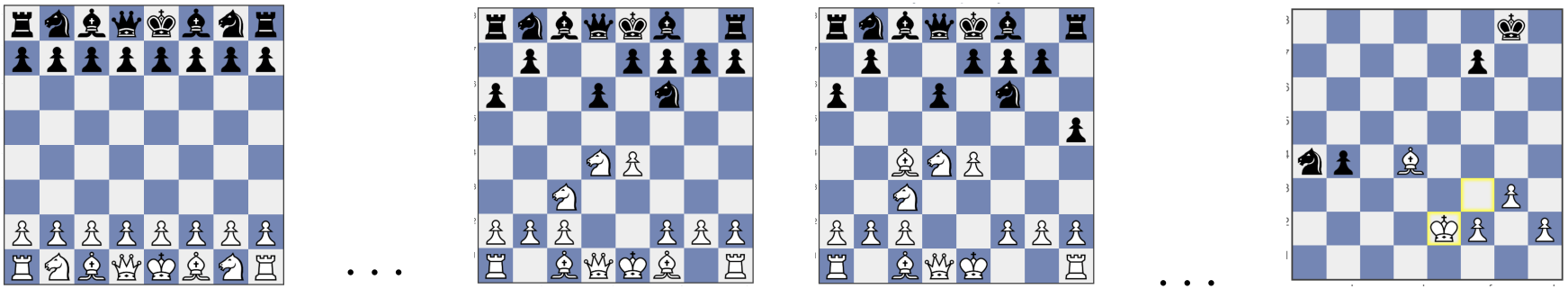



S_0 $\cdots$ S_t S_{t+1} $\cdots$ S_T



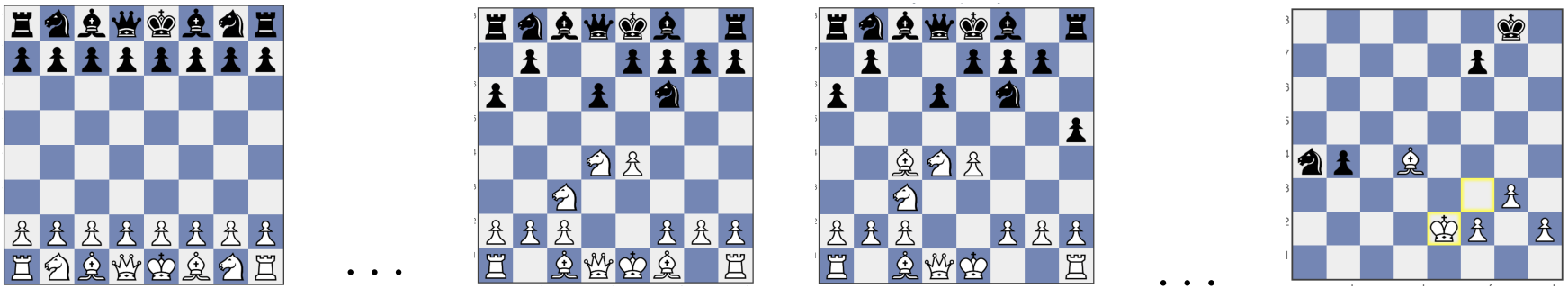
- S_T es estado terminal, se gana o pierde la partida.

S_0 $\cdots$ S_t S_{t+1} $\cdots$ S_T



- S_T es estado terminal, se gana o pierde la partida.
- Problema de asignación de crédito:

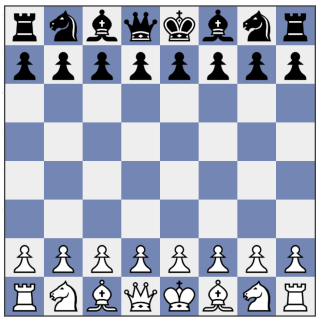
S_0 $\cdots$ S_t S_{t+1} $\cdots$ S_T



- S_T es estado terminal, se gana o pierde la partida.
- Problema de asignación de crédito:
  - Qué jugadas influyeron más/menos para ganar o perder?

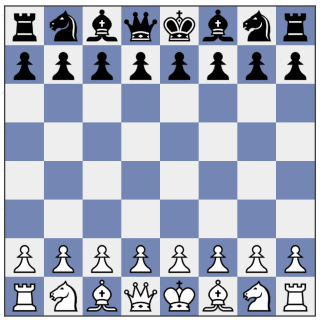
Función de valor

$$V(S_0) \quad \cdots \quad V(S_t) \quad V(S_{t+1}) \quad \cdots \quad V(S_T)$$



Función de valor

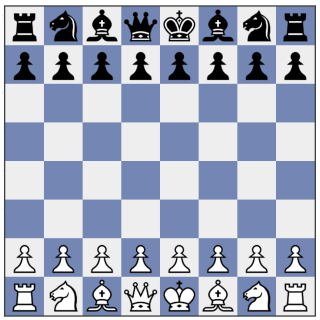
$$V(S_0) \quad \cdots \quad V(S_t) \quad V(S_{t+1}) \quad \cdots \quad V(S_T)$$



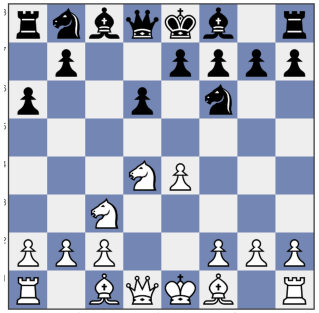
$\cdots$

Función de valor

$V(S_0)$ $\cdots$ $V(S_t)$ $V(S_{t+1})$ $\cdots$ $V(S_T)$

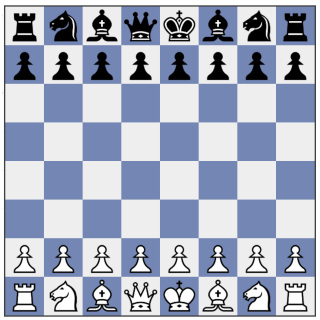


$\cdots$

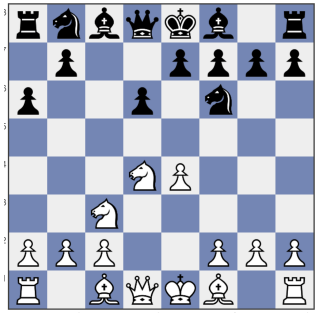


Función de valor

$V(S_0)$ $\cdots$ $V(S_t)$ $V(S_{t+1})$ $\cdots$ $V(S_T)$

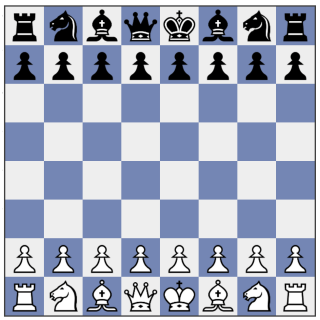
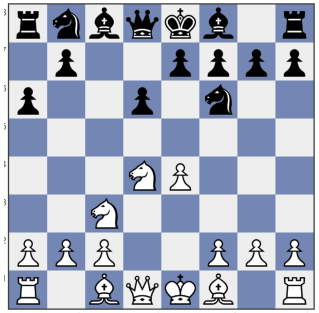


$\cdots$



Función de valor

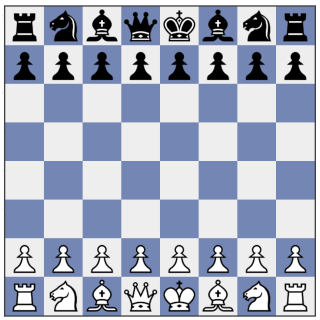
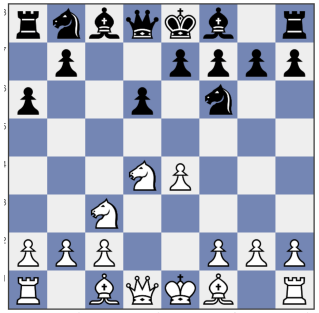
$V(S_0)$ $\cdots$ $V(S_t)$ $V(S_{t+1})$ $\cdots$ $V(S_T)$

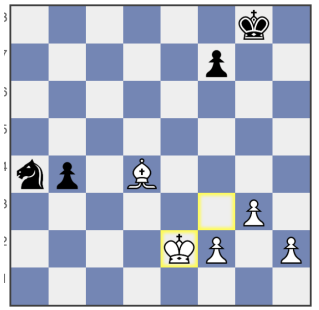




$\cdots$ $\cdots$

Función de valor

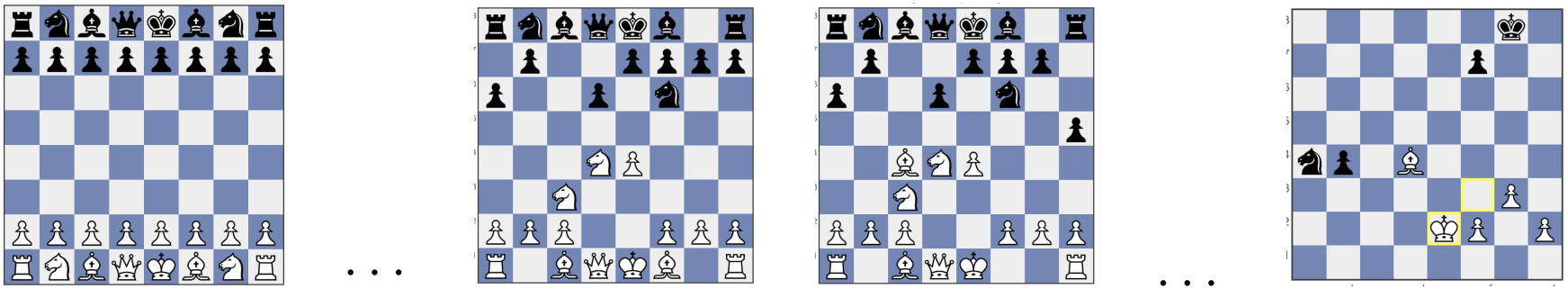
$V(S_0)$ $\cdots$ $V(S_t)$ $V(S_{t+1})$ $\cdots$ $V(S_T)$



Función de valor

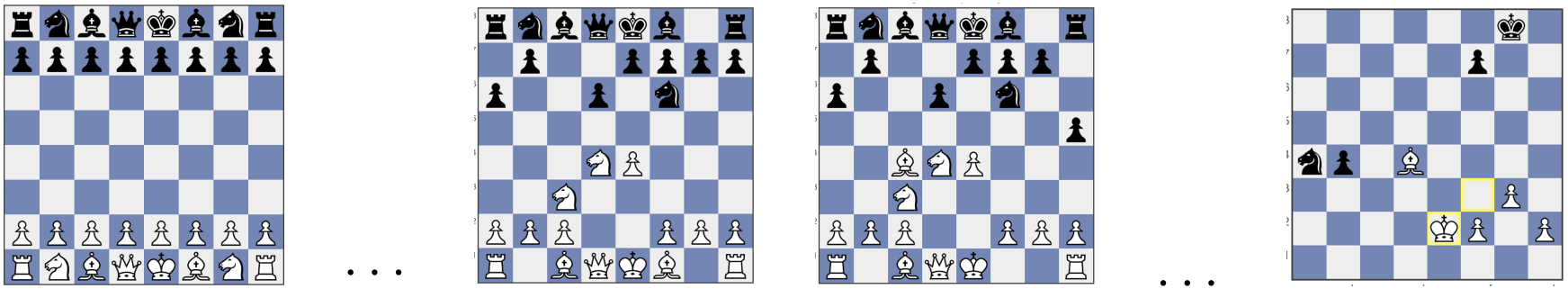
$V(S_0)$ $\cdots$ $V(S_t)$ $V(S_{t+1})$ $\cdots$ $V(S_T)$



- $V(S_t)$ dice que tan valioso es este estado con respecto a la meta que se quiere lograr.

Función de valor

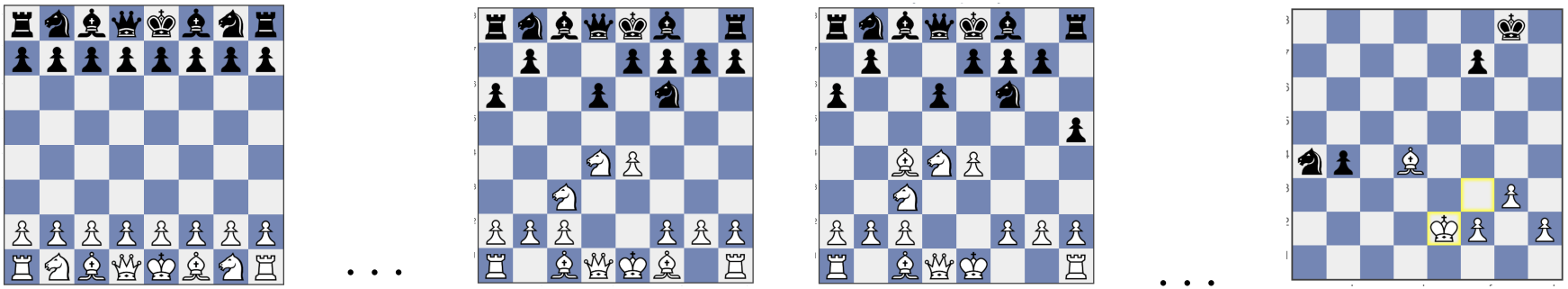
$V(S_0)$ $\cdots$ $V(S_t)$ $V(S_{t+1})$ $\cdots$ $V(S_T)$



- $V(S_t)$ dice que tan valioso es este estado con respecto a la meta que se quiere lograr.
- Estimar valores de $V(S_t)$ a partir de la experiencia adquirida al interactuar con el ambiente.

Función de valor

$V(S_0)$ $\cdots$ $V(S_t)$ $V(S_{t+1})$ $\cdots$ $V(S_T)$



- $V(S_t)$ dice que tan valioso es este estado con respecto a la meta que se quiere lograr.
- Estimar valores de $V(S_t)$ a partir de la experiencia adquirida al interactuar con el ambiente.
- Una posibilidad:

$$V(S_t) \leftarrow V(S_t) + \alpha \left[V(S_{t+1}) - V(S_t) \right]$$

- Dilema exploración/explotación:

- Dilema exploración/explotación:
  - Explotación: Escoger jugadas que llevan a estados con estimativos de V más grandes:

- Dilema exploración/explotación:
  - Explotación: Escoger jugadas que llevan a estados con estimativos de V más grandes: puede ignorar mejores jugadas.

- Dilema exploración/explotación:
  - Explotación: Escoger jugadas que llevan a estados con estimativos de V más grandes: puede ignorar mejores jugadas.
  - Exploración: Ensayar jugadas diferentes:

- Dilema exploración/explotación:
  - Explotación: Escoger jugadas que llevan a estados con estimativos de V más grandes: puede ignorar mejores jugadas.
  - Exploración: Ensayar jugadas diferentes: No adquiere experiencia suficiente sobre los estados.

- Dilema exploración/explotación:
  - Explotación: Escoger jugadas que llevan a estados con estimativos de V más grandes: puede ignorar mejores jugadas.
  - Exploración: Ensayar jugadas diferentes: No adquiere experiencia suficiente sobre los estados.

- Dilema exploración/explotación:
  - Explotación: Escoger jugadas que llevan a estados con estimativos de V más grandes: puede ignorar mejores jugadas.
  - Exploración: Ensayar jugadas diferentes: No adquiere experiencia suficiente sobre los estados.
- Aproximación de funciones (e.g. redes neuronales).

Ejemplos

Ejemplos

- Programa que aprende a jugar juego de tablero o videojuego.

Ejemplos

- Programa que aprende a jugar juego de tablero o videojuego.
- Robot que aprende a navegar en un ambiente.

Ejemplos

- Programa que aprende a jugar juego de tablero o videojuego.
- Robot que aprende a navegar en un ambiente.
- Control de ascensores en el ML.

Ejemplos

- Programa que aprende a jugar juego de tablero o videojuego.
- Robot que aprende a navegar en un ambiente.
- Control de ascensores en el ML.
- Control de semáforos para optimizar tiempos de viaje.

Ejemplos

- Programa que aprende a jugar juego de tablero o videojuego.
- Robot que aprende a navegar en un ambiente.
- Control de ascensores en el ML.
- Control de semáforos para optimizar tiempos de viaje.
- Gestión de energía en microredes.

Ejemplos

- Programa que aprende a jugar juego de tablero o videojuego.
- Robot que aprende a navegar en un ambiente.
- Control de ascensores en el ML.
- Control de semáforos para optimizar tiempos de viaje.
- Gestión de energía en microredes.
- Programa que aprende a armar rompecabezas.

Ejemplos

- Programa que aprende a jugar juego de tablero o videojuego.
- Robot que aprende a navegar en un ambiente.
- Control de ascensores en el ML.
- Control de semáforos para optimizar tiempos de viaje.
- Gestión de energía en microredes.
- Programa que aprende a armar rompecabezas.
- Control óptimo.

Ejemplos

- Programa que aprende a jugar juego de tablero o videojuego.
- Robot que aprende a navegar en un ambiente.
- Control de ascensores en el ML.
- Control de semáforos para optimizar tiempos de viaje.
- Gestión de energía en microredes.
- Programa que aprende a armar rompecabezas.
- Control óptimo.
- Banco central dirige economía hacia una meta específica.

Ejemplos

- Programa que aprende a jugar juego de tablero o videojuego.
- Robot que aprende a navegar en un ambiente.
- Control de ascensores en el ML.
- Control de semáforos para optimizar tiempos de viaje.
- Gestión de energía en microredes.
- Programa que aprende a armar rompecabezas.
- Control óptimo.
- Banco central dirige economía hacia una meta específica.

Aprendizaje por refuerzo

Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.

Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.
- Agente busca lograr una meta, a través de una serie de acciones.

Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.
- Agente busca lograr una meta, a través de una serie de acciones.
- No existe un maestro o supervisor.

Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.
- Agente busca lograr una meta, a través de una serie de acciones.
- No existe un maestro o supervisor.
- Agente debe aprender de su experiencia.

Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.
- Agente busca lograr una meta, a través de una serie de acciones.
- No existe un maestro o supervisor.
- Agente debe aprender de su experiencia.
- Debe lidiar con incertidumbre con respecto al ambiente.

Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.
- Agente busca lograr una meta, a través de una serie de acciones.
- No existe un maestro o supervisor.
- Agente debe aprender de su experiencia.
- Debe lidiar con incertidumbre con respecto al ambiente.
- Acciones correctas dependen de resultados de acciones pasadas:

Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.
- Agente busca lograr una meta, a través de una serie de acciones.
- No existe un maestro o supervisor.
- Agente debe aprender de su experiencia.
- Debe lidiar con incertidumbre con respecto al ambiente.
- Acciones correctas dependen de resultados de acciones pasadas: planeación.

Aprendizaje por refuerzo

- Aprendizaje de un agente que interactúa con el ambiente en el que está inmerso.
- Agente busca lograr una meta, a través de una serie de acciones.
- No existe un maestro o supervisor.
- Agente debe aprender de su experiencia.
- Debe lidiar con incertidumbre con respecto al ambiente.
- Acciones correctas dependen de resultados de acciones pasadas: planeación.
- Efectos de acciones en el medio ambiente no pueden ser predichas por completo: Agente debe monitorear el ambiente.

Elementos de un problema de aprendizaje por refuerzo

Elementos de un problema de aprendizaje por refuerzo

- Agente.

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente$\rightarrow$ representación del estado del ambiente.

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente$\rightarrow$ representación del estado del ambiente.
- Política (policy):

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente→ representación del estado del ambiente.
- Política (policy): Determina qué debe hacer el agente en una situación dada.

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente→ representación del estado del ambiente.
- Política (policy): Determina qué debe hacer el agente en una situación dada.
  - Mapeo de percepción del ambiente a acciones.

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente→ representación del estado del ambiente.
- Política (policy): Determina qué debe hacer el agente en una situación dada.
  - Mapeo de percepción del ambiente a acciones.
  - Reglas de asociación estímulo-respuesta.

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente→ representación del estado del ambiente.
- Política (policy): Determina qué debe hacer el agente en una situación dada.
  - Mapeo de percepción del ambiente a acciones.
  - Reglas de asociación estímulo-respuesta.
  - Aprendizaje:

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente→ representación del estado del ambiente.
- Política (policy): Determina qué debe hacer el agente en una situación dada.
  - Mapeo de percepción del ambiente a acciones.
  - Reglas de asociación estímulo-respuesta.
  - Aprendizaje: Identificar buenas políticas.

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente→ representación del estado del ambiente.
- Política (policy): Determina qué debe hacer el agente en una situación dada.
  - Mapeo de percepción del ambiente a acciones.
  - Reglas de asociación estímulo-respuesta.
  - Aprendizaje: Identificar buenas políticas.
- Función de recompensas:

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente→ representación del estado del ambiente.
- Política (policy): Determina qué debe hacer el agente en una situación dada.
  - Mapeo de percepción del ambiente a acciones.
  - Reglas de asociación estímulo-respuesta.
  - Aprendizaje: Identificar buenas políticas.
- Función de recompensas:
  - Define la meta del problema de aprendizaje.

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente→ representación del estado del ambiente.
- Política (policy): Determina qué debe hacer el agente en una situación dada.
  - Mapeo de percepción del ambiente a acciones.
  - Reglas de asociación estímulo-respuesta.
  - Aprendizaje: Identificar buenas políticas.
- Función de recompensas:
  - Define la meta del problema de aprendizaje.
  - Mapeo de par estado-acción a un número que indica la deseabilidad del estado resultante.

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente→ representación del estado del ambiente.
- Política (policy): Determina qué debe hacer el agente en una situación dada.
  - Mapeo de percepción del ambiente a acciones.
  - Reglas de asociación estímulo-respuesta.
  - Aprendizaje: Identificar buenas políticas.
- Función de recompensas:
  - Define la meta del problema de aprendizaje.
  - Mapeo de par estado-acción a un número que indica la deseabilidad del estado resultante.
  - Objetivo del agente es maximizar la recompensa total que recibe a lo largo del tiempo.

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente→ representación del estado del ambiente.
- Política (policy): Determina qué debe hacer el agente en una situación dada.
  - Mapeo de percepción del ambiente a acciones.
  - Reglas de asociación estímulo-respuesta.
  - Aprendizaje: Identificar buenas políticas.
- Función de recompensas:
  - Define la meta del problema de aprendizaje.
  - Mapeo de par estado-acción a un número que indica la deseabilidad del estado resultante.
  - Objetivo del agente es maximizar la recompensa total que recibe a lo largo del tiempo.
  - Agente no puede modificar la recompensa.

Elementos de un problema de aprendizaje por refuerzo

- Agente.
- Ambiente en el que está inmerso el agente→ representación del estado del ambiente.
- Política (policy): Determina qué debe hacer el agente en una situación dada.
  - Mapeo de percepción del ambiente a acciones.
  - Reglas de asociación estímulo-respuesta.
  - Aprendizaje: Identificar buenas políticas.
- Función de recompensas:
  - Define la meta del problema de aprendizaje.
  - Mapeo de par estado-acción a un número que indica la deseabilidad del estado resultante.
  - Objetivo del agente es maximizar la recompensa total que recibe a lo largo del tiempo.
  - Agente no puede modificar la recompensa.

- Función de valor:

- Función de valor:
  - Recompensa total que el agente puede esperar acumular comenzando en un estado dado.

- Función de valor:
  - Recompensa total que el agente puede esperar acumular comenzando en un estado dado.
  - Especifica qué estados son deseables a largo plazo.

- Función de valor:
  - Recompensa total que el agente puede esperar acumular comenzando en un estado dado.
  - Especifica qué estados son deseables a largo plazo.
  - Depende de las acciones que son probables en ese estado y de los estados resultantes de esas acciones.

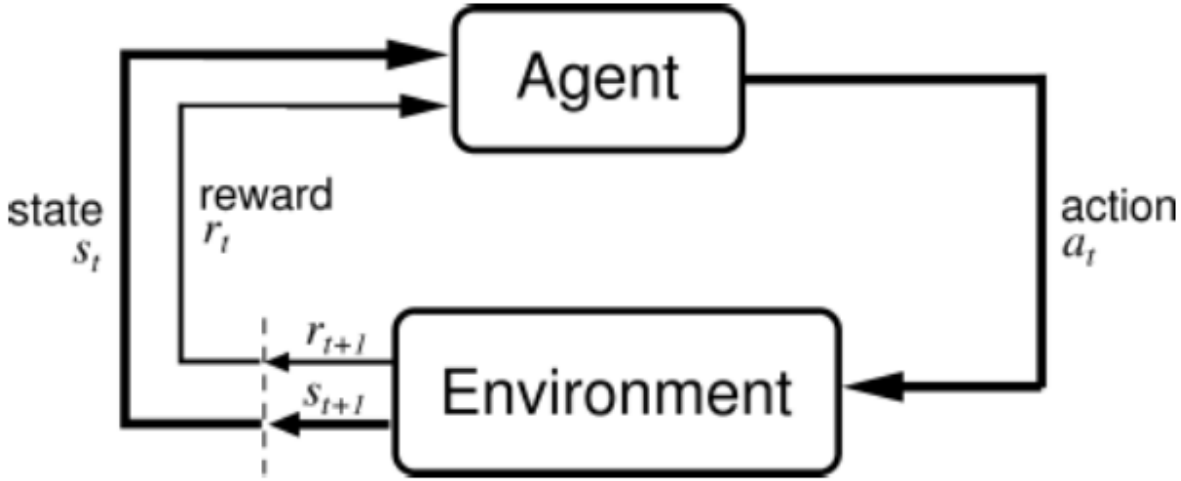
- Función de valor:
  - Recompensa total que el agente puede esperar acumular comenzando en un estado dado.
  - Especifica qué estados son deseables a largo plazo.
  - Depende de las acciones que son probables en ese estado y de los estados resultantes de esas acciones.
  - Se usa para modificar la política.

- Función de valor:
  - Recompensa total que el agente puede esperar acumular comenzando en un estado dado.
  - Especifica qué estados son deseables a largo plazo.
  - Depende de las acciones que son probables en ese estado y de los estados resultantes de esas acciones.
  - Se usa para modificar la política.
- Modelo del ambiente:

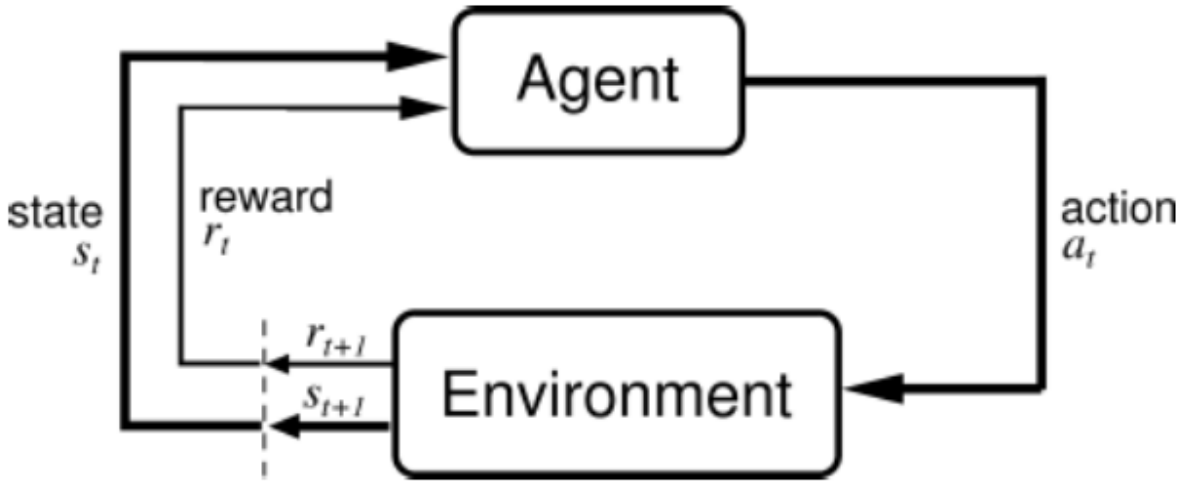
- Función de valor:
  - Recompensa total que el agente puede esperar acumular comenzando en un estado dado.
  - Especifica qué estados son deseables a largo plazo.
  - Depende de las acciones que son probables en ese estado y de los estados resultantes de esas acciones.
  - Se usa para modificar la política.
- Modelo del ambiente:
  - Simula el comportamiento del ambiente.

- Función de valor:
  - Recompensa total que el agente puede esperar acumular comenzando en un estado dado.
  - Especifica qué estados son deseables a largo plazo.
  - Depende de las acciones que son probables en ese estado y de los estados resultantes de esas acciones.
  - Se usa para modificar la política.
- Modelo del ambiente:
  - Simula el comportamiento del ambiente.
  - Dado un estado y una acción, modelo predice recompensa y estado siguientes.

Interfaz Agente-Ambiente

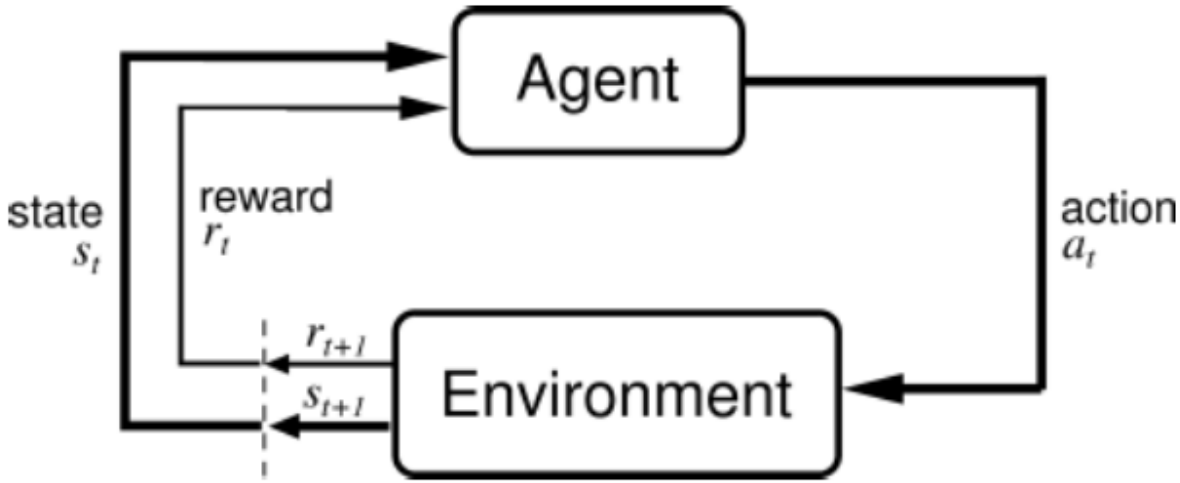


Interfaz Agente-Ambiente



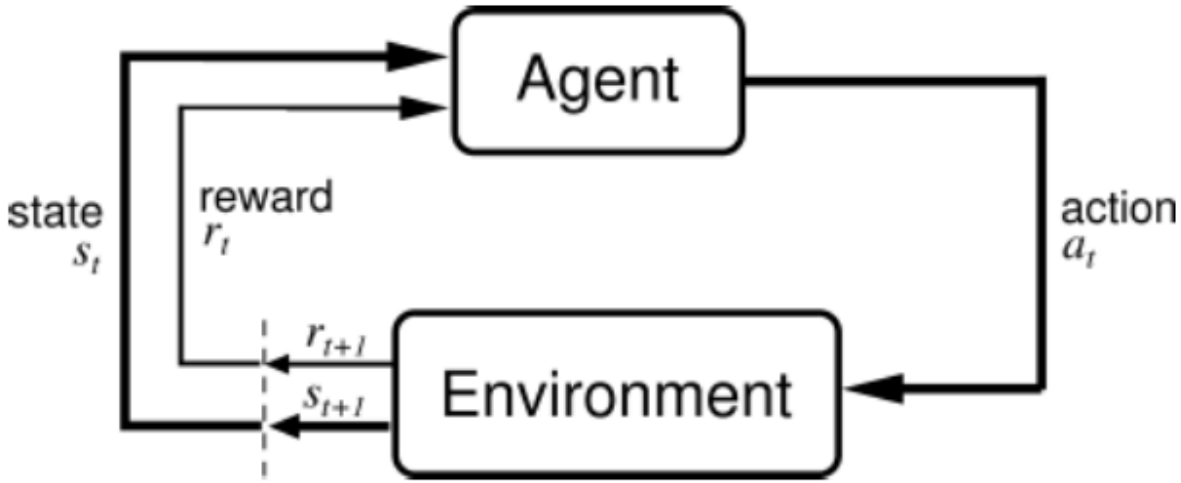
- Ambiente no puede ser modificado arbitrariamente por el agente.

Interfaz Agente-Ambiente



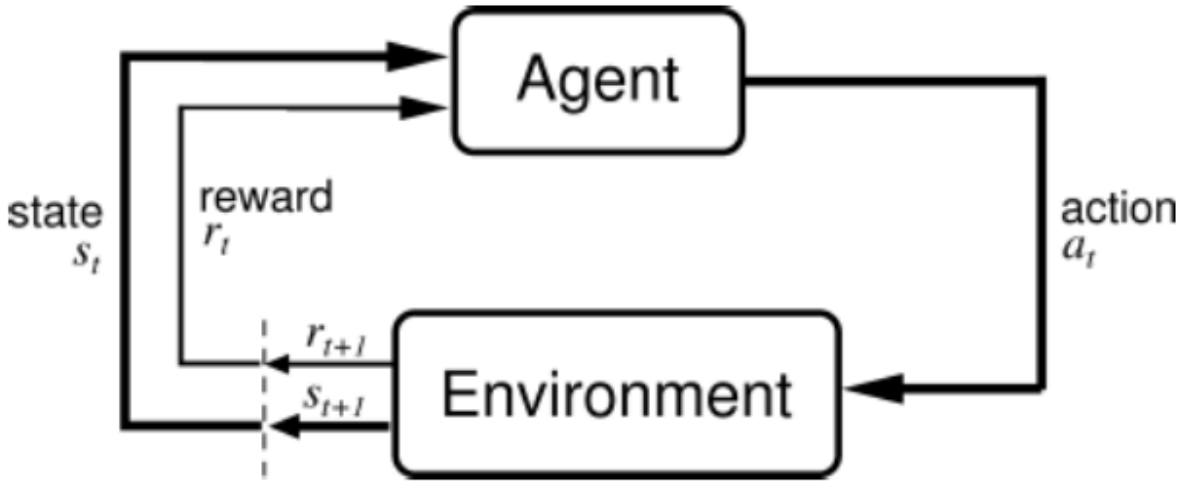
- Ambiente no puede ser modificado arbitrariamente por el agente.
- Ambiente y agente interactúan en pasos $t = 1, 2, \ldots$:

Interfaz Agente-Ambiente



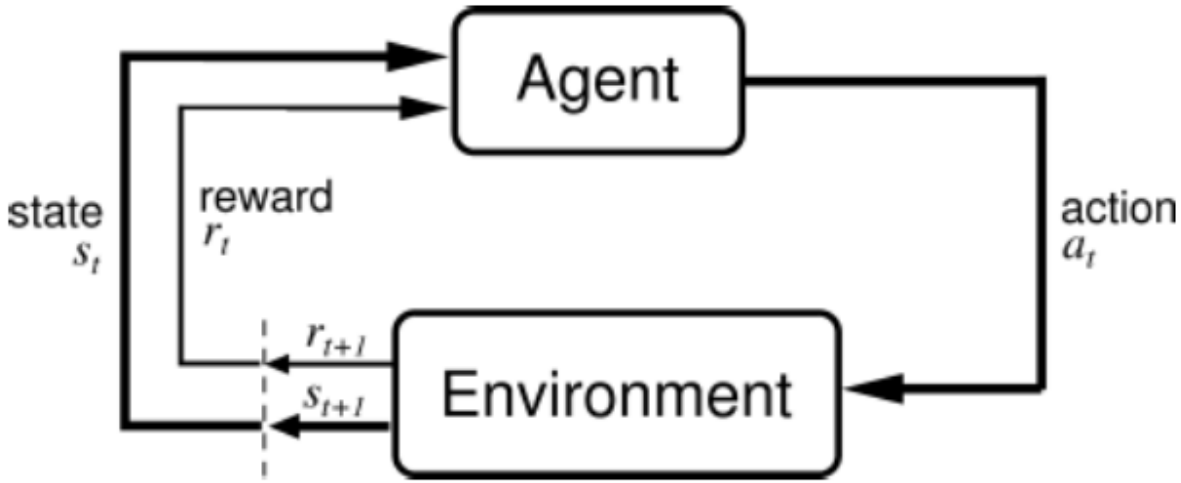
- Ambiente no puede ser modificado arbitrariamente por el agente.
- Ambiente y agente interactúan en pasos $t = 1, 2, \ldots$:
  1. Agente recibe representación del estado del ambiente $s_t \in \mathcal{S}$

Interfaz Agente-Ambiente



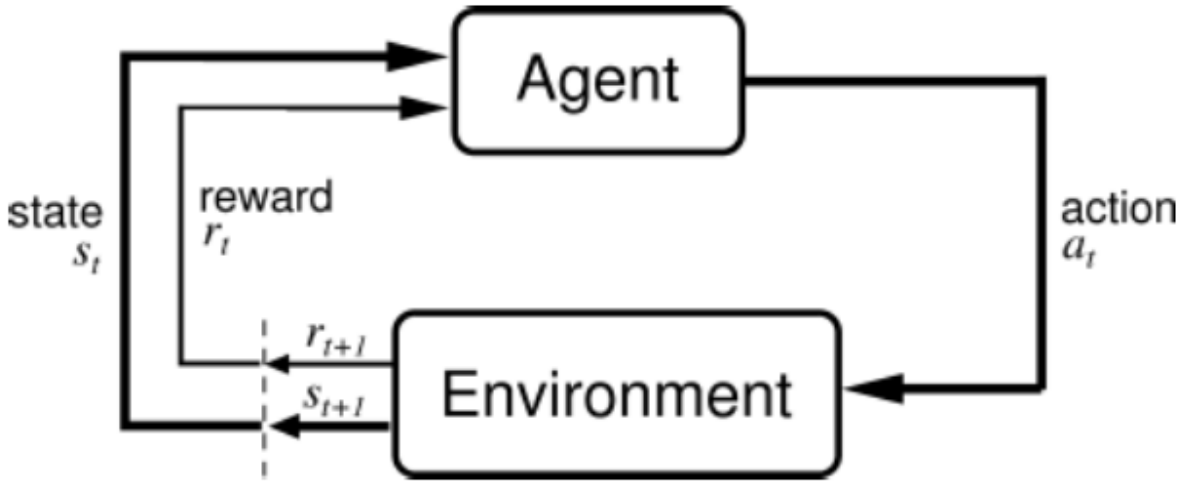
- Ambiente no puede ser modificado arbitrariamente por el agente.
- Ambiente y agente interactúan en pasos $t = 1, 2, \ldots$:
  1. Agente recibe representación del estado del ambiente $s_t \in \mathcal{S}$
  2. Selecciona acción $a_t \in \mathcal{A}(s_t)$.

Interfaz Agente-Ambiente



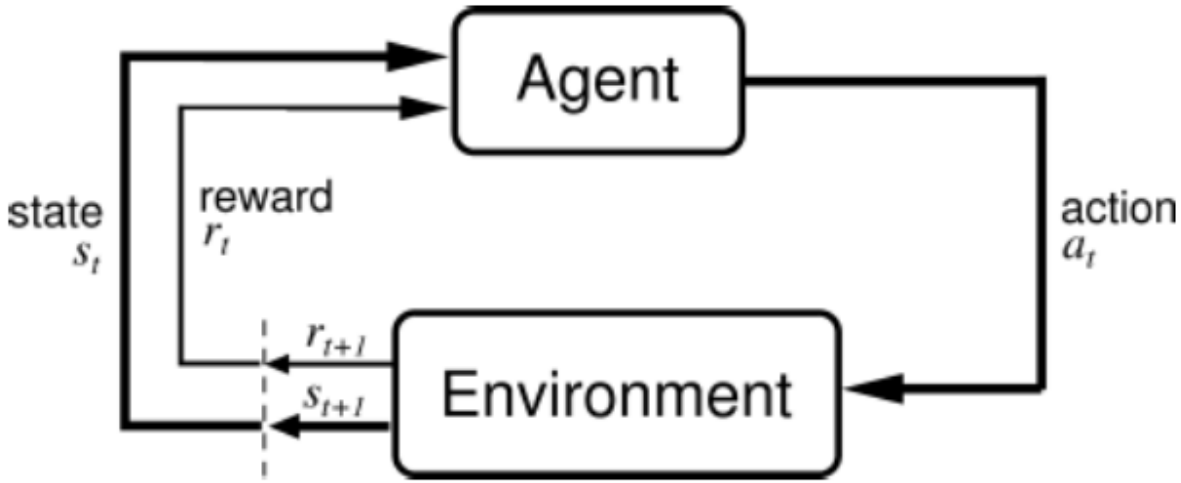
- Ambiente no puede ser modificado arbitrariamente por el agente.
- Ambiente y agente interactúan en pasos $t = 1, 2, \ldots$:
  1. Agente recibe representación del estado del ambiente $s_t \in \mathcal{S}$
  2. Selecciona acción $a_t \in \mathcal{A}(s_t)$.
  3. Recibe recompensa $r_{t+1} \in \mathbb{R}$

Interfaz Agente-Ambiente



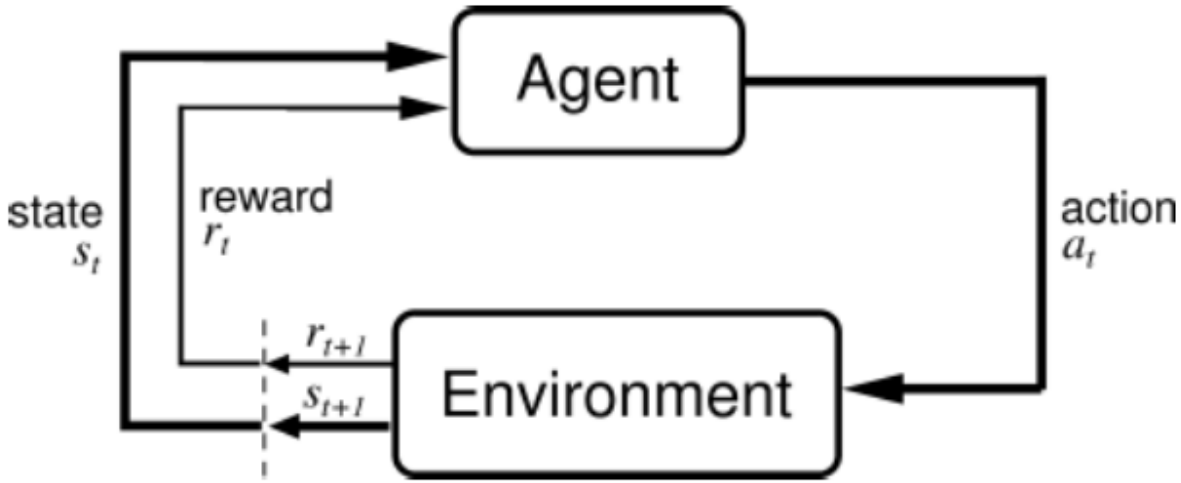
- Ambiente no puede ser modificado arbitrariamente por el agente.
- Ambiente y agente interactúan en pasos $t = 1, 2, \ldots$:
  1. Agente recibe representación del estado del ambiente $s_t \in \mathcal{S}$
  2. Selecciona acción $a_t \in \mathcal{A}(s_t)$.
  3. Recibe recompensa $r_{t+1} \in \mathbb{R}$ y pasa a estado $s_{t+1} \in \mathcal{S}$

Interfaz Agente-Ambiente



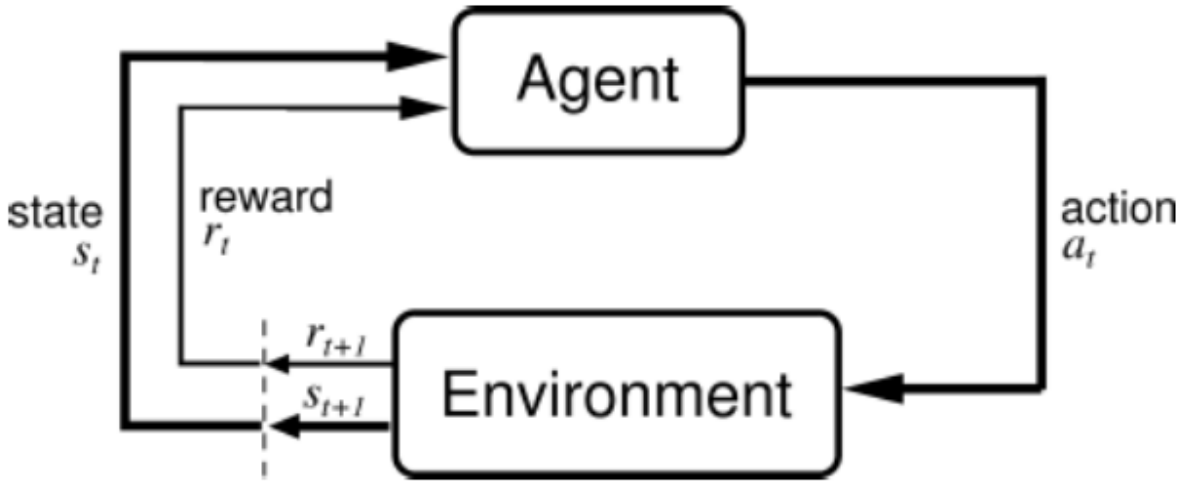
- Ambiente no puede ser modificado arbitrariamente por el agente.
- Ambiente y agente interactúan en pasos $t = 1, 2, \ldots$:
  1. Agente recibe representación del estado del ambiente $s_t \in \mathcal{S}$
  2. Selecciona acción $a_t \in \mathcal{A}(s_t)$.
  3. Recibe recompensa $r_{t+1} \in \mathbb{R}$ y pasa a estado $s_{t+1} \in \mathcal{S}$
- Mapeo estado $\rightarrow$ probabilidad de seleccionar acción

Interfaz Agente-Ambiente



- Ambiente no puede ser modificado arbitrariamente por el agente.
- Ambiente y agente interactúan en pasos $t = 1, 2, \ldots$:
  1. Agente recibe representación del estado del ambiente $s_t \in \mathcal{S}$
  2. Selecciona acción $a_t \in \mathcal{A}(s_t)$.
  3. Recibe recompensa $r_{t+1} \in \mathbb{R}$ y pasa a estado $s_{t+1} \in \mathcal{S}$
- Mapeo estado $\rightarrow$ probabilidad de seleccionar acción: política (policy).

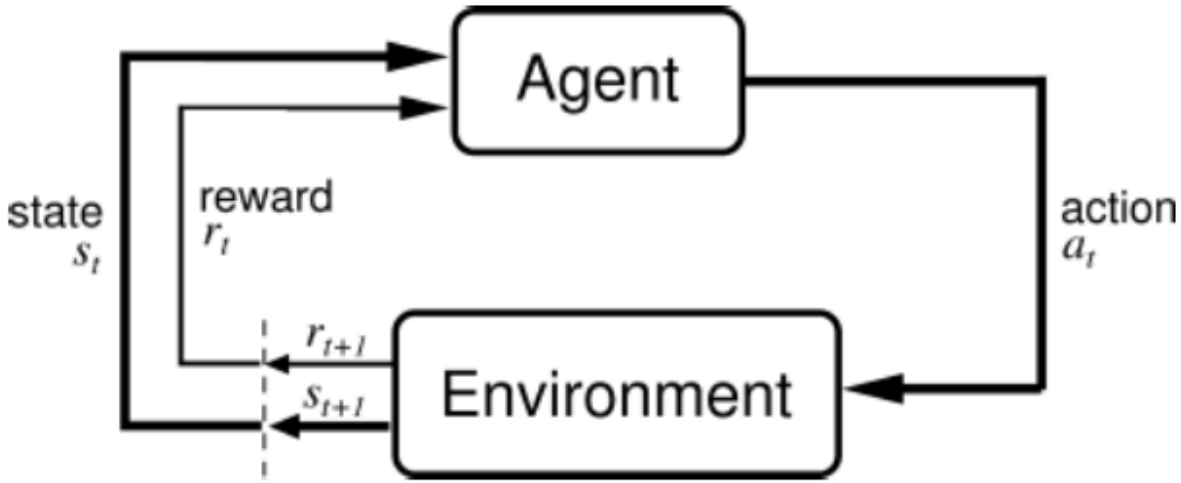
Interfaz Agente-Ambiente



- Ambiente no puede ser modificado arbitrariamente por el agente.
- Ambiente y agente interactúan en pasos $t = 1, 2, \ldots$:
  1. Agente recibe representación del estado del ambiente $s_t \in \mathcal{S}$
  2. Selecciona acción $a_t \in \mathcal{A}(s_t)$.
  3. Recibe recompensa $r_{t+1} \in \mathbb{R}$ y pasa a estado $s_{t+1} \in \mathcal{S}$
- Mapeo estado $\rightarrow$ probabilidad de seleccionar acción: política (policy).

$$\pi_t(s, a)$$

Interfaz Agente-Ambiente



- Ambiente no puede ser modificado arbitrariamente por el agente.
- Ambiente y agente interactúan en pasos $t = 1, 2, \ldots$:
  1. Agente recibe representación del estado del ambiente $s_t \in \mathcal{S}$
  2. Selecciona acción $a_t \in \mathcal{A}(s_t)$.
  3. Recibe recompensa $r_{t+1} \in \mathbb{R}$ y pasa a estado $s_{t+1} \in \mathcal{S}$
- Mapeo estado $\rightarrow$ probabilidad de seleccionar acción: política (policy).

$$\pi_t(s, a) = \mathbf{P}\left[a_t = a \mid s_t = s\right]$$