

INF 1771 – Inteligência Artificial

Aula 21 – Aprendizado Por Reforço

Edirlei Soares de Lima
<elima@inf.puc-rio.br>

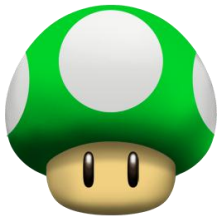


Formas de Aprendizado

- ❏ Aprendizado Supervisionado
 - ❏ Árvores de Decisão.
 - ❏ K-Nearest Neighbor (KNN).
 - ❏ Support Vector Machines (SVM).
 - ❏ Redes Neurais.

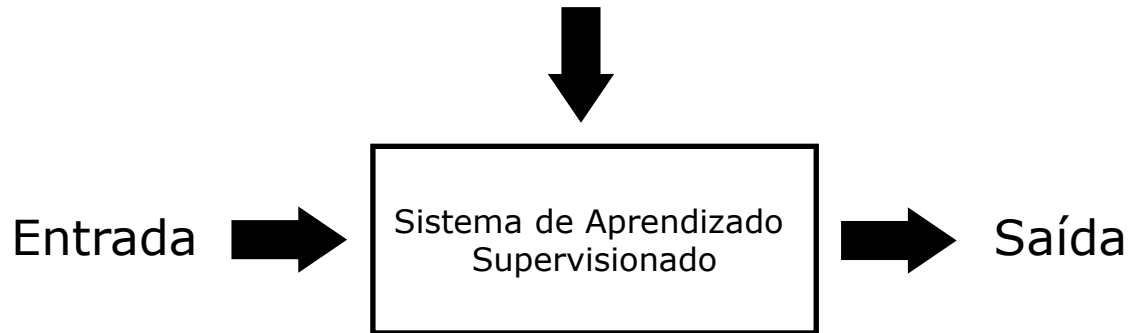
- ❏ Aprendizado Não-Supervisionado

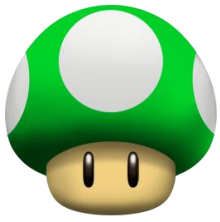
- ❏ **Aprendizado Por Reforço**



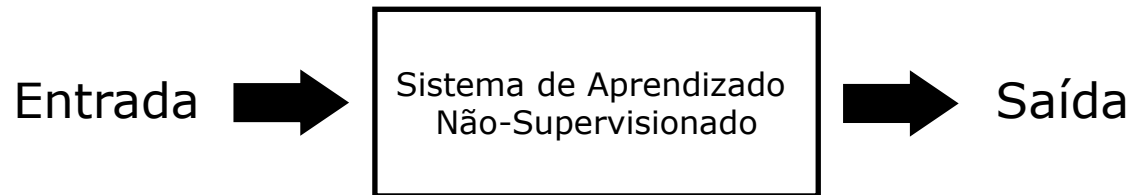
Aprendizado Supervisionado

Informação de Treinamento = Entradas + Saídas





Aprendizado Não-Supervisionado

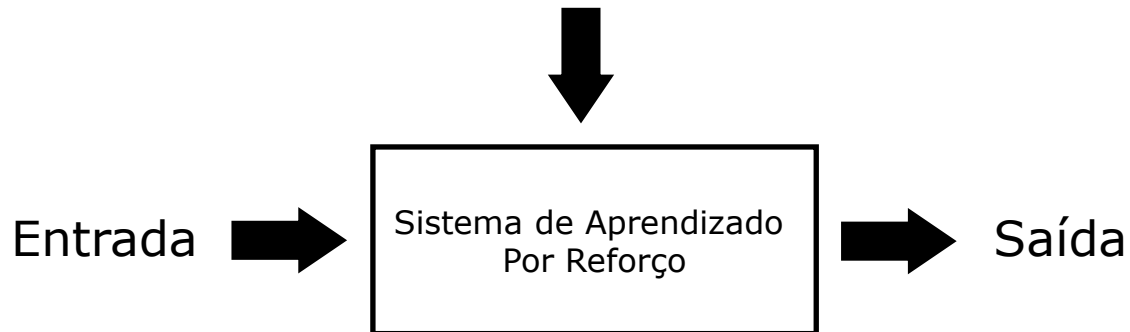


Objetivo: Agrupar objetos semelhantes

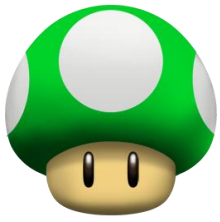


Aprendizado Por Reforço

Informação de Treinamento = Avaliação(Recompensas, Punições)



Objetivo: Conseguir o máximo de reforço possível

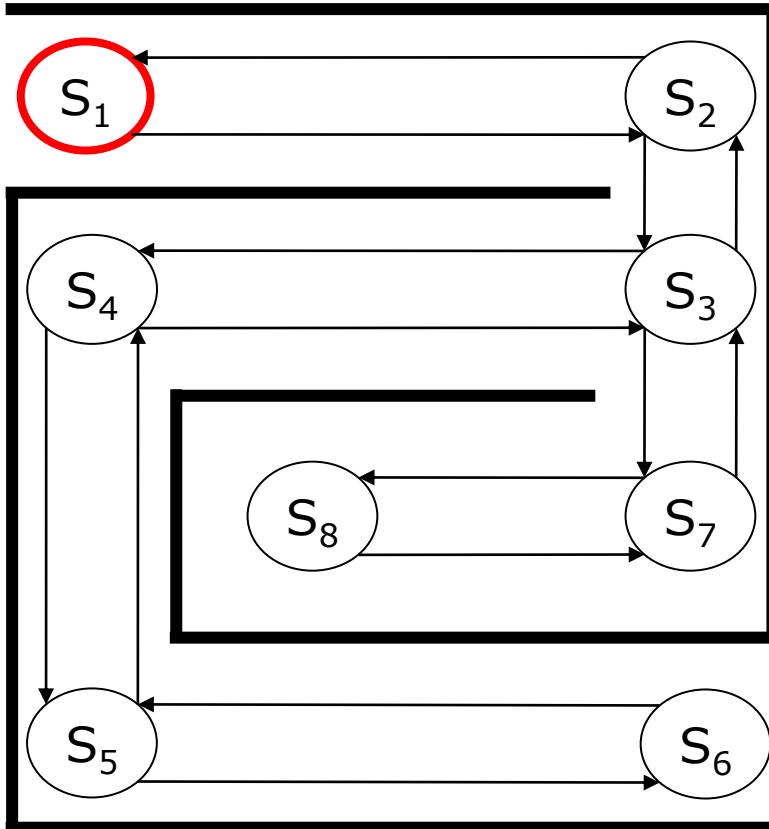


Introdução

- ❏ **Como um agente aprende a escolher ações apenas interagindo com o ambiente?**
 - ❏ Muitas vezes é impraticável o uso de aprendizado supervisionado.
 - ❏ Como obter exemplos do comportamento correto e representativo para qualquer situação?
 - ❏ E se o agente for atuar em um ambiente desconhecido?
 - ❏ Exemplos:
 - ❏ Criança adquirindo coordenação motora.
 - ❏ Robô interagindo com um ambiente para atingir objetivos.



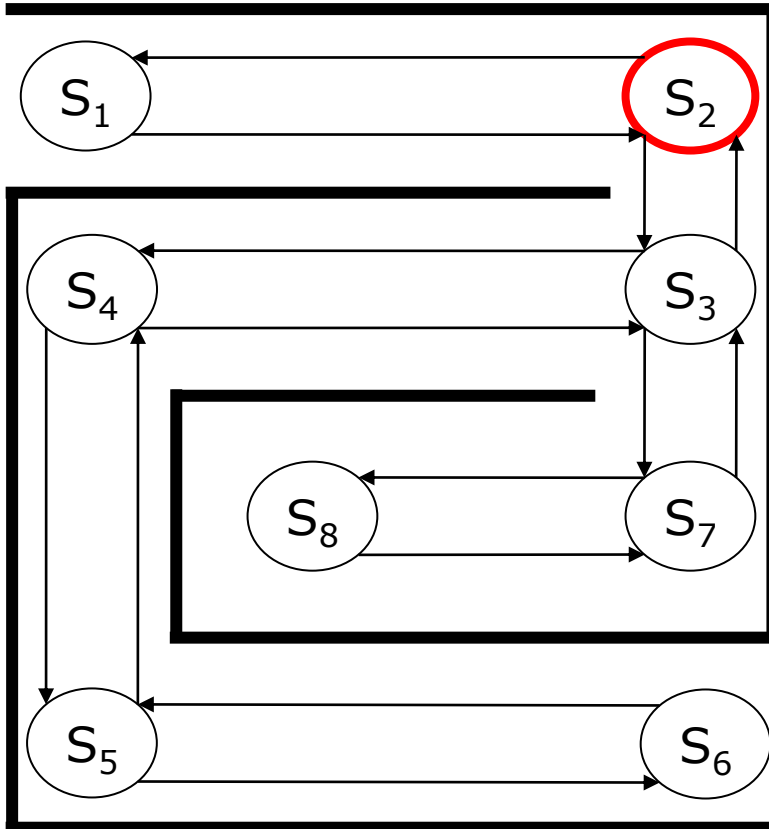
Exemplo



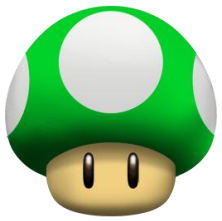
- As setas indicam a “força” entre dois estados.
- Inicialmente todas as setas possuem o mesmo valor de força.
- Iniciando em S_1 como chegar em S_6 ?



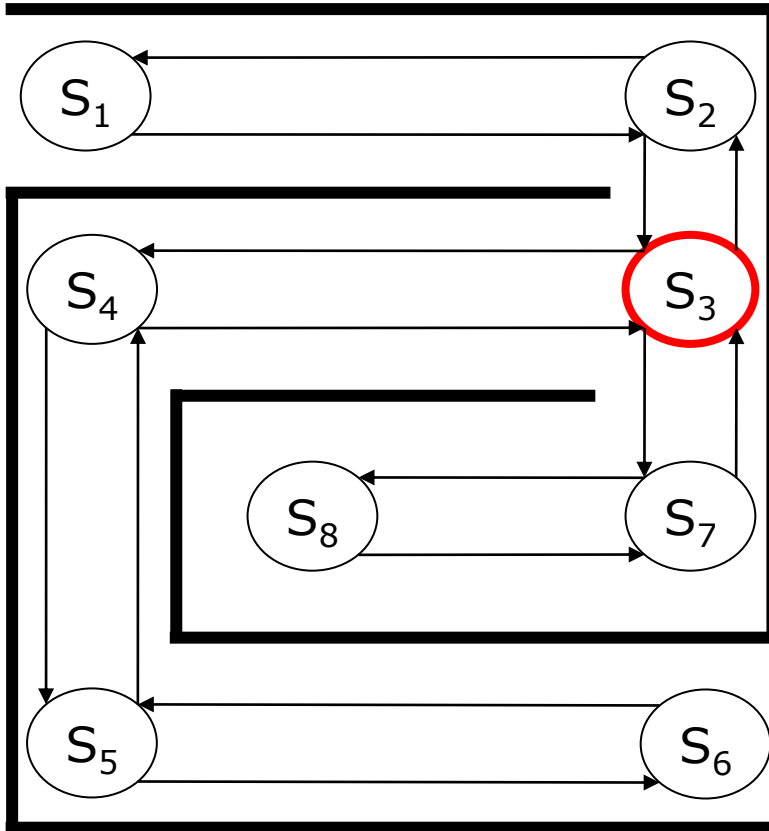
Exemplo



- ❏ Próximo estado é escolhido aleatoriamente de um dos próximos estados possíveis (ponderado pela força da associação).
- ❏ A primeira ação só pode levar para S_2 .



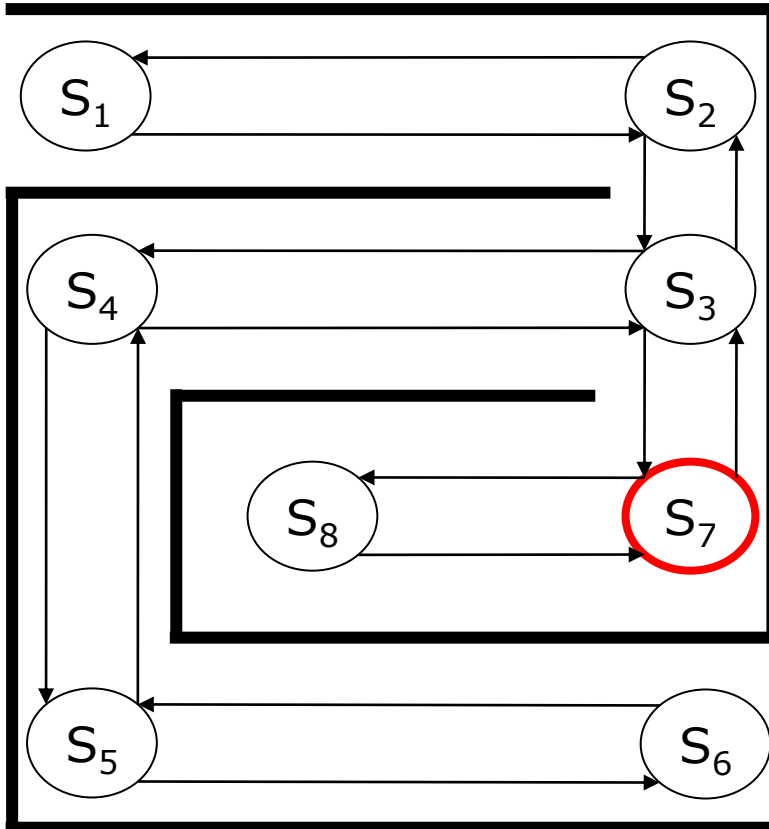
Exemplo



- Supondo que a próxima escolha leve a S_3 .



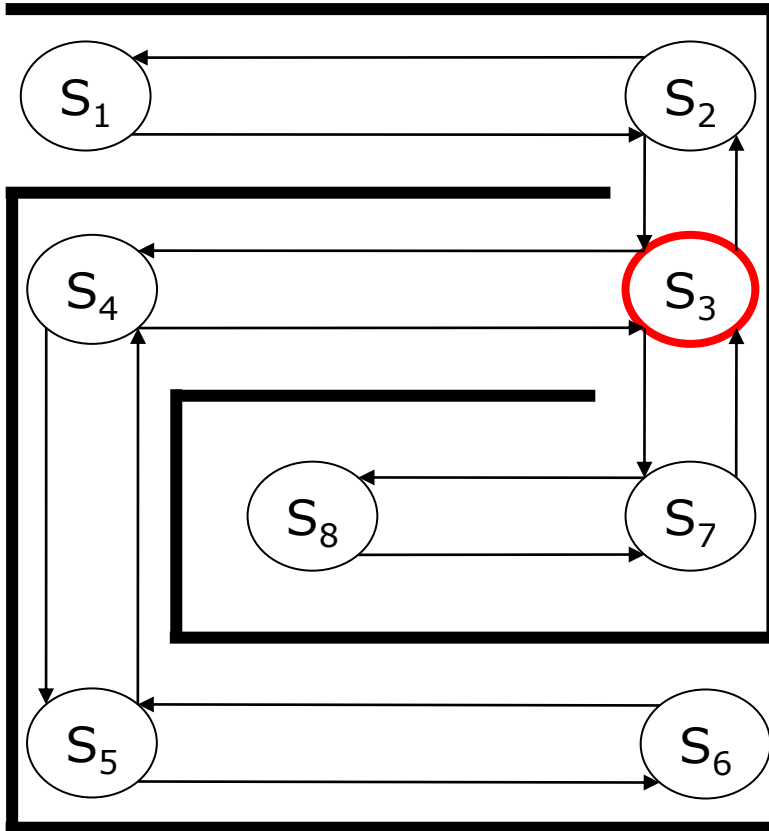
Exemplo



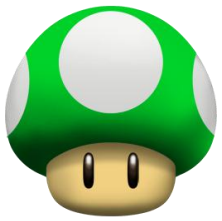
- Em S_3 , as possíveis escolhas são S_2 , S_4 , ou S_7 .
- Vamos supor que S_7 é escolhido aleatoriamente.



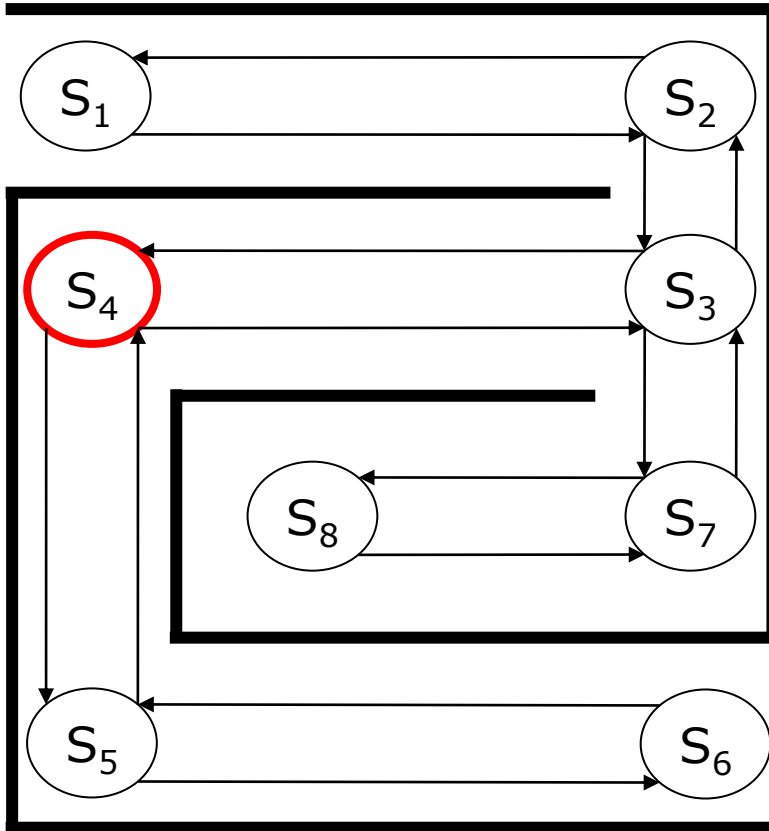
Exemplo



- Por sorteio, S_3 é o próximo escolhido.



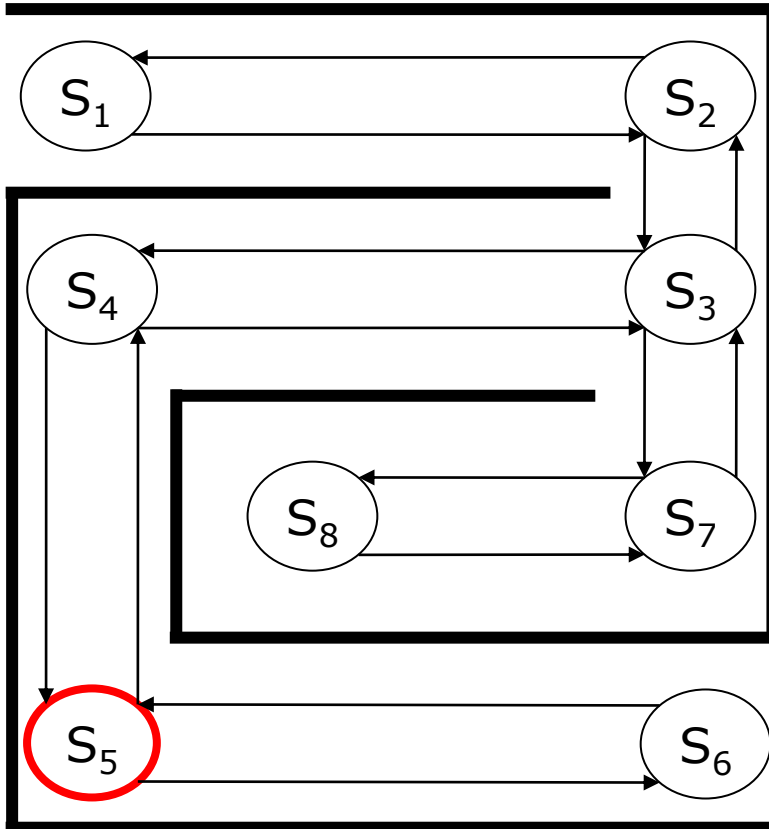
Exemplo



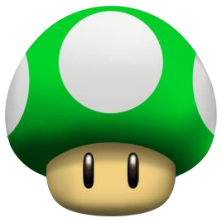
- 🔑 O próximo é S_4 .



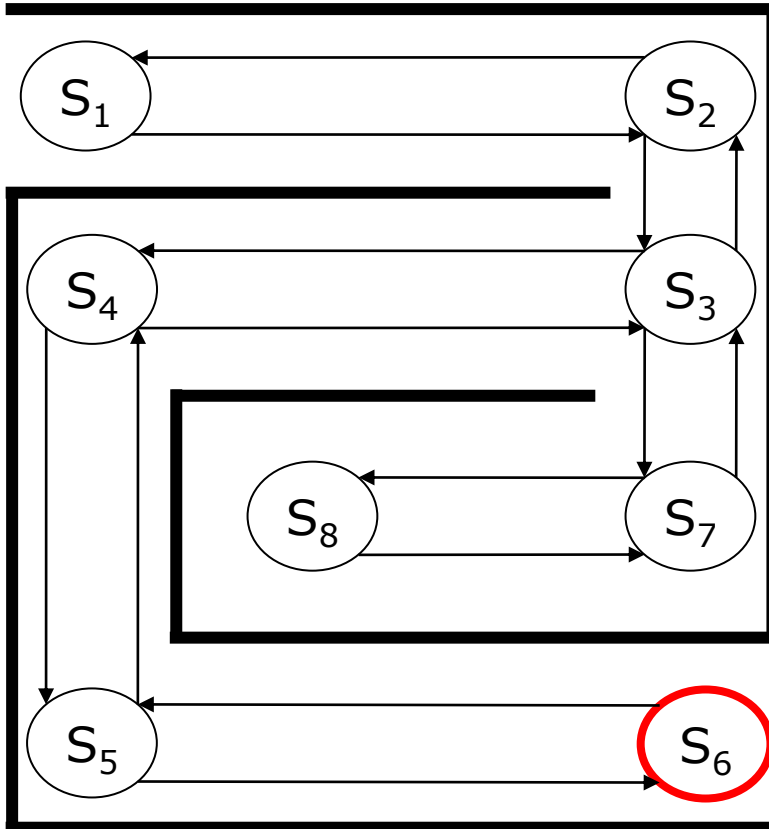
Exemplo



- 🔑 E então S_5 é escolhido aleatoriamente.



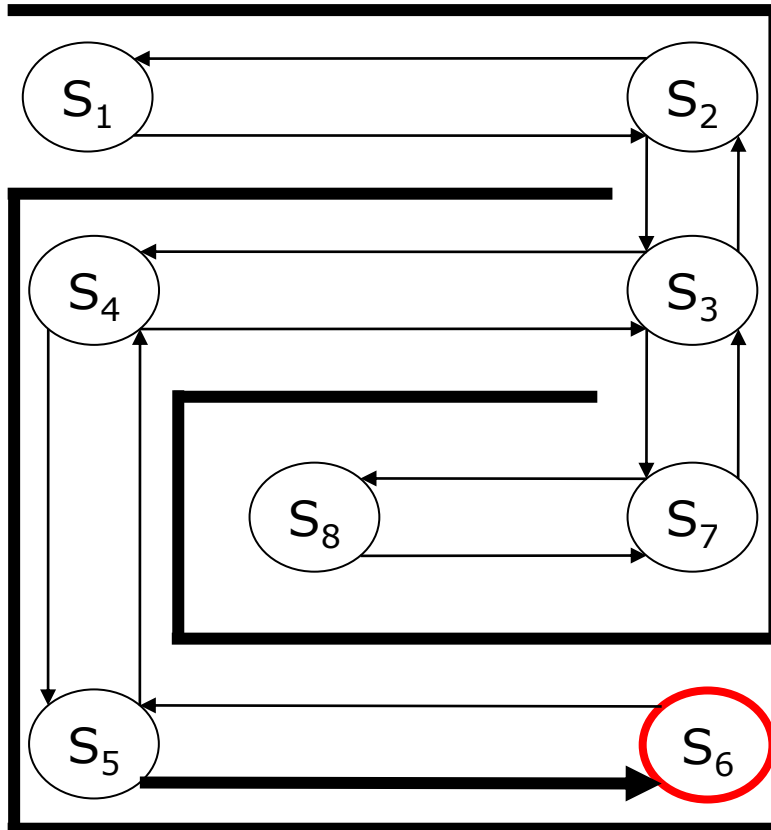
Exemplo



- 🔑 E finalmente atingimos o objetivo S_6 .



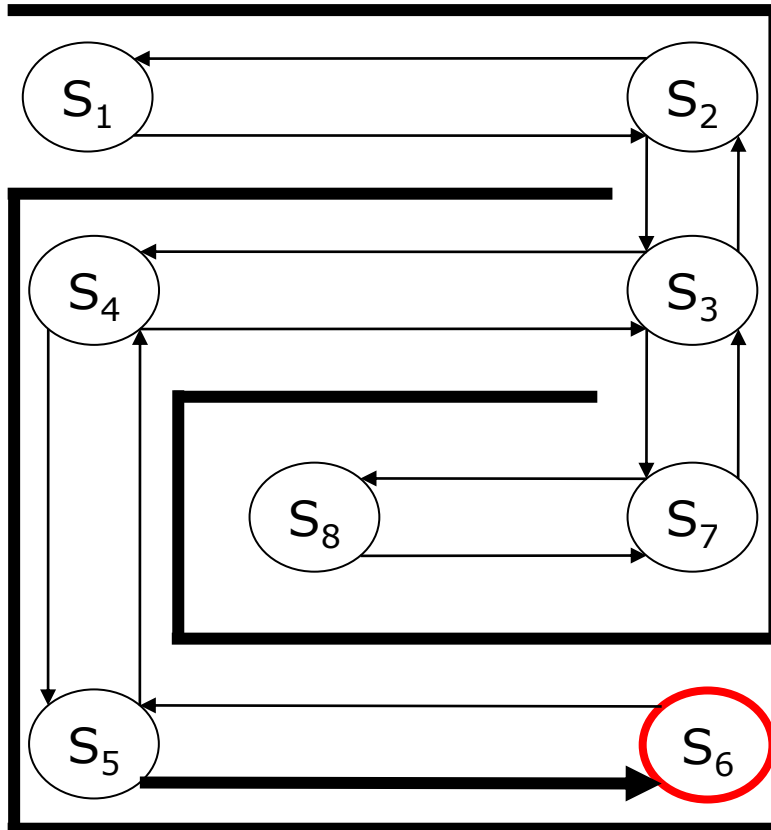
Exemplo



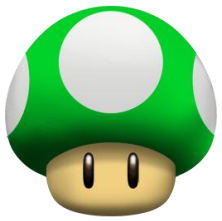
- Quando o estado objetivo é atingida, reforça-se a conexão entre ele e o estado que levou a ele.
- Na próxima vez que S_5 for alcançado, parte da força de associação será passada para S_4 .



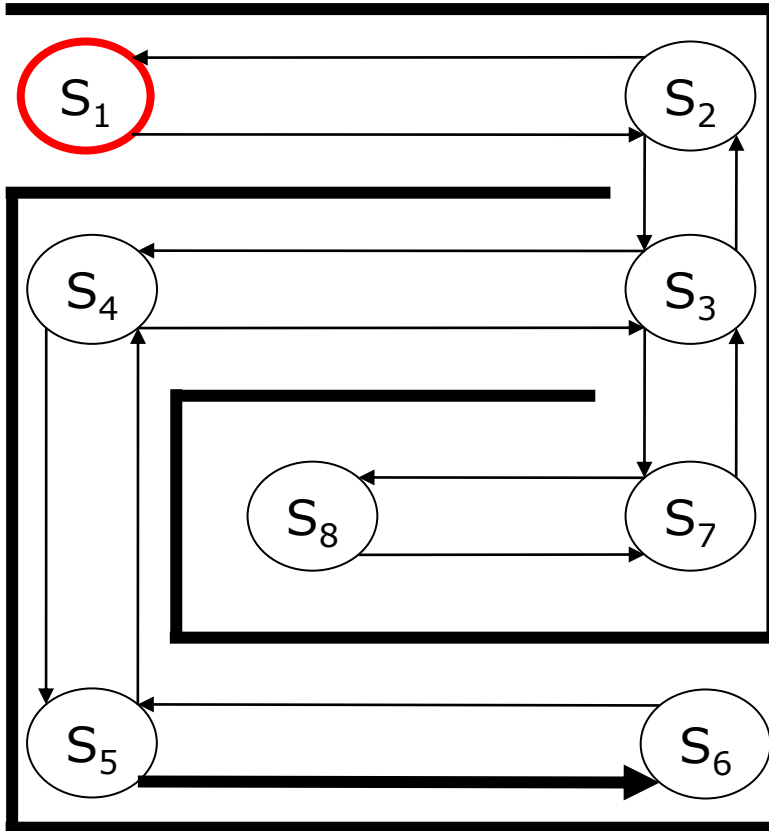
Exemplo



- Quando o estado objetivo é atingida, reforça-se a conexão entre ele e o estado que levou a ele.
- Na próxima vez que S_5 for alcançado, parte da força de associação será passada para S_4 .



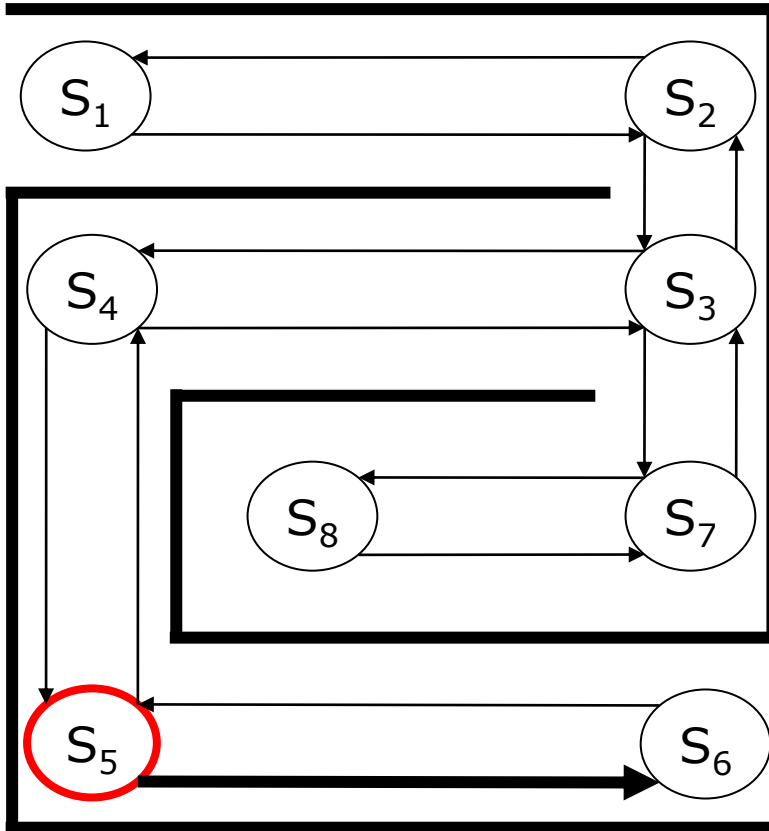
Exemplo



- 🔑 Iniciando novamente o percurso.



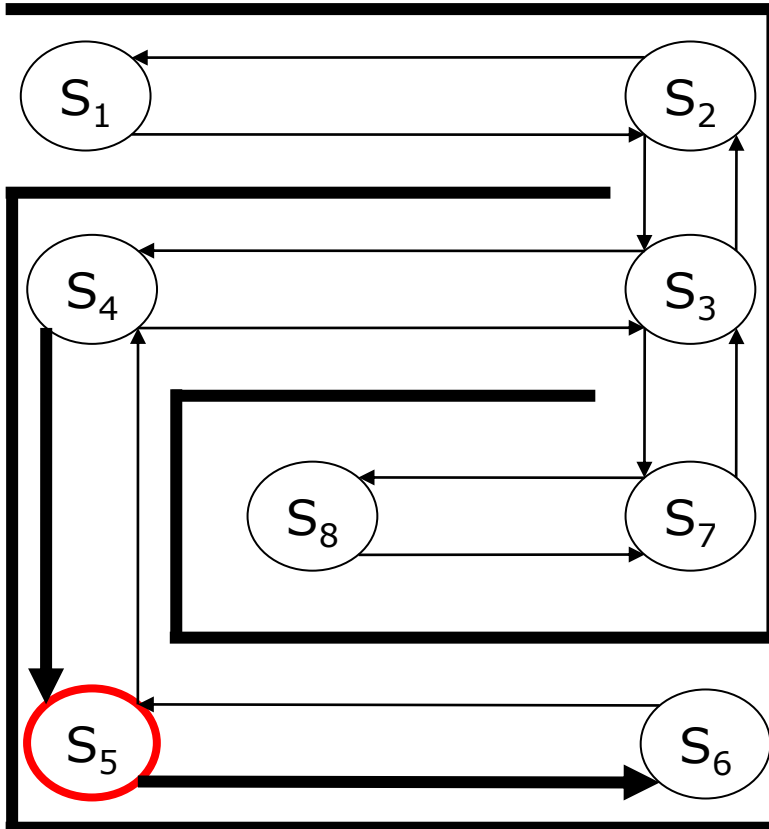
Exemplo



- Supondo que após alguns movimentos o agente chega novamente em S_5 .



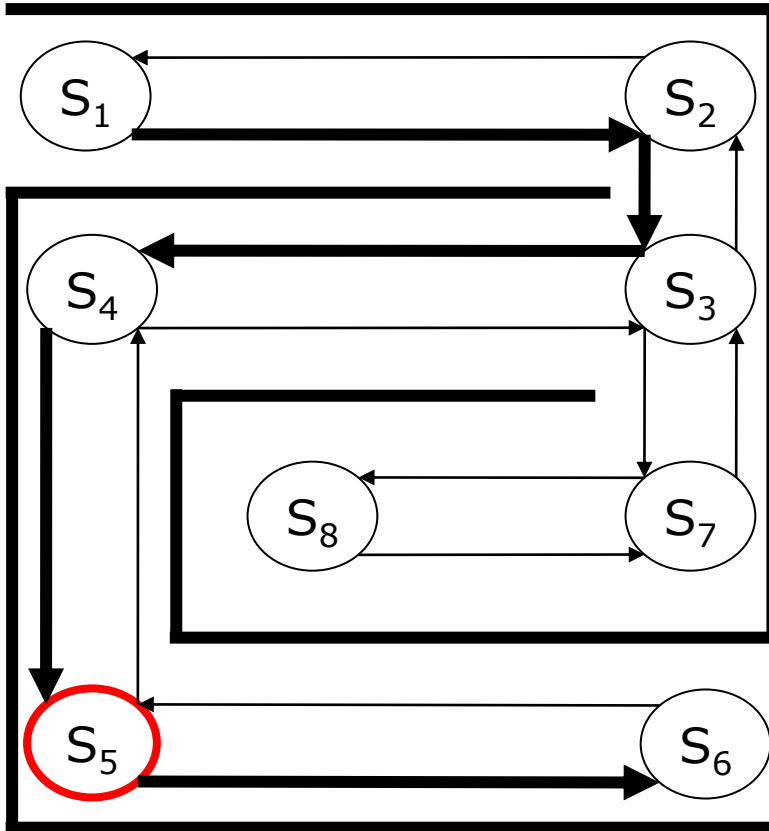
Exemplo



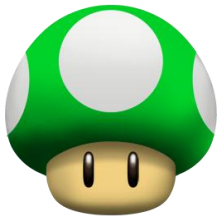
- ❗ S₅ tem grande chance de atingir a meta pela rota com mais força.
- ❗ Em aprendizado por reforço, essa "força" é passada de volta para o estado anterior.
- ❗ Esse processo leva a criar um caminho entre o início e a meta.



Exemplo

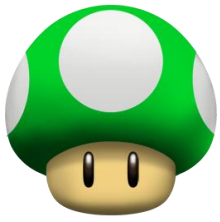


- Após reiniciar o percurso varias vezes, o agente aprenderia o melhor caminho a ser seguido.

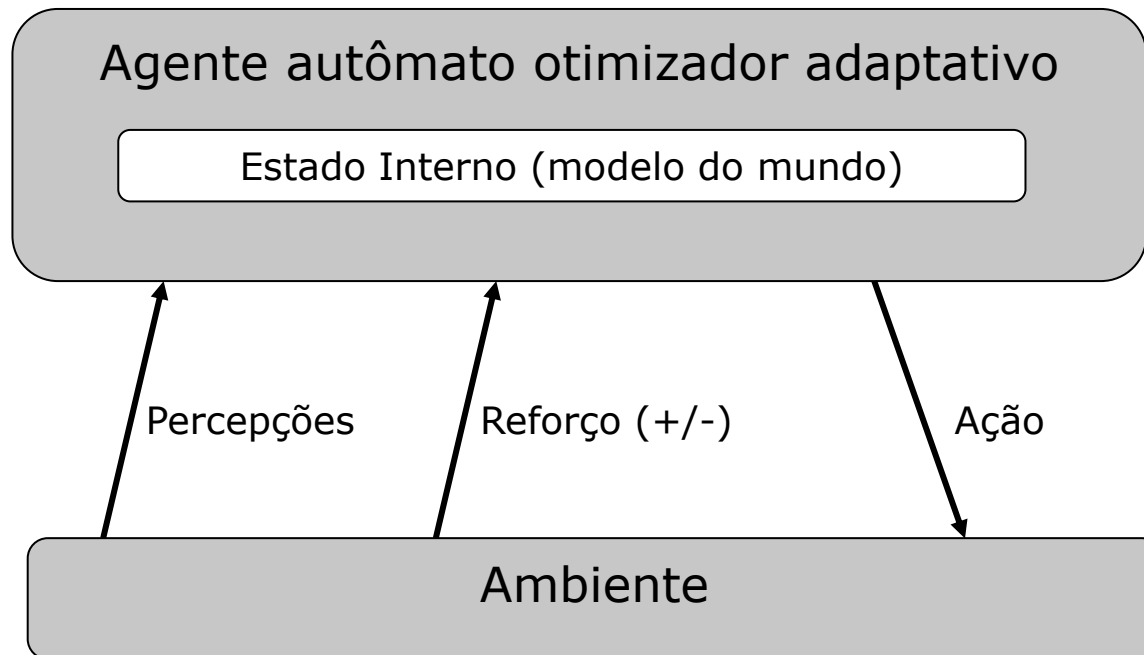


Aprendizado Por Reforço

- ❏ Um agente em um **ambiente**.
- ❏ A cada instante do tempo t :
 - ❏ o agente está em um **estado** s .
 - ❏ executa uma **ação** a .
 - ❏ vai para um **estado** s' .
 - ❏ recebe uma **recompensa** r .
- ❏ Problema da aprendizagem por reforço:
 - ❏ Como escolher uma política de ações que **maximize** o **total de recompensas** recebidas pelo agente.



Aprendizado Por Reforço





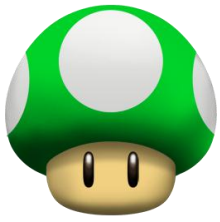
Aprendizado Por Reforço

❏ **Processo de decisão de Markov (MDP)**

- ❏ Conjunto de **estados** S .
- ❏ Conjunto de **ações** A .
- ❏ Uma função de **recompensa** $r(s, a)$.
- ❏ Uma função de **transição de estados** $\alpha(s, a)$.

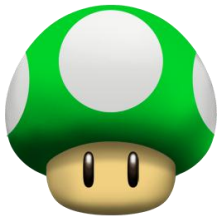
❏ **Política de ações** $\pi(S)$:

- ❏ $\pi : S \rightarrow A$



Estados e Ações

- ❏ **Estado:** conjunto de características que descrevem o ambiente.
 - ❏ Formado pelas **percepções do agente + modelo do mundo.**
 - ❏ Deve **prover informação** para o agente de quais ações podem ser executadas.
- ❏ A representação deste estado deve ser suficiente para que o agente **tome suas decisões.**
- ❏ A decisão de que ação tomar não pode depender da sequência de estados anteriores.
 - ❏ Um tabuleiro de dama satisfaz esta propriedade, mas de xadrez não.



A Função de Recompensa

- ❗ **Feedback do ambiente** sobre o comportamento do agente.
- ❗ Indicada por $R(S, A) \rightarrow R$
 - ❗ $r(s,a)$ indica a recompensa recebida quando o agente está no estado s e executa a ação a .
 - ❗ Pode ser determinística ou estocástica



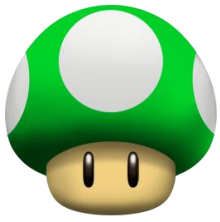
Função de Transição de Estados

- ❏ $\alpha(S, A) \rightarrow S$
- ❏ $\alpha(s, a)$ indica em qual estado o agente está, dado que:
 - ❏ Estava no estado s .
 - ❏ executou a ação a .
- ❏ Ambientes não-determinísticos:
 - ❏ $\alpha(s, a, s')$
 - ❏ Indica a probabilidade de ir para um estado s' dado que estava em s e executou a ação a .



Exemplos de Problemas

Problema	Estados	Ações	Recompensas
Agente jogador de damas.	Configurações do tabuleiro.	Mover uma determinada peça.	+ Capturas - Perdas
Agente em jogo de luta.	Posição, energia dos lutadores, tempo, estar ou estar sendo atacado, etc...	Mover-se em uma direção, lançar magia, bater, etc...	+ Tirar energia do oponente. - Perder energia.
Agente patrulhador.	Posição no mapa (atual e passadas), ociosidade da vizinhança, etc...	Ir para algum lugar vizinho do Mapa	Ociosidade (tempo sem visitas) do lugar visitado Atualmente.



Política de Ações $\pi(s)$

- ❏ Função que modela o comportamento do agente
 - ❏ Mapeia estados em ações.
 - ❏ Pode ser vista como um conjunto de regras do tipo $s_n \rightarrow a_m$

- ❏ Exemplo:

Se estado $s =$ (inimigo próximo, estou perdendo) **então**

ação $a =$ (usar magia);

Se estado $s =$ (outro estado) **então**

ação $a =$ (outra ação);



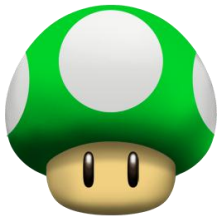
Função Valor dos Estados $V_{\pi}(s)$ ($S \rightarrow R$)

- ❏ **Como saber se um determinado estado é bom ou ruim?**
 - ❏ A função valor $V_{\pi}(s)$ expressa esta noção, em termos das recompensas e da política de ações.
 - ❏ **Representa a recompensa a receber em um estado s , mais as recompensas futuras se ele seguir uma política de ações π .**
 - ❏ Exemplo: tornar-se diretor é bom pelo que o cargo permite e permitirá nas próximas promoções.
- ❏ $V_{\pi}(s_0) = r_0 + r_1 + r_2 + r_3 + \dots$
 - ❏ Problema: se o tempo for infinito, a função valor do estado tende a infinito.



Função Valor das Ações $Q_{\pi}(s, a) : (S, A) \rightarrow R$

- ❏ **A função valor das ações** $Q_{\pi}(s, a)$ indica a soma das recompensas a obter, dado que:
 - ❏ o agente está no estado s .
 - ❏ executou uma ação a .
 - ❏ a partir daí, seguiu uma política de ações π .
- ❏ **$Q_{\pi}(s, a) = r(s, a) + V_{\pi}(s')$** , onde:
 - ❏ $S' = \alpha(s, a)$ = indica em qual estado o agente está, dado que ele estava no estado s e executou a ação a .
 - ❏ **O valor da ação é a recompensa da ação mais o valor do estado para onde o agente vai devido à ação.**



Aprendizado Por Reforço

❗ **O aprendizado por reforço consiste em aprender uma política de ações π^* ótima, que maximiza a função $V\pi(V^*)$ ou a função $Q\pi(Q^*)$**

❗ $\pi^* = \operatorname{argmax}_{\pi}[V\pi(s)]$

❗ Em outras palavras, de que maneira o agente deve agir para maximizar as suas recompensas futuras.



Aprendendo uma Política Ótima

- ❏ Se o **ambiente é determinístico** $\alpha(s, a) = s'$ (função de transição de estados) é conhecida e $r(s, a)$ (função de recompensa) é conhecida, é possível computar uma política ótima:
 - ❏ $V^*(s) = \max_a [r(s, a) + V^*(\alpha(s, a))]$
 - ❏ $\pi^*(s) = \operatorname{argmax}_a [r(s, a) + V^*(\alpha(s, a))]$
 - ❏ Tempo polinomial.
 - ❏ **Problema:** se não temos conhecimento prévio das recompensas e transição de estados.
- ❏ Se o **ambiente é não-determinístico**, mas a função de probabilidade de transição de estados for conhecida, também é possível computar π^*
 - ❏ **Problema:** É difícil estimar probabilidades.



Q Learning

❏ Algoritmo Q Learning

❏ Para todo estado s e ação a , inicialize a tabela $Q[s][a] = 0$;

❏ Para sempre, faça:

❏ Observe o estado atual s ;

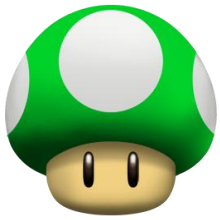
❏ Escolha uma ação a e execute;

❏ Observe o próximo estado s' e recompensa r .

❏ Atualize a tabela Q :

❏ $Q[s][a] = r + \max_{a'}(Q[s'][a'])$

Usufruir valores conhecidos ou explorar valores não computados?



Dilema de Explorar ou Usufruir

❏ **Usufruir**

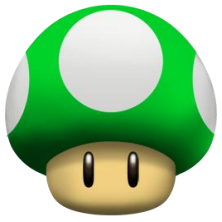
- ❏ Escolher a ação que atualmente está com maior valor $Q(s,a)$

❏ **Explorar**

- ❏ Escolher uma ação randômica, para que seu valor $Q(s,a)$ seja atualizado

❏ **Dilema**

- ❏ Dado que eu aprendi que $Q(s, a)$ vale 100, vale a pena tentar executar a ação a' se $Q(s, a')$ por enquanto vale 20?
- ❏ Depende do ambiente, da quantidade de ações já tomadas e da quantidade de ações restantes.

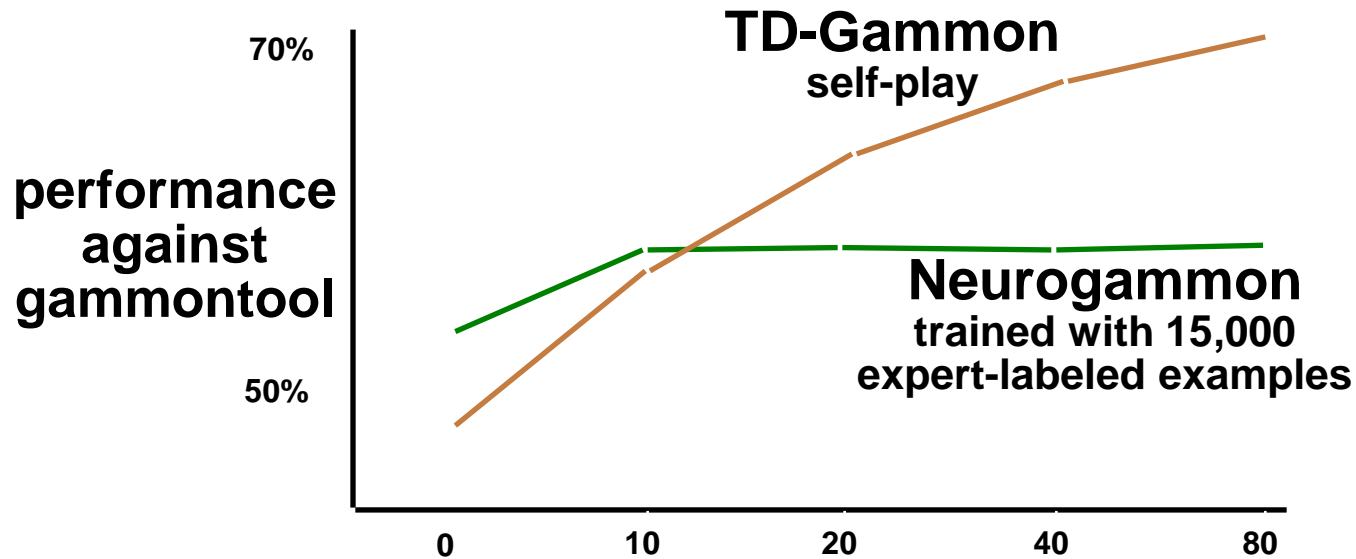


Aplicações

- 💡 [Tesauro, 1995] Modelagem do **jogo de gamão** como um problema de aprendizagem por reforço:
 - 💡 Vitória: +100
 - 💡 Derrota: -100
- 💡 Após 1 milhão de partidas contra ele mesmo, joga tão bem quanto o melhor jogador humano.



Aplicações



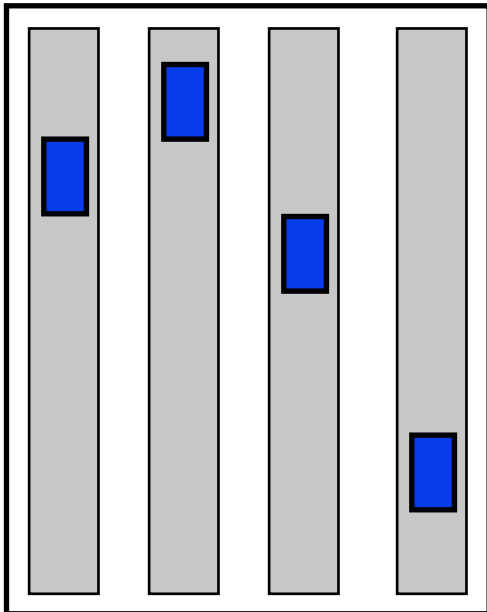
Tesauro, 1992



Aplicações

🔑 [Crites and Barto, 1996] Controle de Elevadores

10 andares, 4 cabines



Estados: estados dos botões; posição, direção, e estado de movimentação dos elevadores; passageiros nos elevadores e esperando.

Ações: parar em, passar, próximo andar.

Recompensas: simplesmente -1 por tempo em que cada pessoa ficava esperando.