

# **APRENDIZAGEM POR REFORÇO**

Luís Morgado

ISEL-ADEETC

# APRENDIZAGEM ASSOCIATIVA

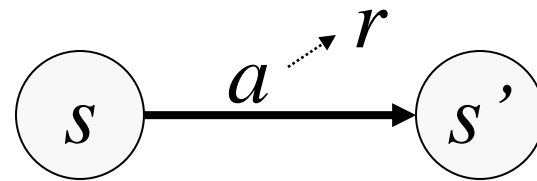
- Estado pode evoluir ao longo do tempo
  - Estados observados
    - $s \in S$
  - Acções realizadas
    - $a \in A$
  - Reforços obtidos
    - $r \in \mathbb{R}$
  - **Valor de num estado realizar uma acção**
    - $Q(s,a)$

# APRENDIZAGEM POR REFORÇO

- Aprendizagem a partir da **interacção** com o ambiente

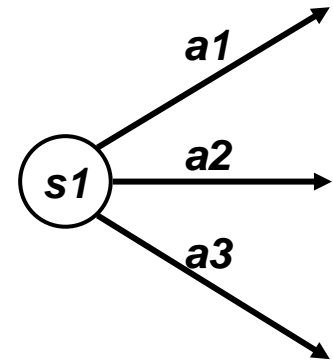
- **Acção**
- **Estado**
- **Reforço**

- Ganho / perda



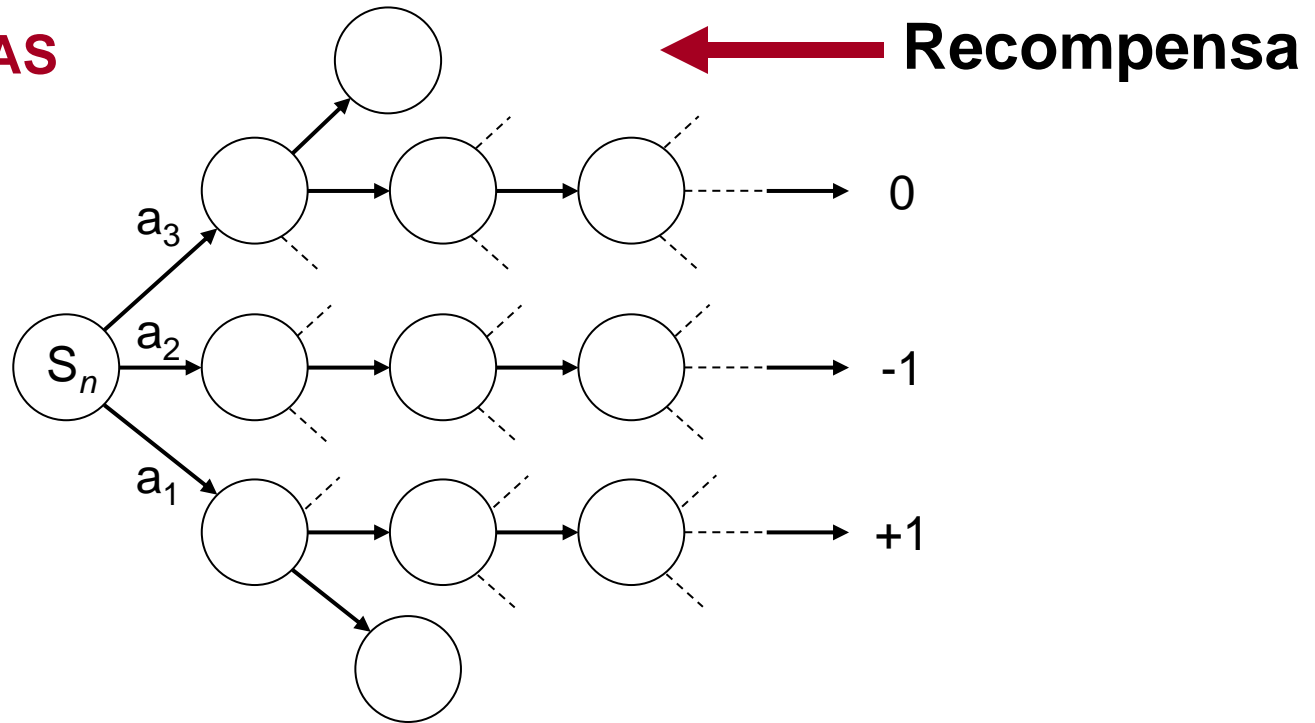
- Aprendizagem de **comportamentos**

- O que fazer
- Relação entre situações e acções



# APRENDIZAGEM POR REFORÇO

RECOMPENSAS  
DIFERIDAS

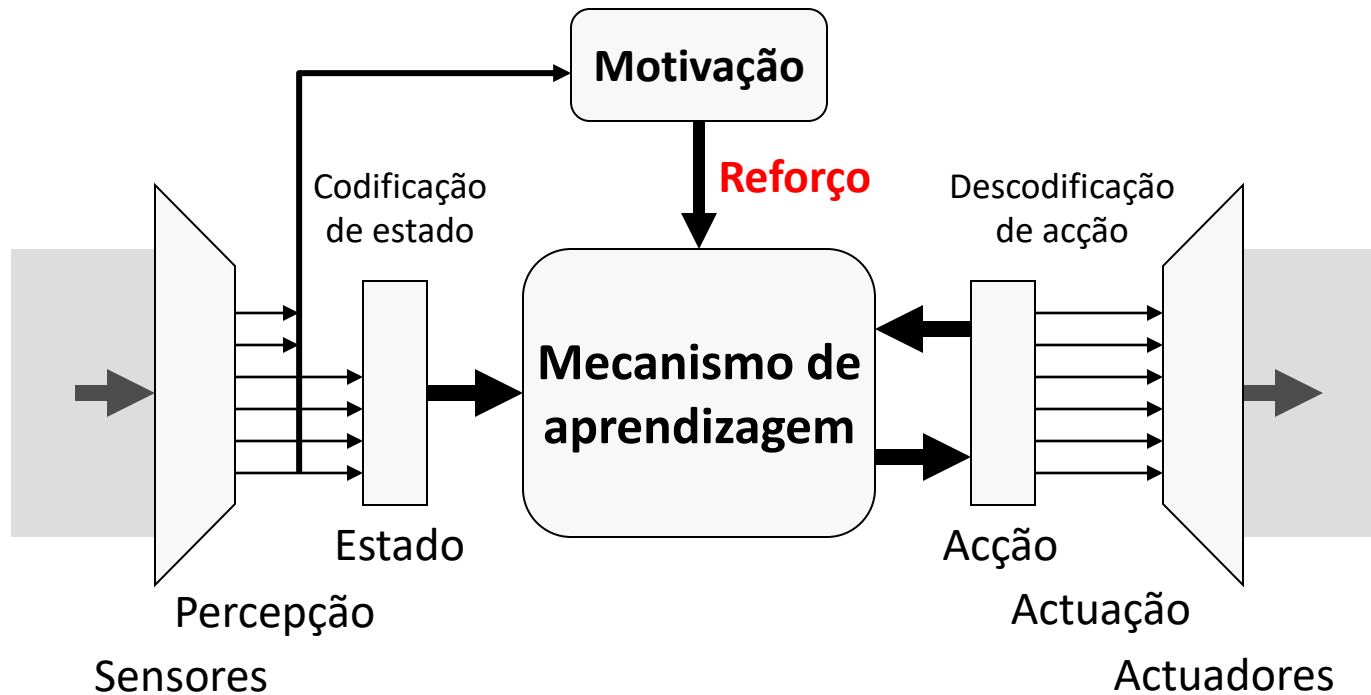


- Aprendizagem incremental a partir da experiência

$$s \rightarrow a \rightarrow r \rightarrow s' \rightarrow a' \rightarrow \dots$$

# APRENDIZAGEM POR REFORÇO

## MECANISMO DE APRENDIZAGEM

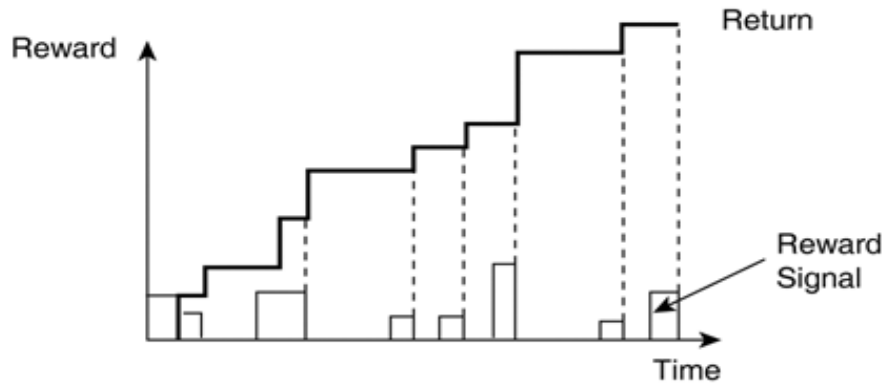


# PROCESSO DE APRENDIZAGEM

- Memória de pares **estado-acção**
  - Regras (reacções) **estímulo - resposta**
- Valorização
  - **Valor** associado ao par **estado-acção**
  - $Q(s,a)$
- Aprendizagem incremental a partir da experiência

$$s \rightarrow a \rightarrow r \rightarrow s' \rightarrow a'$$

# RECOMPENSA E RETORNO



**ACUMULAÇÃO**

**HORIZONTE  
FINITO**

$$R_t = r_{t+1} + r_{t+2} + \dots + r_{t+n} = \sum_{i=0}^n r_{t+i}$$

**HORIZONTE  
INFINITO**

$$R_t = \lim_{n \rightarrow \infty} \left( \frac{1}{n} \sum_{i=1}^n r_{t+i} \right)$$

**HORIZONTE  
INFINITO COM  
DESCONTO  
TEMPORAL**

$$R_t = r_{t+1} + \gamma \cdot r_{t+2} + \gamma^2 \cdot r_{t+3} + \gamma^3 \cdot r_{t+4} + \dots = \sum_{i=1}^{\infty} \gamma^{i-1} \cdot r_{t+i}$$

$\gamma \in [0,1]$  - Factor de desconto

# APRENDIZAGEM POR DIFERENÇA TEMPORAL

Actualização de uma **estimativa** de valor de **estado-acção** com base na **mudança** desse valor (diferença temporal) entre instantes sucessivos

$$NewEstimate \leftarrow OldEstimate + StepSize [Target - OldEstimate]$$

$$Q_{n+1} = Q_n + \alpha [R_n - Q_n]$$



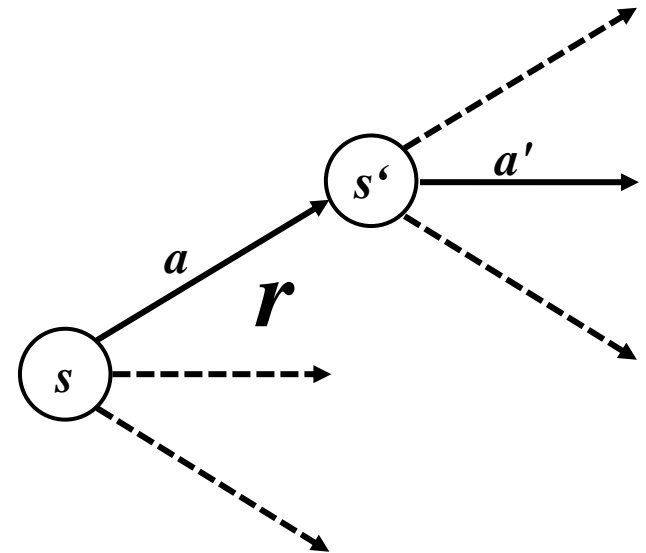
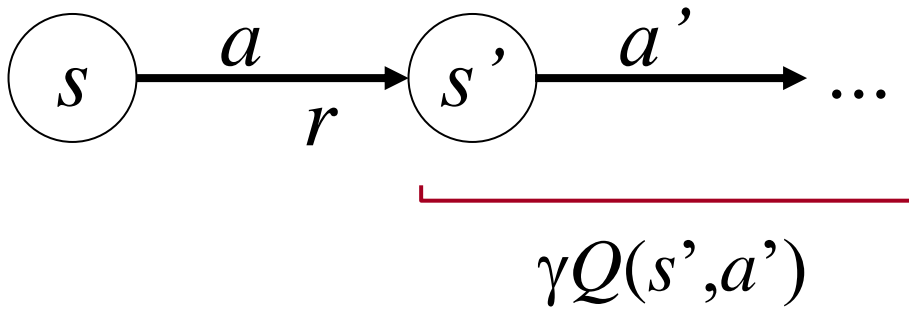
$$Q(s,a) \leftarrow Q(s,a) + \alpha [R - Q(s,a)]$$

?



# Valor de realizar uma acção $a$ num estado $s$

**Valor Estado-Acção:**  $Q(s,a)$



$$R = r + \underbrace{\gamma Q(s', a')}_{\text{Estimativa actual de } R}$$

Reforço

# APRENDIZAGEM POR DIFERENÇA TEMPORAL

Actualização de uma **estimativa** de valor de **estado-acção** com base na **mudança** desse valor (diferença temporal) entre instantes sucessivos

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R - Q(s,a)]$$



$$Q(s,a) \leftarrow Q(s,a) + \alpha[\overset{\text{Reforço}}{\downarrow} r + \underbrace{\gamma Q(s',a')}_{\substack{\uparrow \\ \text{Estimativa de } R}} - Q(s,a)]$$

$\underbrace{\hspace{15em}}_{\substack{\uparrow \\ \text{Diferença temporal}}}$

# ALGORITMO SARSA

## Controlo de aprendizagem *On-Policy*

- Utilização da mesma política para acção e aprendizagem
- Exploração de todas as acções (políticas  *$\varepsilon$ -greedy*,  *$\varepsilon$ -soft*)

Initialize  $Q(s, a)$  arbitrarily

Repeat (for each episode):

Initialize  $s$

Choose  $a$  from  $s$  using policy derived from  $Q$  (e.g.,  $\varepsilon$ -greedy)

Repeat (for each step of episode):

Take action  $a$ , observe  $r, s'$

Choose  $a'$  from  $s'$  using policy derived from  $Q$  (e.g.,  $\varepsilon$ -greedy)

$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$

$s \leftarrow s'; a \leftarrow a';$

until  $s$  is terminal

# ALGORITMO Q-LEARNING

## Controlo de aprendizagem *Off-Policy*

- Separação entre acção e aprendizagem com políticas específicas
- Aprendizagem utiliza uma política *greedy*
- Acção utiliza uma política  $\varepsilon$ -*greedy*

Initialize  $Q(s, a)$  arbitrarily

Repeat (for each episode):

Initialize  $s$

Repeat (for each step of episode):

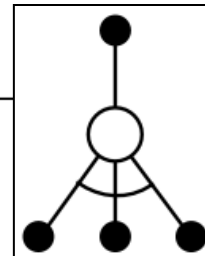
Choose  $a$  from  $s$  using policy derived from  $Q$  (e.g.,  $\varepsilon$ -greedy)

Take action  $a$ , observe  $r, s'$

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$

$s \leftarrow s'$ ;

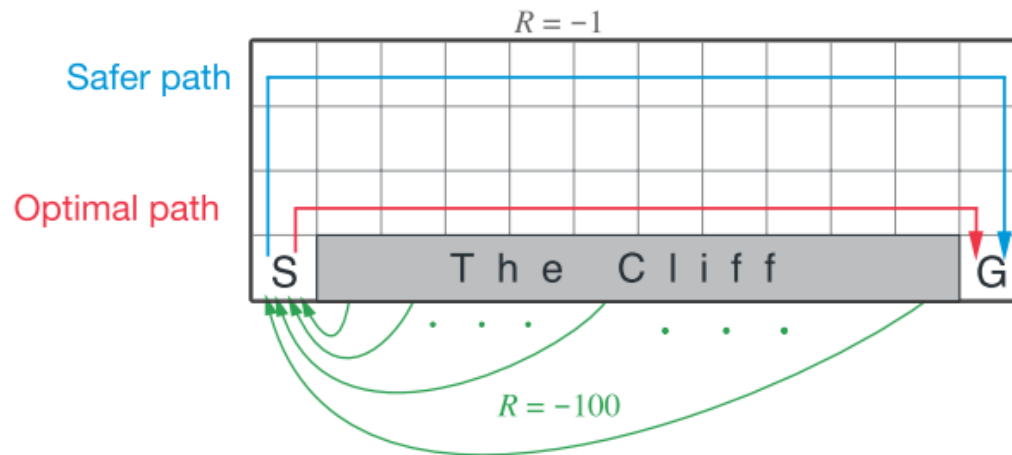
until  $s$  is terminal



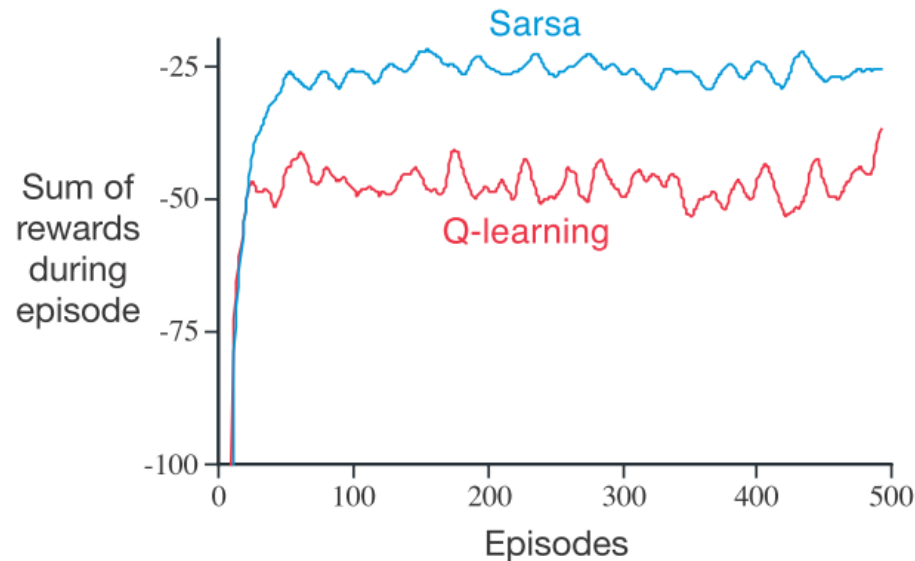
# PROCESSO DE APRENDIZAGEM

- **Dois tipos de aprendizagem**
  - **Política de selecção de acção única**  
**(On-policy)**
    - Utilização da mesma política de selecção de acção para comportamento e para propagação de valor
    - Exploração de todas as acções (e.g. política  $\epsilon$ -greedy)
  - **Políticas de selecção de acção diferenciadas**  
**(Off-policy)**
    - Utilização de diferentes políticas de selecção de acção para comportamento e para propagação de valor
    - **Optimização da função valor  $Q(s,a)$**

# SARSA vs. Q-LEARNING



$\epsilon$ -greedy,  $\epsilon = 0.1$



[Sutton & Barto, 2020]

# POLÍTICAS COMPORTAMENTAIS

## Política Comportamental

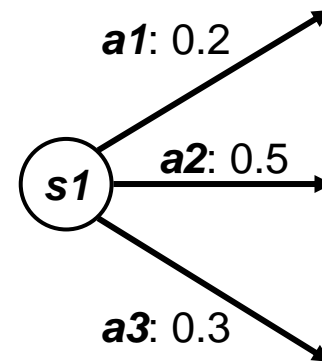
- Forma de representação do comportamento do agente
- Estratégia de acção que *define qual a acção que deve ser realizada em cada estado*

- Política **determinista**

$$\pi : S \rightarrow A(s) ; s \in S$$

- Política **não determinista**

$$\pi : S \times A(s) \rightarrow [0,1] ; s \in S$$



# POLÍTICA COMPORTAMENTAL ÓPTIMA

- Função valor de estado-acção

$$Q^{\pi}(s, a)$$

- Valor óptimo

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$$

- Política óptima

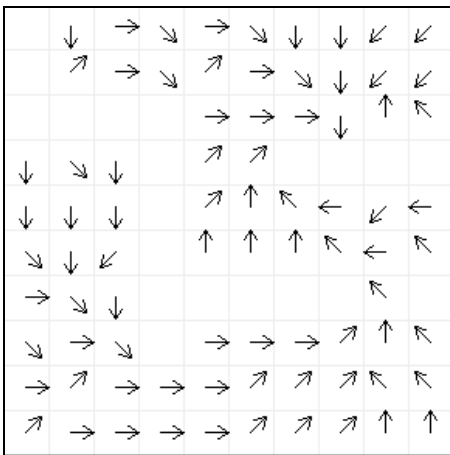
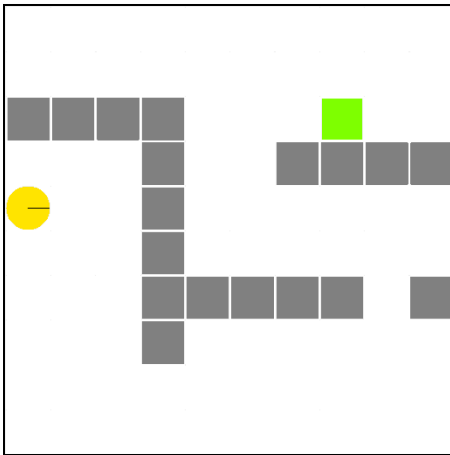
$$\pi^*(s) = \arg \max_a Q^*(s, a)$$

Política “*greedy*” em relação a  $Q^*$

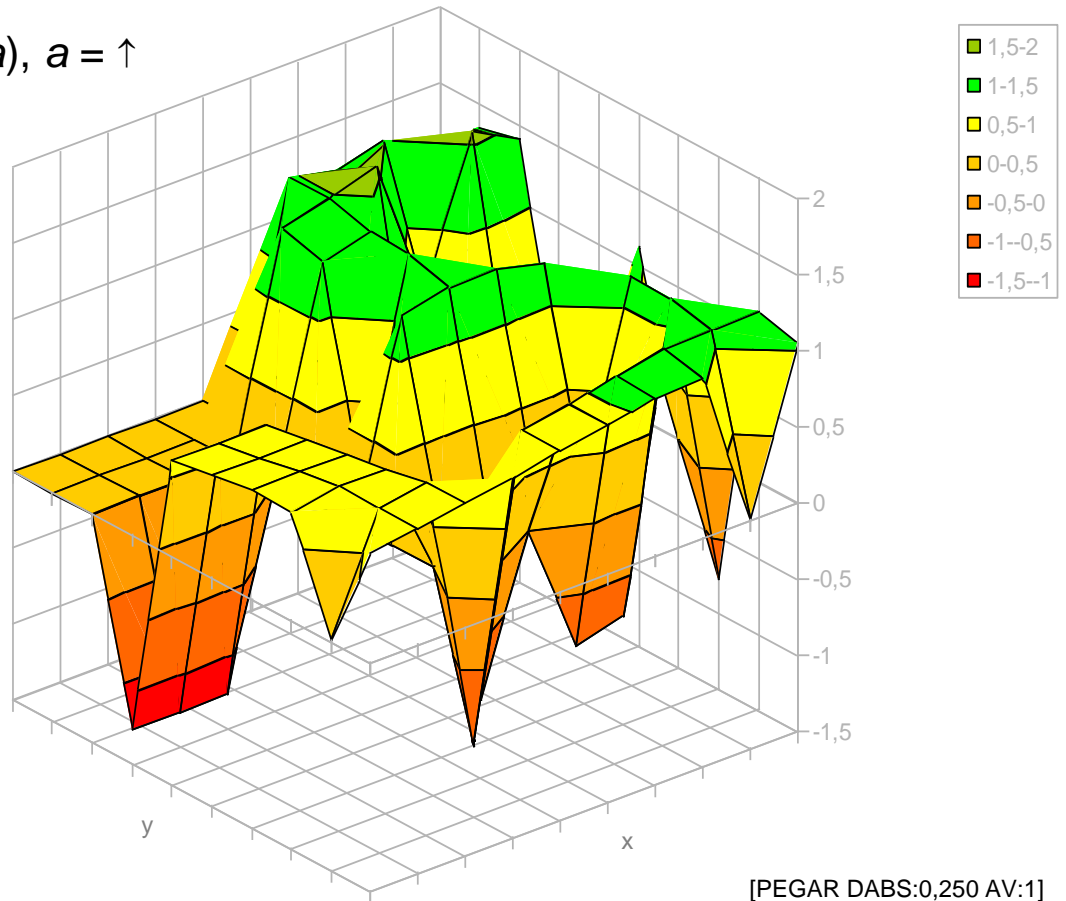


# APRENDIZAGEM POR REFORÇO

## FORMAÇÃO DE POLÍTICAS COMPORTAMENTAIS



$Q(s,a), a = \uparrow$



# DILEMA EXPLORAR / APROVEITAR

- **Para convergir para o valor óptimo**
  - Não se pode apenas explorar
  - Não se pode apenas aproveitar
- **Estratégia Sôfrega (*Greedy*)**
  - Mínimos/máximos locais
- **Nunca se pode parar de explorar**
  - Convergência assintótica
- **Deve-se progressivamente reduzir a exploração**
  - GLIE (*Greedy in the Limit of Infinite Exploration*)

# ALGORITMO Q-LEARNING

- **Propriedades**

- Os valores da matriz  $Q$  **convergem** no limite se:
  - Se cada par estado-acção  $(s,a)$  for visitado um número ilimitado de vezes
  - O parâmetro  $\alpha$  tender para 0 no limite
- No limite a estratégia  $\varepsilon$ -greedy de aproveitamento de  $Q(s,a)$  converge para a **política óptima**

- **Estes requisitos podem ser satisfeitos através de:**

- $\alpha(s,a) \approx 1/k$ 
  - Sendo  $k$  o número de vezes que a acção  $a$  foi seleccionada em  $s$
- Estratégia de selecção de acção  $\varepsilon$ -greedy com  $\varepsilon \approx 1/t$ 
  - Sendo  $t$  função do tempo ou do número de tentativas de aprendizagem

# APRENDIZAGEM POR REFORÇO

- **PROBLEMAS**

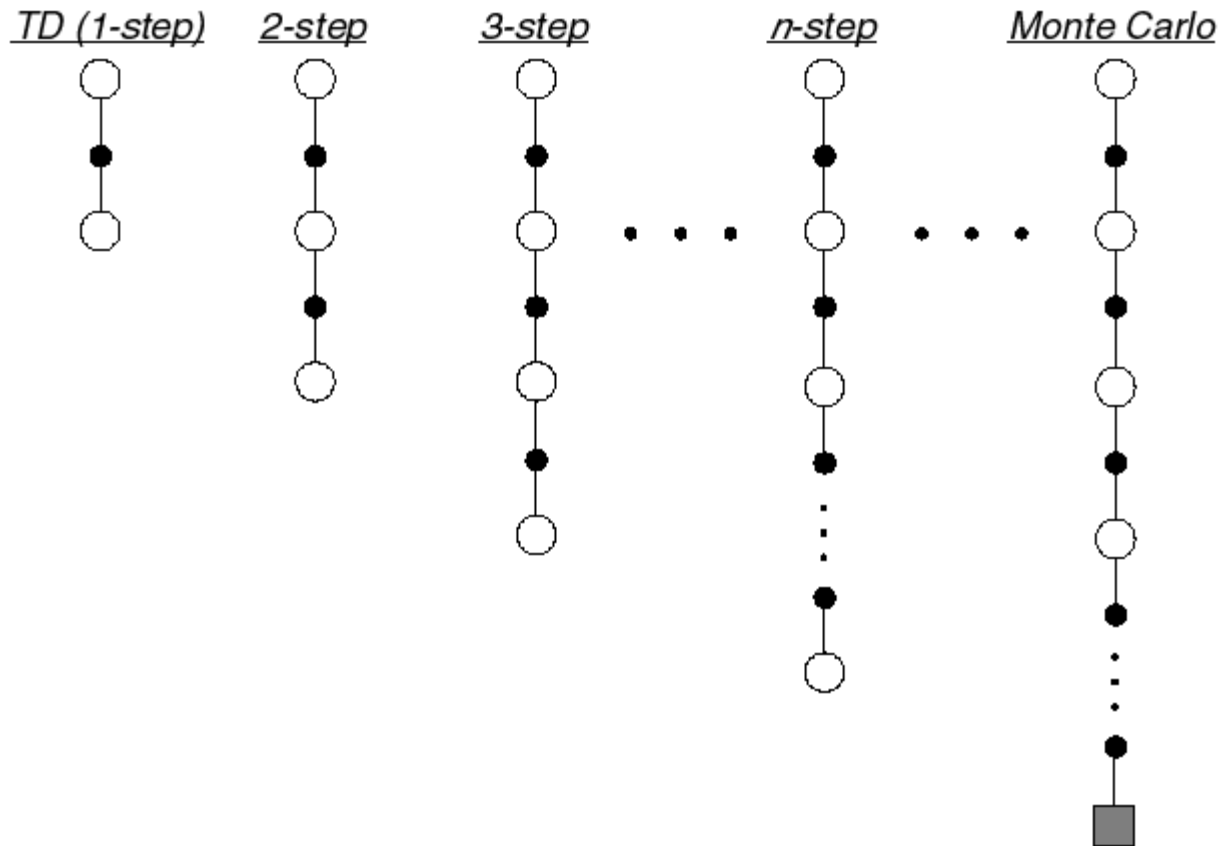
- Complexidade dos espaços de estados
- Tempo de convergência

- **SOLUÇÕES**

- Memória de experiência
- Utilização de modelos do mundo
- Generalização / Abstracção
  - Redes neuronais
  - Modelos hierárquicos
- Arquitecturas híbridas

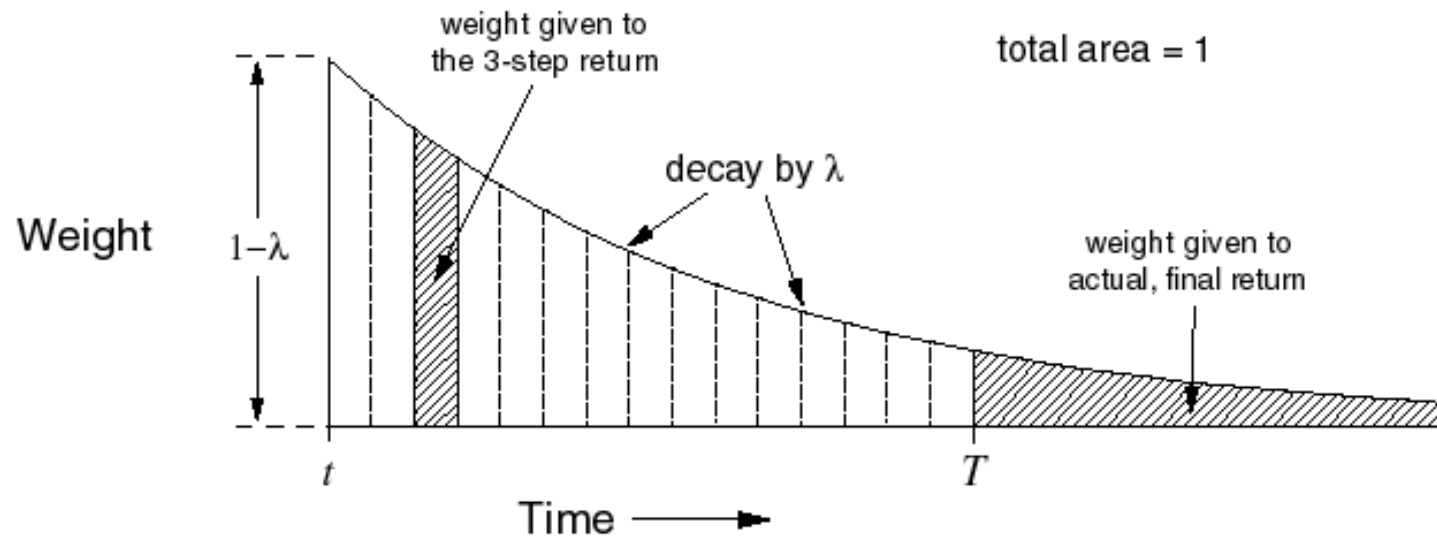
# FORMAS DE ACELERAR A APRENDIZAGEM

Recordar resultado de acções realizadas



# FORMAS DE ACELERAR A APRENDIZAGEM

$\lambda \in [0,1]$  - Return Weighting Function

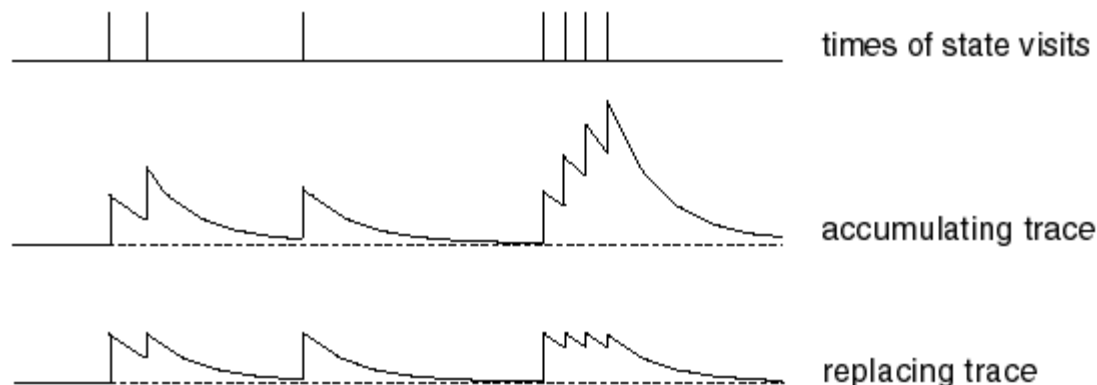


# PROPAGAÇÃO DE INFLUÊNCIA

- RASTO DE ELIGIBILIDADE (*Eligibility Traces*)  $e_t(s) \in \mathbb{R}^+$

- *Accumulating trace* 
$$e_t(s) = \begin{cases} \gamma \lambda e_{t-1}(s) & \text{if } s \neq s_t \\ \gamma \lambda e_{t-1}(s) + 1 & \text{if } s = s_t \end{cases}$$

- *Replacing trace* 
$$e_t(s) = \begin{cases} \gamma \lambda e_{t-1}(s) & \text{if } s \neq s_t \\ 1 & \text{if } s = s_t \end{cases}$$

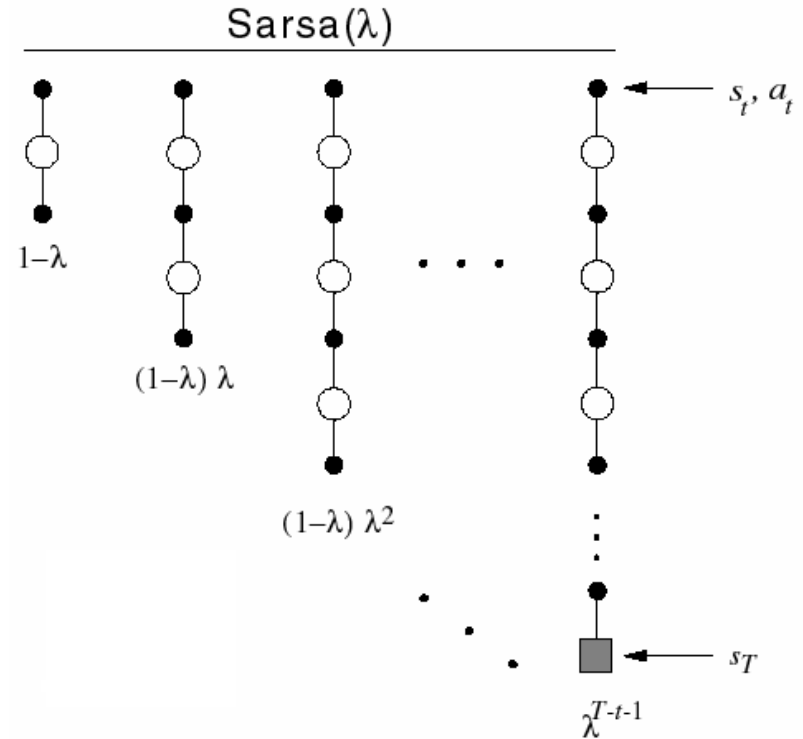


# SARSA( $\lambda$ )

$e(s,a)$  – Rasto de elegibilidade

$$\delta_t = r_{t+1} + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)$$

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha \delta_t e_t(s, a)$$





# SARSA( $\lambda$ )

Initialize  $Q(s,a)$  arbitrarily

Repeat (for each episode) :

$e(s,a) = 0$ , for all  $s,a$

Initialize  $s,a$

Repeat (for each step of episode) :

Take action  $a$ , observe  $r,s'$

Choose  $a'$  from  $s'$  using policy derived from  $Q$  (e.g.  $\epsilon$ -greedy)

$\delta \leftarrow r + \gamma Q(s',a') - Q(s,a)$

$e(s,a) \leftarrow e(s,a) + 1$

For all  $s,a$  :

$Q(s,a) \leftarrow Q(s,a) + \alpha \delta e(s,a)$

$e(s,a) \leftarrow \gamma \lambda e(s,a)$

$s \leftarrow s'; a \leftarrow a'$

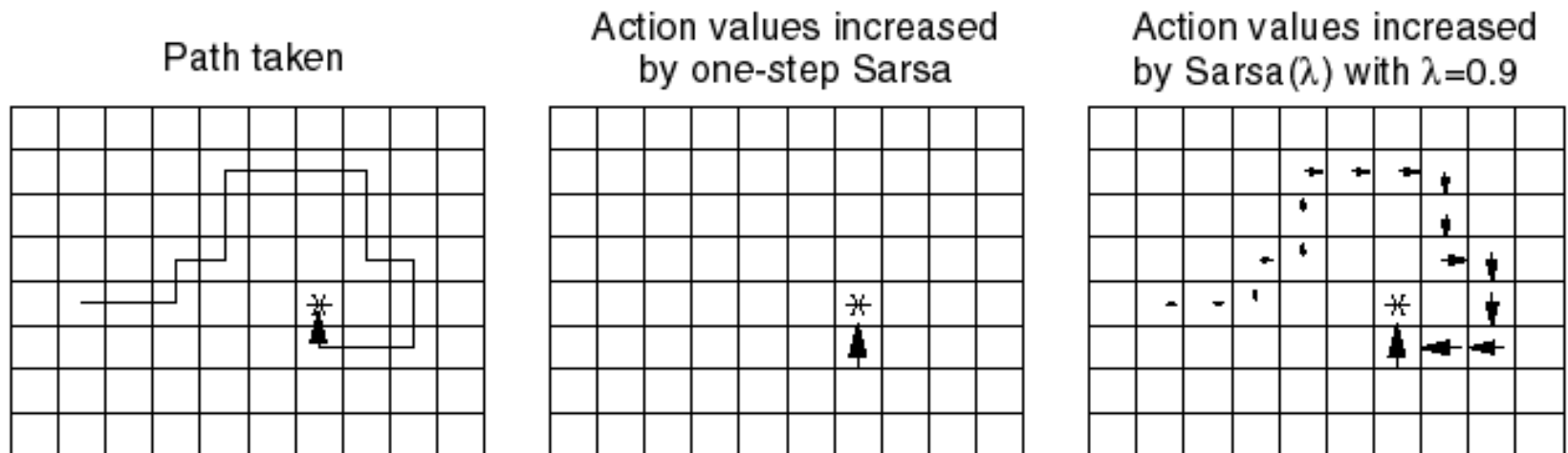
Until  $s$  is terminal

# SARSA( $\lambda$ )

## Rastros de elegibilidade podem acelerar a aprendizagem

- Mais informação disponível
- Pode **não** ser a melhor!

## Exemplo:



# REFERÊNCIAS

[Sutton & Barto, 2020]

R. Sutton, A. Barto, “Reinforcement Learning: An Introduction”, 2<sup>nd</sup> Edition, MIT Press, 2020

[Poole & Mackworth, 2010]

D. Poole, A. Mackworth, Artificial Intelligence: Foundations of Computational Agents, Cambridge University Press, 2010

[Barnard, 2003]

C. Barnard, “Animal Behaviour: Mechanism, Development, Ecology and Evolution”, Prentice Hall, 2003