

**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

# Aprendizagem por Reforço Reinforcement Learning

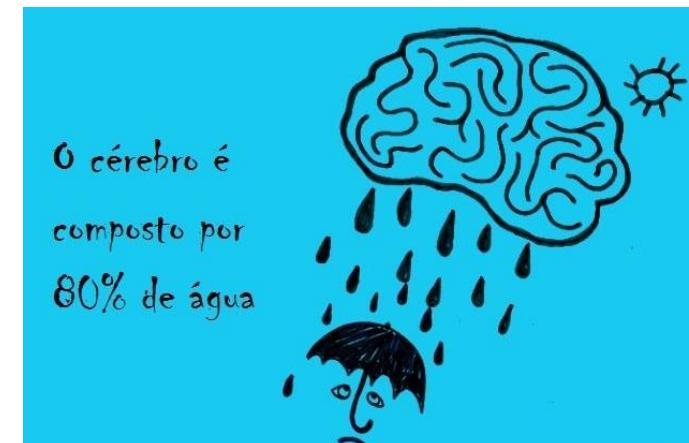
Aprendizagem e Extração de Conhecimento  
Perfil Sistemas Inteligentes @ MiEI/4º – 1º Semestre

Cesar Analide, Filipe Gonçalves

## Definição

- Aprendizagem (<https://pt.wikipedia.org/wiki/Aprendizagem>):
  - Ato ou efeito de aprender; Tempo durante o qual se aprende; Experiência que tem quem aprendeu.
- Reforço ([https://pt.wikipedia.org/wiki/Reforço\\_\(educação\)](https://pt.wikipedia.org/wiki/Reforço_(educação)))
  - Ato ou efeito de reforçar; Aquilo que reforça; Aumento de força; Auxílio; Tropas auxiliares; (...)
- Aprendizagem por Reforço:
  - Aprendizagem com estímulo

*in Dicionário Priberam da Língua Portuguesa*



(recolha de opiniões em aula)



■ “A VISITOR to our planet might be puzzled about the role of computers in our technology. On the one hand, he would read and hear all about wonderful “mechanical brains” baffling their creators with prodigious intellectual performance. And he (or it) would be warned that these machines must be restrained, lest they overwhelm us by might, persuasion, or even by the revelation of truths too terrible to be borne. On the other hand, our visitor would find the machines being denounced, on all sides, for their slavish obedience, unimaginative literal interpretations, and incapacity for innovation or initiative; in short, for their inhuman dullness.”

## Steps Toward Artificial Intelligence\*

MARVIN MINSKY†, MEMBER, IRE

# Aprendizagem por Reforço

The work toward attaining "artificial intelligence" is the center of considerable computer research, design, and application. The field is in its starting transient, characterized by many varied and independent efforts. Marvin Minsky has been requested to draw this work together into a coherent summary, supplement it with appropriate explanatory or theoretical noncomputer information, and introduce his assessment of the state-of-the-art. This paper emphasizes the class of activities in which a general purpose computer, complete with a library of basic programs, is further programmed to perform operations leading to ever higher-level information processing functions such as learning and problem solving. This informative article will be of real interest to both the general PROCEEDINGS reader and the computer specialist.—*The Guest Editor*

"

**Summary**—The problems of heuristic programming—of making computers solve really difficult problems—are divided into five main areas: Search, Pattern-Recognition, Learning, Planning, and Induction.

A computer can do, in a sense, only what it is told to do. But even when we do not know how to solve a certain problem, we may program a machine (computer) to *Search* through some large space of solution attempts. Unfortunately, this usually leads to an enormously inefficient process. With *Pattern-Recognition* techniques, efficiency can often be improved, by restricting the application of the machine's methods to appropriate problems. *Pattern-Recognition*, together with *Learning*, can be used to exploit generalizations based on accumulated experience, further reducing search. By analyzing the situation, using *Planning* methods, we may obtain a fundamental improvement by replacing the given search with a much smaller, more appropriate exploration. To manage broad classes of problems, machines will need to construct models of their environments, using some scheme for *Induction*.

Wherever appropriate, the discussion is supported by extensive citation of the literature and by descriptions of a few of the most successful heuristic (problem-solving) programs constructed to date.

### INTRODUCTION

A VISITOR to our planet might be puzzled about the role of computers in our technology. On the one hand, he would read and hear all about wonderful "mechanical brains" baffling their creators with prodigious intellectual performance. And he (or it) would be warned that these machines must be restrained, lest they overwhelm us by might, persuasion, or even by the revelation of truths too terrible to be borne. On the other hand, our visitor would find the machines being denounced, on all sides, for their slavish obedience, unimaginative literal interpretations, and incapacity for innovation or initiative; in short, for their inhuman dullness.

Our visitor might remain puzzled if he set out to find, and judge for himself, these monsters. For he would

\* Received by the IRE, October 24, 1960. The author's work summarized here—which was done at Lincoln Lab., a center for research operated by M.I.T. at Lexington, Mass., a center for support of the U. S. Army, Navy, and Air Force under Air Force Contract AF 19(604)-5200; and at the Res. Lab. of Electronics, M.I.T., Cambridge, Mass., which is supported in part by the U. S. Army Signal Corps, the Ford Office of Scientific Research, and the ONR—is based on earlier work done by the author as a Junior Fellow of the Society of Fellows, Harvard Univ., Cambridge.

† Dept. of Mathematics and Computation Center, Res. Lab. of Electronics, M.I.T., Cambridge, Mass.

find only a few machines (mostly "general-purpose" computers, programmed for the moment to behave according to some specification) doing things that might claim any real intellectual status. Some would be proving mathematical theorems of rather undistinguished character. A few machines might be playing certain games, occasionally defeating their designers. Some might be distinguishing between hand-printed letters. Is this enough to justify so much interest, let alone deep concern? I believe that it is; that we are on the threshold of an era that will be strongly influenced, and quite possibly dominated, by intelligent problem-solving machines. But our purpose is not to guess about what the future may bring; it is only to try to describe and explain what seem now to be our first steps toward the construction of "artificial intelligence."

Along with the development of general-purpose computers, the past few years have seen an increase in effort toward the discovery and mechanization of problem-solving processes. Quite a number of papers have appeared describing theories or actual computer programs concerned with game-playing, theorem-proving, pattern-recognition, and other domains which would seem to require some intelligence. The literature does not include any general discussion of the outstanding problems of this field.

In this article, an attempt will be made to separate out, analyze, and find the relations between some of these problems. Analysis will be supported with enough examples from the literature to serve the introductory function of a review article, but there remains much relevant work not described here. This paper is highly compressed, and therefore, cannot begin to discuss all these matters in the available space.

There is, of course, no generally accepted theory of "intelligence"; the analysis is our own and may be controversial. We regret that we cannot give full personal acknowledgments here—suffice it to say that we have discussed these matters with almost every one of the cited authors.

It is convenient to divide the problems into five main areas: Search, Pattern-Recognition, Learning, Planning, and Induction; these comprise the main divisions



Synthetic Intelligence Lab



**Aprendizagem  
com Supervisão**  
*Supervised Learning*

**Aprendizagem  
por Reforço**  
*Reinforcement Learning*

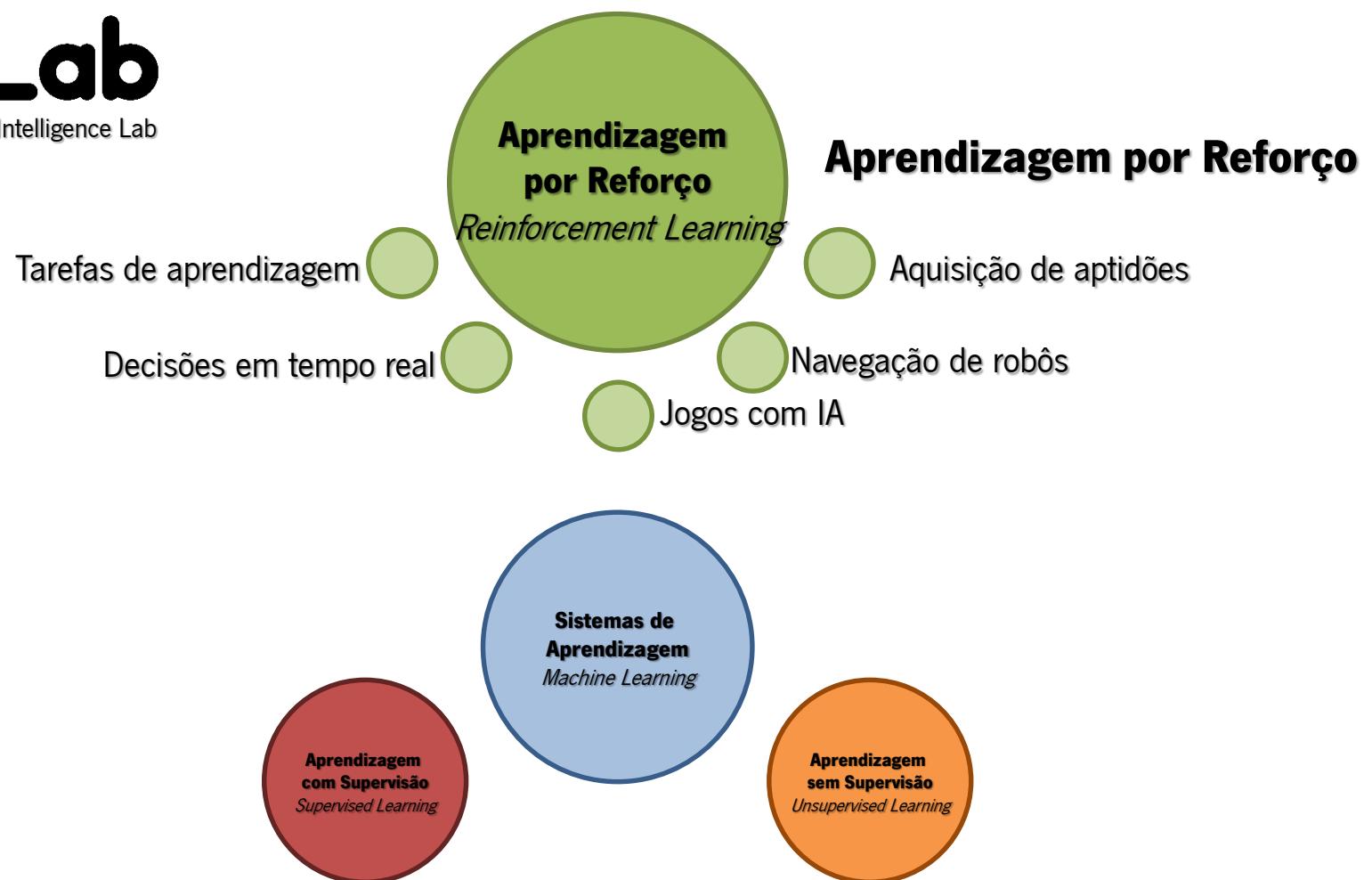
**Aprendizagem por Reforço**

**Sistemas de  
Aprendizagem**  
*Machine Learning*

**Aprendizagem  
sem Supervisão**  
*Unsupervised Learning*



Synthetic Intelligence Lab

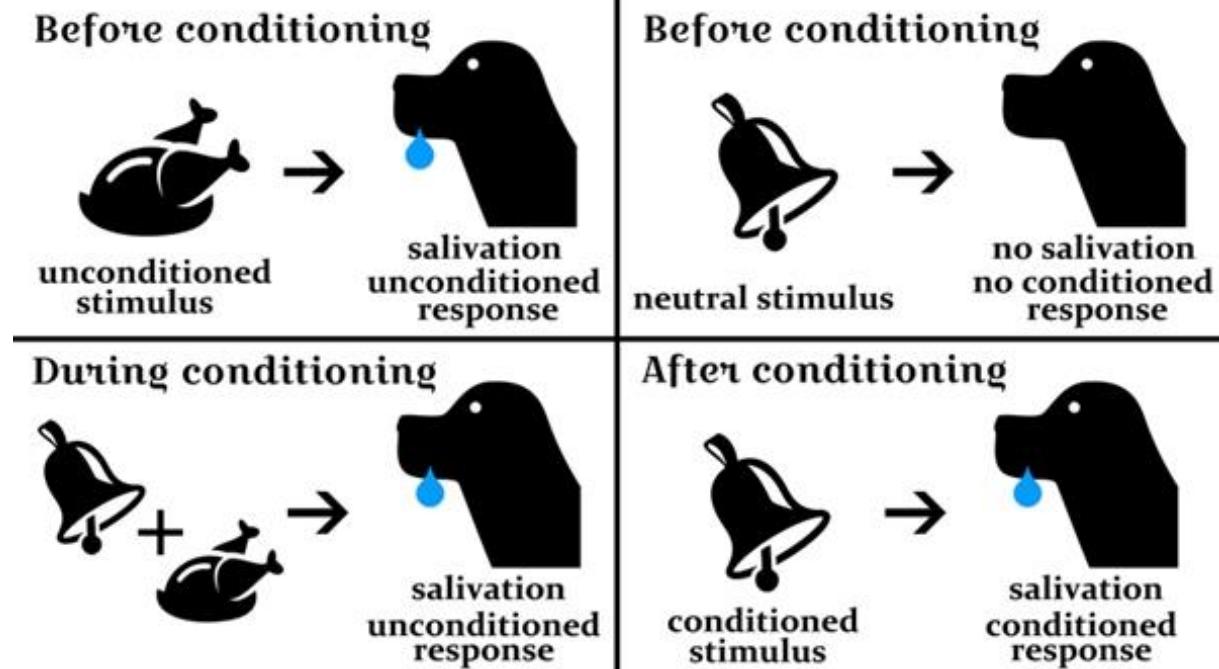


## Aprendizagem por Reforço

- A primeira referência ao termo “Reinforcement Learning” (Aprendizagem por Reforço) surge no artigo de Marvin Minsky “Steps towards artificial intelligence”, em 1961;
- Ainda em 1959, Arthur Samuel refere-se a mecanismos de aprendizagem a que chama “Rote Learning” (Aprendizagem por Rotina), aplicados a algoritmos para jogar “damas”;
- Em 1998, Sutton & Barto, refletem sobre a adequação de sistemas de aprendizagem que recompensam “boas” ações, patentes em experiências com animais.



- Inteligência Artificial;
- Sistemas de Aprendizagem (machine learning);
- Psicologia animal;
- Teoria de Controlo.





Synthetic Intelligence Lab

Inspiração

- 
- Inteligência Artificial;
  - Sistemas de Aprendizagem (machine learning);
  - Psicologia animal;
  - Teoria de Controlo.

## Robot Motor Skill Coordination with EM-based Reinforcement Learning

Petar Kormushev, Sylvain Calinon,  
and Darwin G. Caldwell

Italian Institute of Technology

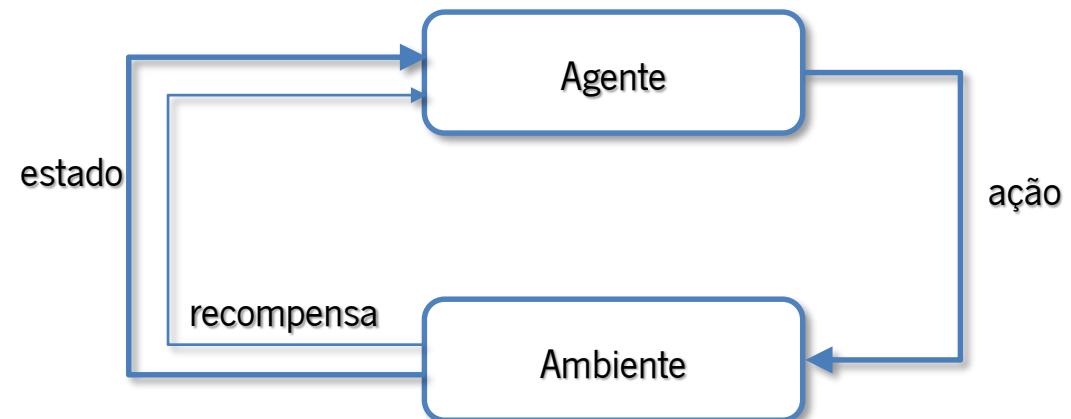
Response

stimulus

Response

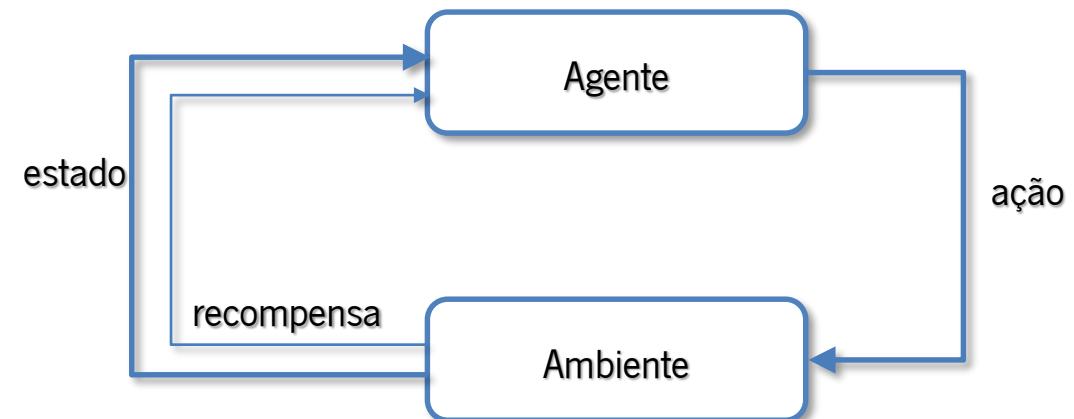
## Aprendizagem por Reforço

- Aprendizagem por Reforço é aprender “o que fazer”, de modo a maximizar um sinal **numérico** de recompensa:
  - O aprendente não é informado de quais as ações que deve executar;
  - O aprendente deve descobrir que ações garantem maior retorno, experimentando-as.

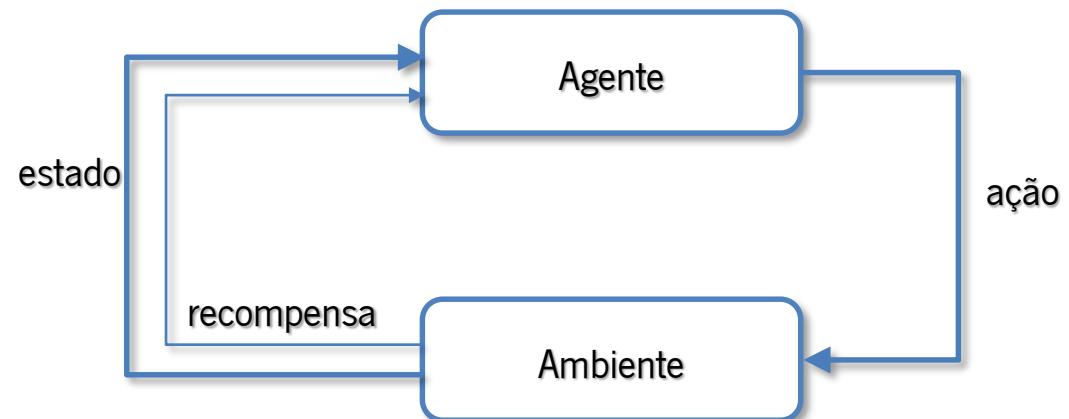


## Aprendizagem por Reforço

- Aprendizagem por Reforço é aprender “o que fazer”, de modo a maximizar um sinal **numérico** de recompensa:
  - O aprendente não é informado de quais as ações que deve executar;
  - O aprendente deve descobrir que ações garantem maior retorno, experimentando-as.
- São características essenciais:
  - Tentativa-erro (*trial-and-error*);
  - Recompensa adiada (*delayed reward*).



- Aprendizagem por Reforço é aprender “o que fazer”, de modo a maximizar um sinal **numérico** de recompensa:
  - O aprendente não é informado de quais as ações que deve executar;
  - O aprendente deve descobrir que ações garantem maior retorno, experimentando-as.
- São características essenciais:
  - Tentativa-erro (*trial-and-error*);
  - Recompensa adiada (*delayed reward*).
- São características complementares:
  - Tempo;
  - Há aprendente, mas não há ensinante!



## Definição de aprender

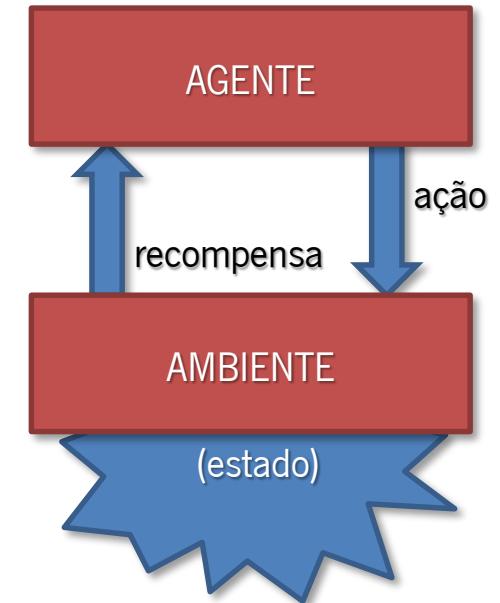
- Aprendizagem por Reforço não é definida pela caracterização de métodos de aprendizagem;  
(não programamos algoritmos para aprender)
- Aprendizagem por Reforço é definida pela caracterização do problema de aprendizagem;  
(programamos as características do problema sobre o qual aprender)
  
- Aprendizagem por Reforço não é olhar para as experiências passadas para tomar decisões (RBC, RNA);
- Aprendizagem por Reforço é olhar para o estado atual e decidir o que fazer, prevendo o futuro esperado;



## Como aprender

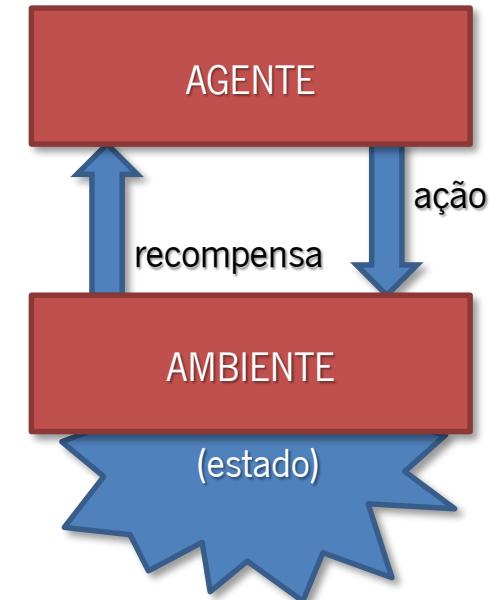
- Aprendizagem por Reforço responde à questão:
  - “Como pode, um agente que sente o ambiente e age sobre ele, aprender a escolher as melhores ações de modo a atingir os seus objetivos?”

(Tom Mitchell, “Machine Learning”, 2011)
- Procedimento:
  - Agente executa ação sobre o ambiente;
  - Ambiente atribui recompensa/penalização;
  - Agente calcula a conveniência da ação;
  - Agente escolhe para executar nova ação sobre o novo ambiente.
- De cada vez que o agente realiza uma ação sobre o ambiente, é-lhe atribuída uma recompensa ou uma penalização demonstrativa da conveniência da sua ação.



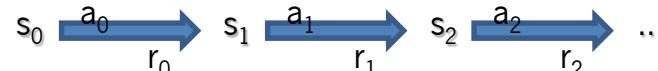
## Procedimento

- O agente...
  - existe num ambiente com um conjunto de estados  $S$ ;
  - pode executar qualquer ação de um conjunto de ações  $A$ ;
  - quando executa uma ação  $a_t$  num estado  $s_t$ , o agente recebe uma recompensa  $r_t$  que indica o valor imediato da transição estado-ação;



## Procedimento

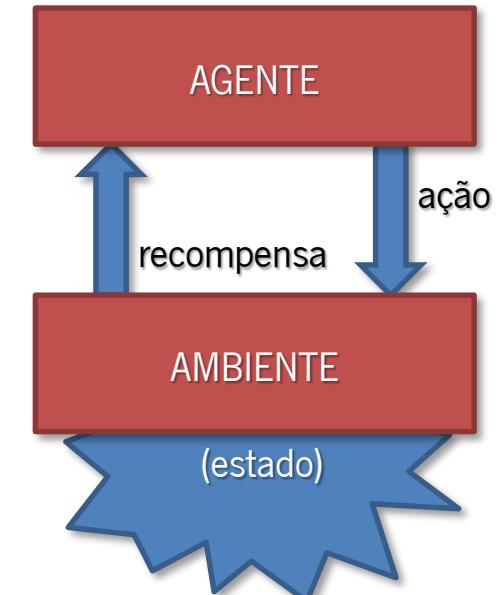
- O ciclo de vida produz uma sequência de estados  $s_i$ , ações  $a_i$  e recompensas imediatas  $r_i$ :



- A tarefa do agente é aprender uma política de controlo,  $\pi: S \rightarrow A$ , que maximiza a soma (esperada) das recompensas, sendo que as recompensas futuras são descontadas exponencialmente pelo seu atraso:

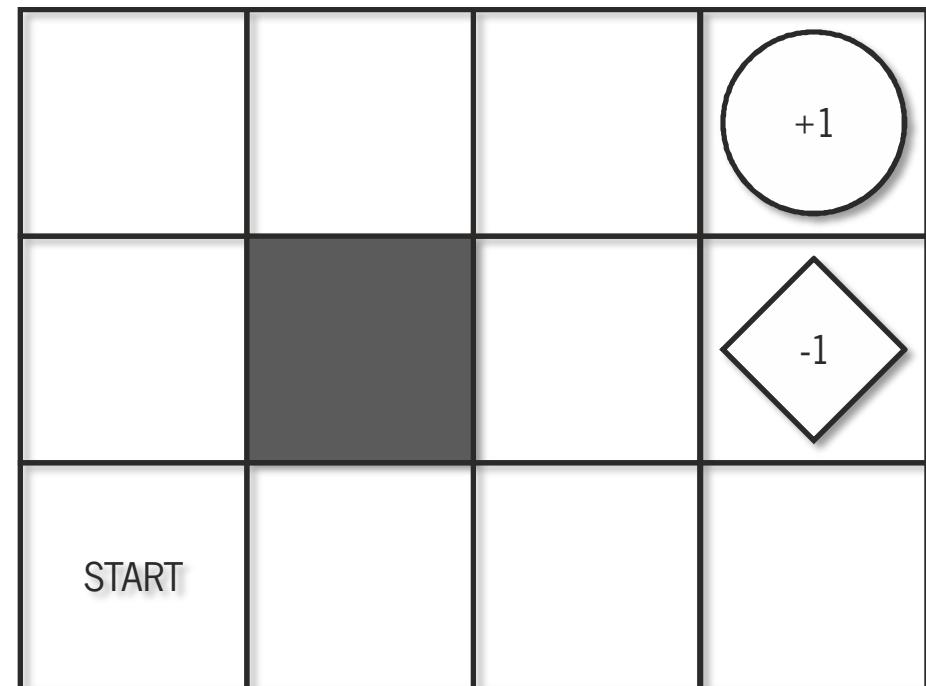
$$r_0 + \gamma^1 r_1 + \gamma^2 r_2 + \dots, \text{ com } 0 \leq \gamma < 1$$

$$\sum_{i=0}^n \gamma^i r_i$$



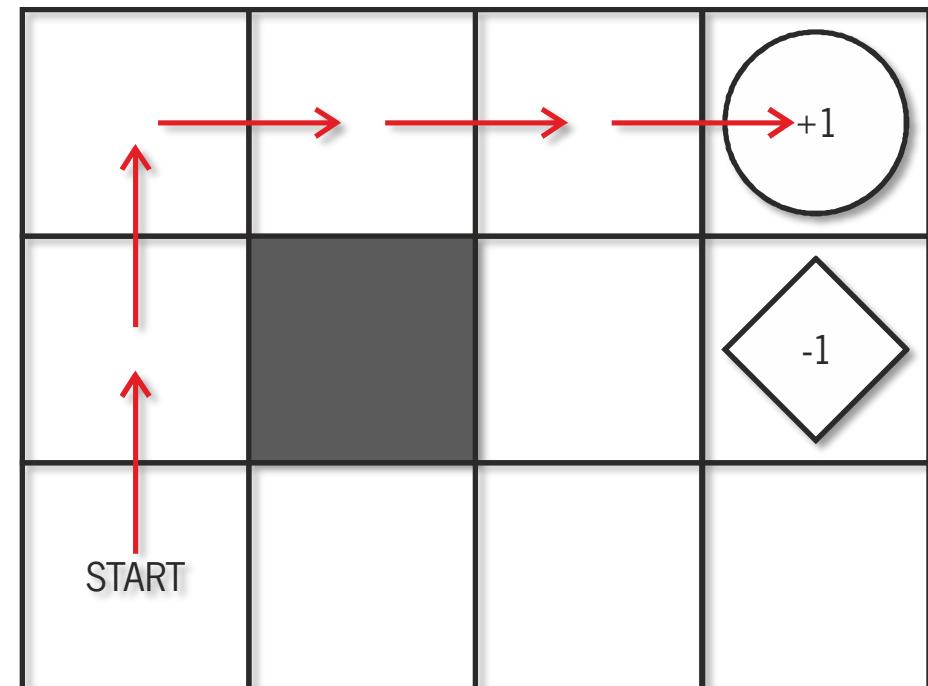
## Aprendizagem por Reforço

- Custo de movimentação: -0.04
- Forma de movimentação:
  - Sentido desejado;
  - À esquerda;
  - À direita;
  - Sentido contrário.



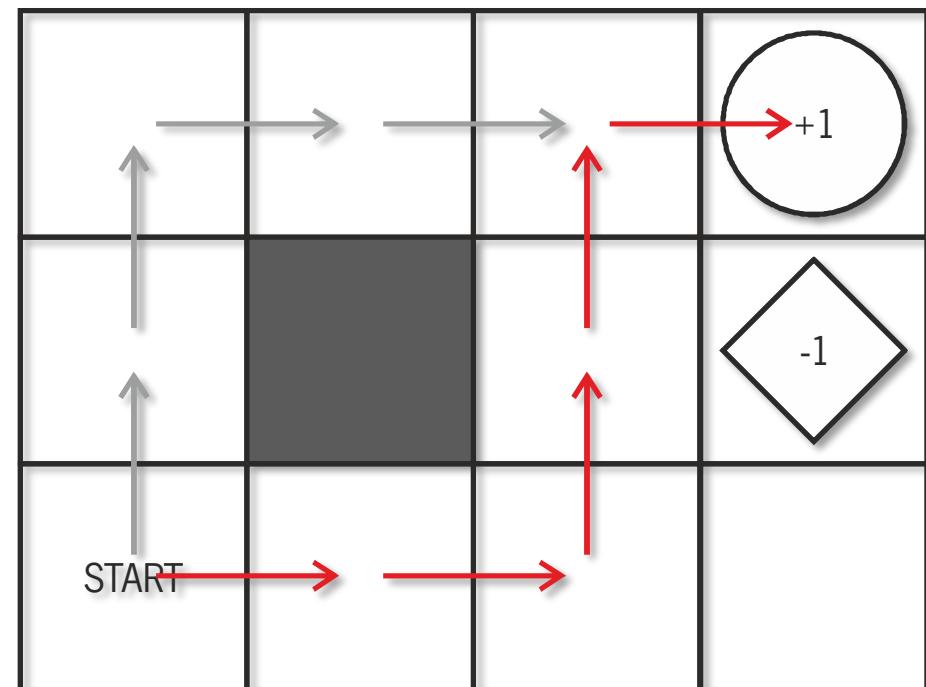
## Aprendizagem por Reforço

- Custo de movimentação: -0,04
  - $5 \times (-0,04) + 1 = 0,80$



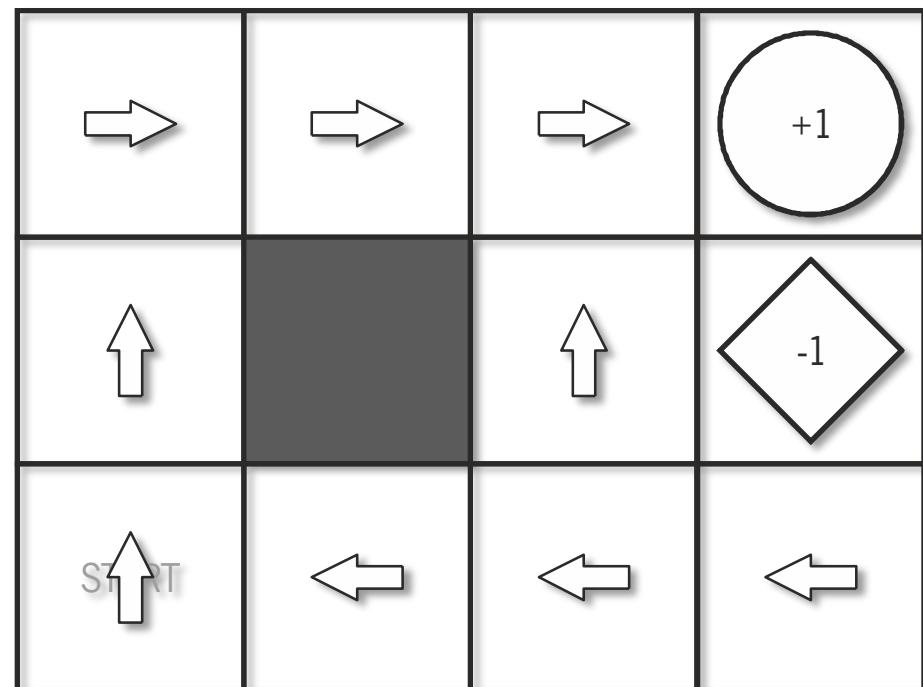
## Aprendizagem por Reforço

- Custo de movimentação: -0,04
  - $5 \times (-0,04) + 1 = 0,80$
  - $5 \times (-0,04) + 1 = 0,80$



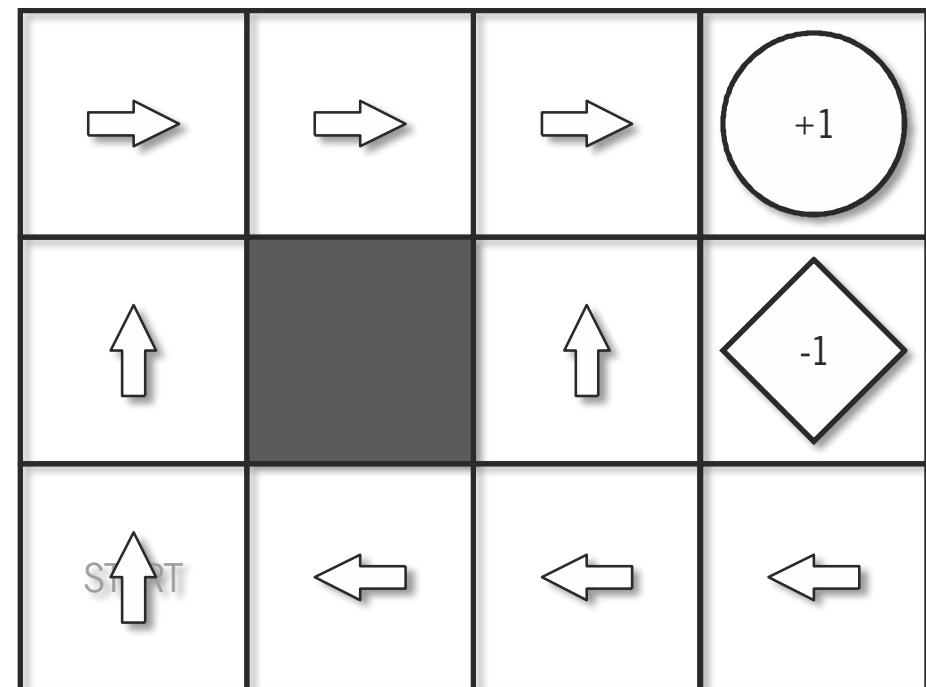
## Aprendizagem por Reforço

- Custo de movimentação: -0,04
- Forma de movimentação:
  - Desejada: 80%;
  - Esquerda: 10%;
  - Direita: 10%;
  - Contrária: 0%.



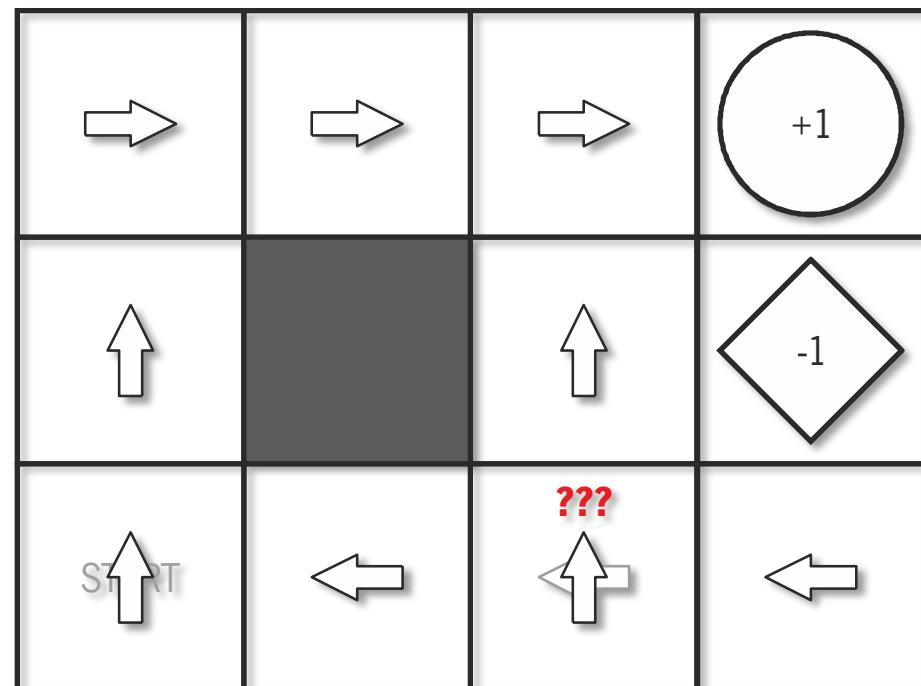
- Custo de movimentação: -0,04
- Forma de movimentação:
  - Desejada: 80%;
  - Esquerda: 10%;
  - Direita: 10%;
  - Contrária: 0%.
- Esta é a política ótima de movimentação.

## Aprendizagem por Reforço



## Aprendizagem por Reforço

- Custo de movimentação: -0,04
- Forma de movimentação:
  - Desejada: 80%;
  - Esquerda: 10%;
  - Direita: 10%;
  - Contrária: 0%.
- Por que razão esta **não é** a política ótima de movimentação?
- O equilíbrio entre velocidade e segurança depende da fiabilidade das ações e do custo de movimentação.



## Aprendizagem por Reforço

- Custo de movimentação: -0,04
- Forma de movimentação:
  - Desejada: 80%;
  - Esquerda: 10%;
  - Direita: 10%;
  - Contrária: 0%.
- Custo de movimentação, assumindo a política ótima;

			+1
0,812	0,868	0,918	
			-1
0,762		0,660	
			
0,705	0,655	0,611	0,388

## Aprendizagem por Reforço

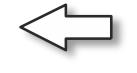
- Custo de movimentação: -0,04
- Forma de movimentação:
  - Desejada: 80%;
  - Esquerda: 10%;
  - Direita: 10%;
  - Contrária: 0%.
- $$0,80 \times (1 + (-0,04)) +$$

$$+ 0,10 \times (0,660 + (-0,04)) +$$

$$+ 0,10 \times (0,918 + (-0,04)) +$$

$$+ 0,00 \times (0,868 + (-0,04)) =$$

$$= ???$$

			+1
0,812	0,868	0,918	
			-1
0,762		0,660	
			
0,705	0,655	0,611	0,388

## Aprendizagem por Reforço

- Custo de movimentação: -0,04
- Forma de movimentação:
  - 10% : 80% : 10%
- O valor em cada ponto do ambiente representa o valor de o agente se movimentar no sentido estabelecido na política ótima;

			+1
			-1
			

0,812      0,868      0,918  
0,762      0,660  
0,705      0,655      0,611      0,388

## Aprendizagem por Reforço

- Custo de movimentação: -0,04
- Forma de movimentação:
  - 10% : 80% : 10%
- Cada valor, em cada ponto do ambiente é designado “valor Q”;
- Um “valor Q” é o valor de tomar uma dada ação, num determinado estado, seguindo um certa política;
- Como se calculam os “valor Q”?

<del>0,812</del>	<del>0,868</del>	<del>0,881 0,812 0,918 0,675</del>	+1
<del>0,762</del>		<del>0,660 0,641 -0,687 0,415</del>	-1
<del>0,705</del>	<del>0,655</del>	<del>0,611</del>	<del>-0,740 0,388 0,209 0,370</del>

- Temporal Difference Learning:

- O agente começa por assumir que todos os estados e todas as ações têm um valor inicial de 0 (zero);
  - O agente atualiza os valores calculando a diferença entre o valor esperado e o valor encontrado.

- Q Learning:

- $Q(s_t, a_t)$  : valor de tomar uma ação  $a_t$  num estado  $s_t$ ;
  - $r_{t+1}$  : recompensa imediata;
  - $\alpha$  : taxa de aprendizagem,  $0 < \alpha \ll 1$ ;
    - Proporção usada para atualizar o valor de utilidade após cada ação;
  - $\gamma$  : fator de desconto ,  $0 \ll \gamma < 1$ ;
    - Encoraja o agente a preferir recompensas imediatas a tardias;
  - $$Q(s_t, a_t) = Q(s_t, a_t) + \alpha [ r_{t+1} + \gamma \times \text{MÁX}_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) ]$$

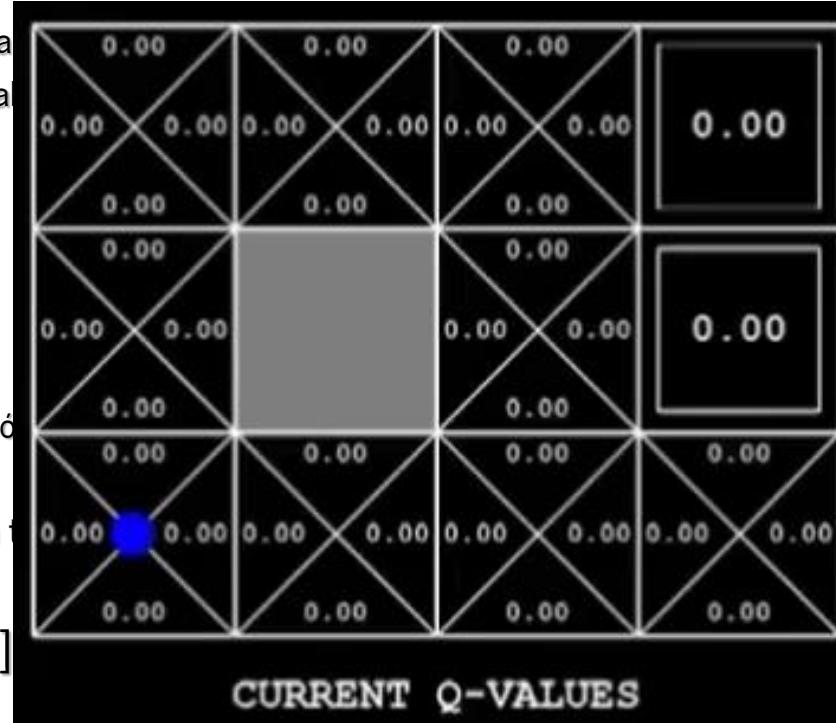
## Cálculo de Q-value

- Temporal Difference Learning:

- O agente começa por assumir que todos os estados e todas as ações têm um valor de utilidade igual a zero;
- O agente atualiza os valores calculando a diferença entre o valor real da ação e o valor estimado.

- Q Learning:

- $Q(s_t, a_t)$  : valor de tomar uma ação  $a_t$  num estado  $s_t$ ;
- $r_{t+1}$  : recompensa imediata;
- $\alpha$  : taxa de aprendizagem,  $0 < \alpha \ll 1$ ;
  - Proporção usada para atualizar o valor de utilidade após cada recompensa;
- $\gamma$  : fator de desconto ,  $0 < \gamma < 1$ ;
  - Encoraja o agente a preferir recompensas imediatas a longo prazo.
- $Q(s_t, a_t) = Q(s_t, a_t) + \alpha [ r_{t+1} + \gamma \times \text{MÁX}_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) ]$



- Temporal Difference Learning:

- O agente começa por assumir que todos os estados e todas as ações têm um valor inicial de 0 (zero);
  - O agente atualiza os valores calculando a diferença entre o valor esperado e o valor encontrado.

- SARSA (State-Action-Reward-State-Action):

- $Q(s_t, a_t)$  : valor de tomar uma ação  $a_t$  num estado  $s_t$ ;
  - $r_{t+1}$  : recompensa imediata;
  - $\alpha$  : taxa de aprendizagem,  $0 < \alpha \ll 1$ ;
    - Proporção usada para atualizar o valor de utilidade após cada ação;
  - $\gamma$  : fator de desconto ,  $0 \ll \gamma < 1$ ;
    - Encoraja o agente a preferir recompensas imediatas a tardias;
  - $$Q(s_t, a_t) = Q(s_t, a_t) + \alpha [ r_{t+1} + \gamma \times Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) ]$$

## **Q Learning vs SARSA**

- SARSA:

- $$Q(s_t, a_t) = Q(s_t, a_t) + \alpha [ r_{t+1} + \gamma \times Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) ]$$

- O agente está no estado 1, realiza a ação 1 e obtém a recompensa 1;
    - No estado 2, realiza a ação 2 e obtém a recompensa 2, e então atualiza o valor da ação 1 no estado 1;

- Q Learning:

- $$Q(s_t, a_t) = Q(s_t, a_t) + \alpha [ r_{t+1} + \gamma \times \text{MÁX}_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) ]$$

- O agente está no estado 1, realiza a ação 1 e obtém a recompensa 1;
    - Vê a recompensa máxima no estado 2 e atualiza o valor da ação 1 realizada no estado 1;

- SARSA considera a política de controlo que está a ser seguida e atualiza o valor das ações;
- Q Learning assume que está a seguir uma política ótima e usa-a para atualização;

## Aprendizagem por Reforço e Aprendizagem Supervisionada

- Aprendizagem supervisionada: aprender a alcançar um objetivo;
- Aprendizagem por reforço: aprender o melhor caminho até ao objetivo;
- A principal diferença entre aprendizagem Supervisionada e por Reforço está em que esta deve **explorar** todo o ambiente de forma insistente;



## ***“Exploitation” vs “Exploration”***

- *Exploitation* versus *Exploration* (Exploração versus Exploração!):
  - *Exploitation*: taking advantage;
  - *Exploration*: investigation;
- Distinção entre “*exploitation*” e “*exploration*”?



## ***“Exploitation” vs “Exploration”***

- O problema “*k-armed bandit*”, ou, o problema das máquinas de jogo (*slot machines*):
  - Existem  $k$  *slot machines*;
  - O agente pode puxar  $h$  vezes o braço de qualquer máquina;
  - Qualquer braço pode ser puxado em cada iteração;
  - O único custo é o de puxar o braço (não há depósito de moedas);
  - Quando um braço  $i$  é puxado, a máquina  $i$  paga 1 ou 0, de acordo com uma probabilidade  $p_i$ , desconhecida;
- **Que estratégia deve ser seguida?**
- Este é o problema que tipifica a distinção entre “*exploitation*” e “*exploration*”.



## ***“Exploitation” vs “Exploration”***

- O problema “*k*-armed bandit”, ou, o problema das máquinas de jogo (*slot machines*):
  - Existem  $k$  *slot machines*;
  - O agente pode puxar  $h$  vezes o braço de qualquer máquina;
  - Qualquer braço pode ser puxado em cada iteração;
  - O único custo é o de puxar o braço (não há depósito de moedas);
  - Quando um braço  $i$  é puxado, a máquina  $i$  paga 1 ou 0, de acordo com uma probabilidade  $p_i$ , desconhecida;
- **Que estratégia deve ser seguida?**
  - Durante quanto tempo se joga?
    - Se  $h \gg 0$ : “*exploration*”;
    - Se  $h \approx 0$ : “*exploitation*” (ganância/*greedy*);
  - Quanto maior a duração do jogo, maior a probabilidade de cair num ótimo local e piores as consequências a longo prazo.



## ***“Exploitation” vs “Exploration”***

- 
- “Exploration” tende a procurar mais informação sobre o ambiente;
  - “Exploitation” tende a maximizar a recompensa através de informação já conhecida;
  
  - Como escolher um restaurante?
    - “Exploration”: \_\_\_\_\_
    - “Exploitation”: \_\_\_\_\_
  
  - Como escolher um curso?
    - “Exploration”: \_\_\_\_\_
    - “Exploitation”: \_\_\_\_\_

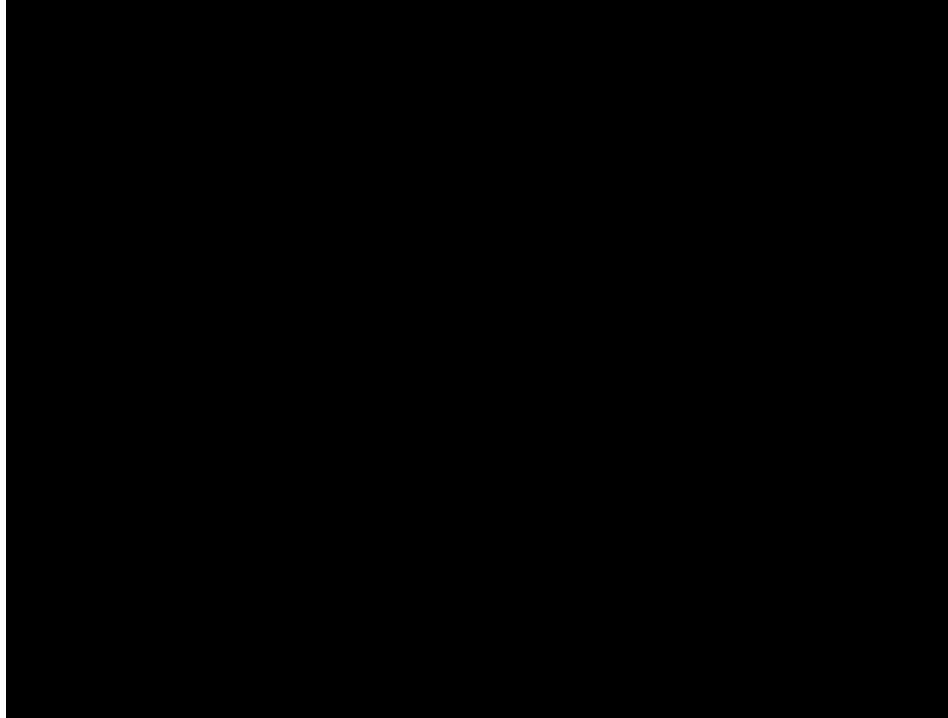
- QBot Learning;



In this demo, I use a model-free reinforcement learning algorithm called q-learning to obtain a walking policy for a crawling robot. Each servo has 3 states and the reward signal is provided by an incremental encoder.

In the beginning, the agent randomly explores the state space and keeps track of his actions and the associated rewards in the  $Q(s,a)$  matrix. The randomness in his actions decreases over time and actions that guarantee a maximization of the cumulated reward are performed more often.

- Inverted Pendulum;



- Reinforcement Learning (<https://www.cse.unsw.edu.au/~cs9417ml/RL1/applet.html>):
  - Executar o *applet* que surge na página *web*;
  - Regras do jogo (gato & rato):
    - Tanto o gato como o rato têm 8 graus de liberdade (N/NE/E/SE/S/SO/O/NO);
    - O rato recebe recompensa de 1 ponto se alcançar o queijo;
    - Sempre que o rato alcança o queijo, é colocada nova porção no tabuleiro;
    - O gato é recompensado com 1 ponto quando alcança o rato;
    - O jogo termina quando o gato alcança o rato.
  - Seguir os tutoriais presentes na página:
    - Part 1: Understanding the applet;
    - Part 2: Learning Settings;

**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

# Sistemas de Aprendizagem

Aprendizagem e Extração de Conhecimento  
Perfil Sistemas Inteligentes @ MiEI/4º – 1º Semestre

Cesar Analide, José Neves, Paulo Novais

## Referências bibliográficas

- Arthur L. Samuel, “Some Studies in Machine Learning Using the Game of Checkers.” IBM Journal of Research and Development 3(3):210–229, 1959.
- Marvin Minsky, “Steps towards artificial intelligence.” Computers and Thought, 406–450, 1961.
- Richard S. Sutton and Andrew G. Barto, “[Reinforcement Learning I: Introduction](#)”, The MIT Press, 1998.
- V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, “Playing Atari with Deep Reinforcement Learning”, Deepmind, 2013.

**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

## Aprendizagem e Extração de Conhecimento Perfil Sistemas Inteligentes @ MiEI/4º – 1º Semestre

Cesar Analide, José Neves, Paulo Novais