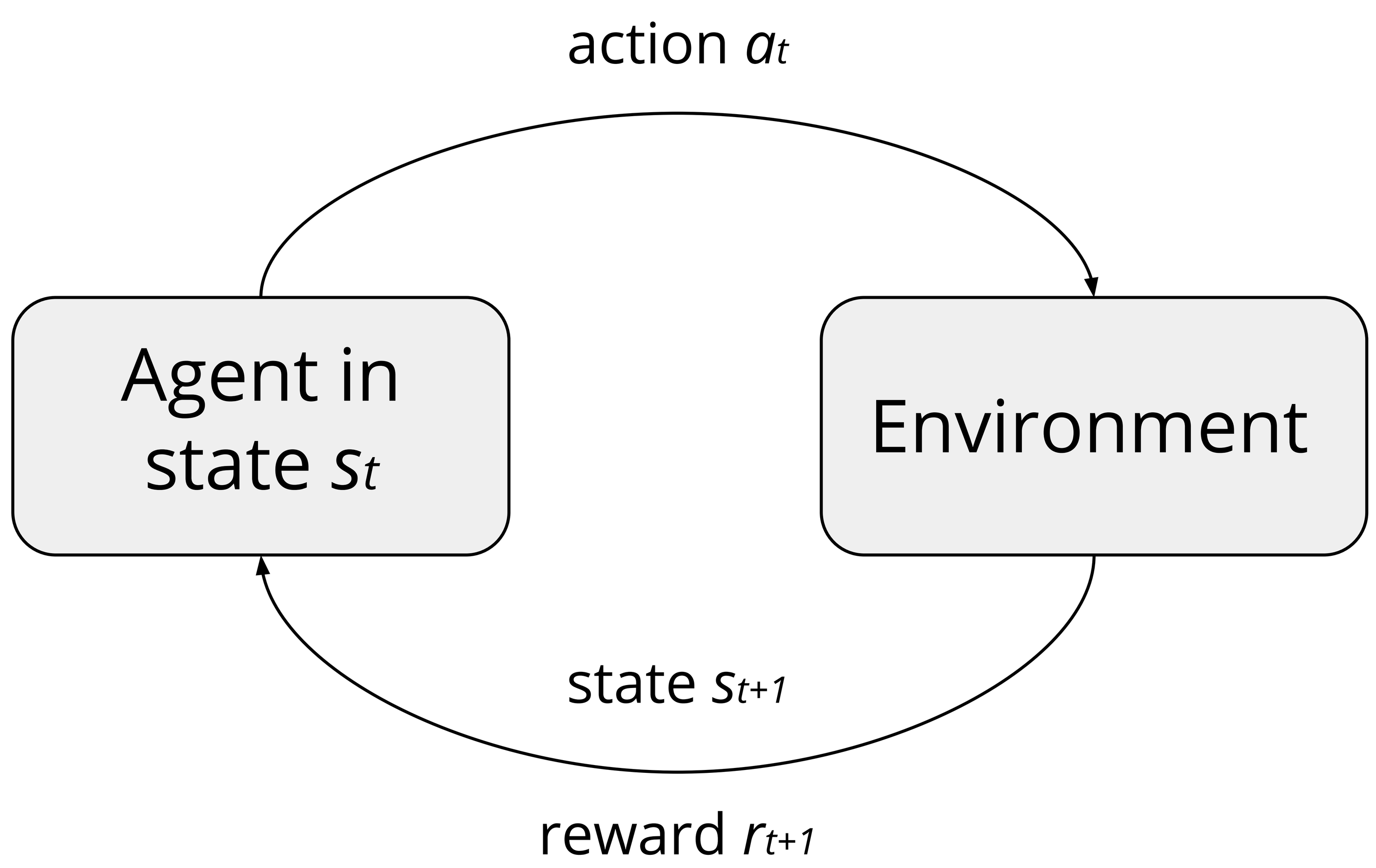
Explainable Reinforcement Learning

Francisco Cruz, Richard Dazeley, Peter Vamplew
School of Information Technology, Deakin University, Australia.
School of Science, Engineering and Information Technology, Federation University, Australia
Contact Email: francisco.cruz@deakin.edu.au

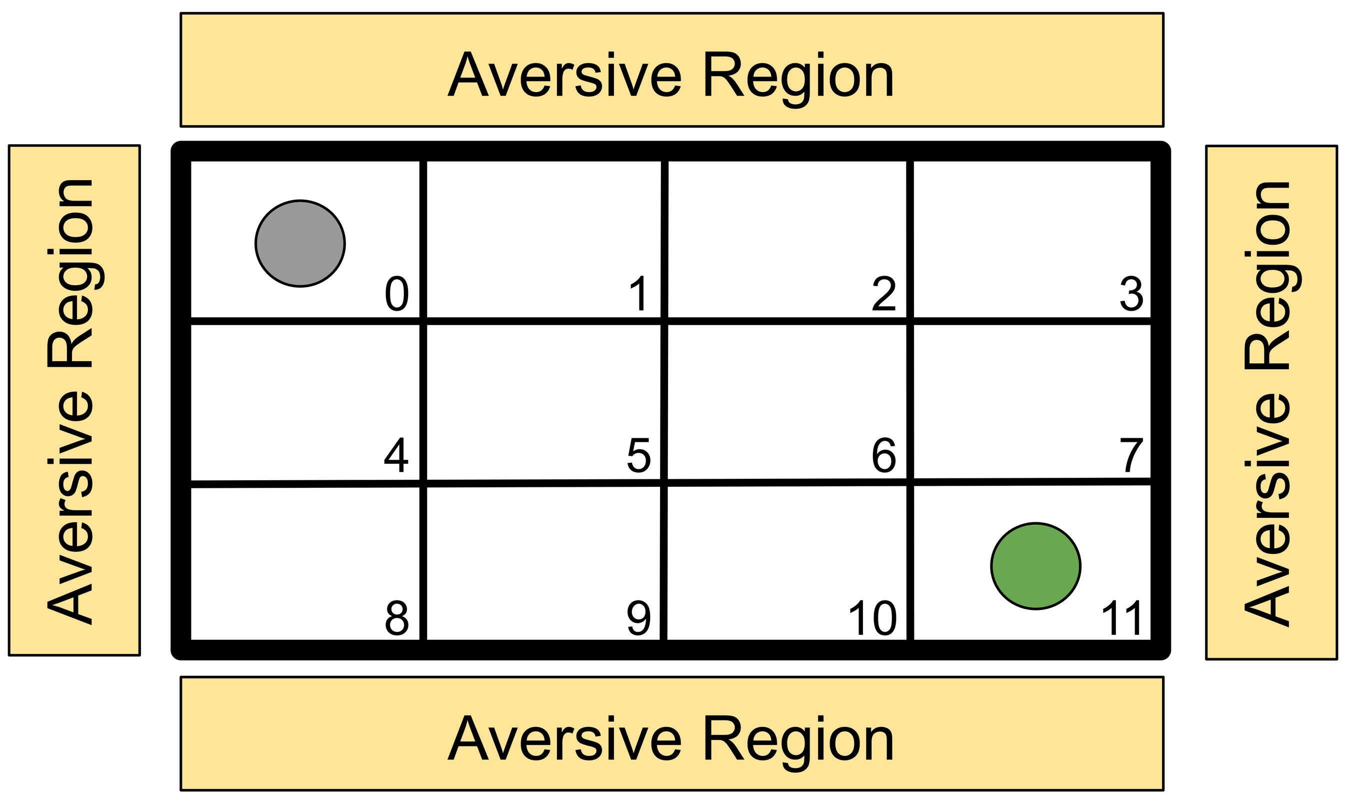
1. Motivation

- Reinforcement learning (RL) [1] is a learning approach based on behavioural psychology.
- The aim of RL is to provide an autonomous agent with the ability to learn new skills by only interacting with its environment.
- An open issue in RL is the lack of visibility and understanding for end-users in terms of decisions taken by an agent.
- We propose a memory-based explainable RL (MXRL) approach, which allows an agent to explain decisions using domain language.



3. Experimental set-up

- A 3x4 grid world scenario in two versions: bounded and unbounded.
- Four allowed actions in this scenario: down, up, right, and left.



5. Conclusions

- We presented MXRL to explain to non-expert end-users the reasons why some decisions are taken in certain situations.
- Using a episodic memory, we have computed Ps and Nt.
- MXRL allows the agent to provide explanations using domain-based language.
- Using another more general method, such as function approximator or phenomenological relations from the Q-values.
- Scale to more complex scenarios.

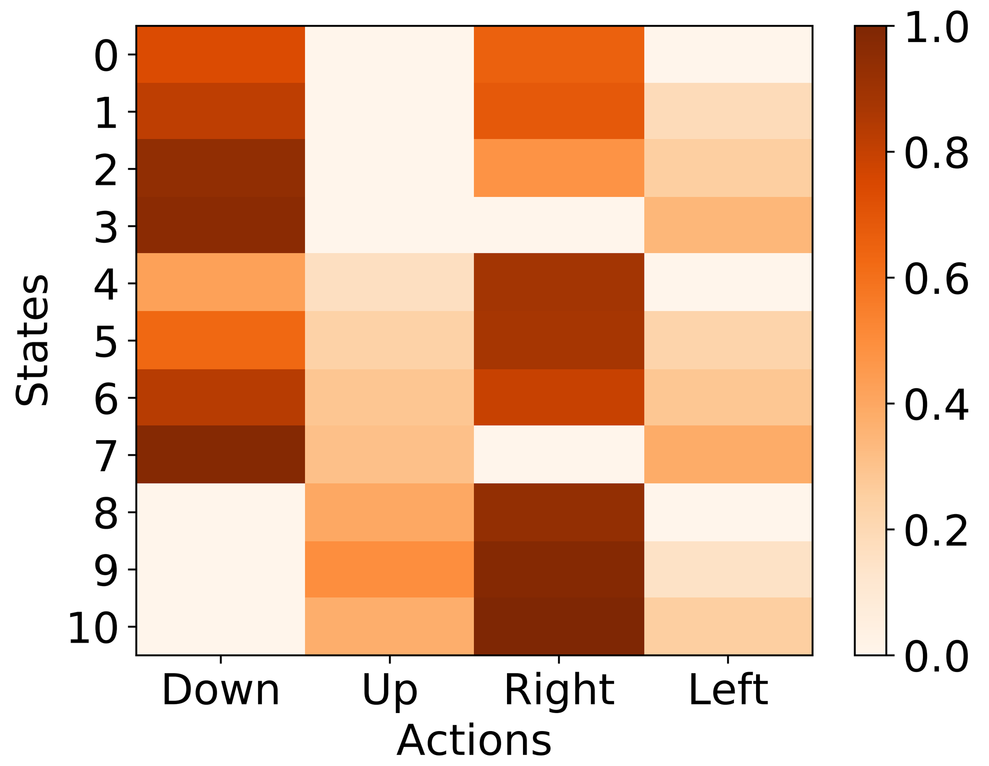
2. Memory-based Explainable Reinforcement Learning

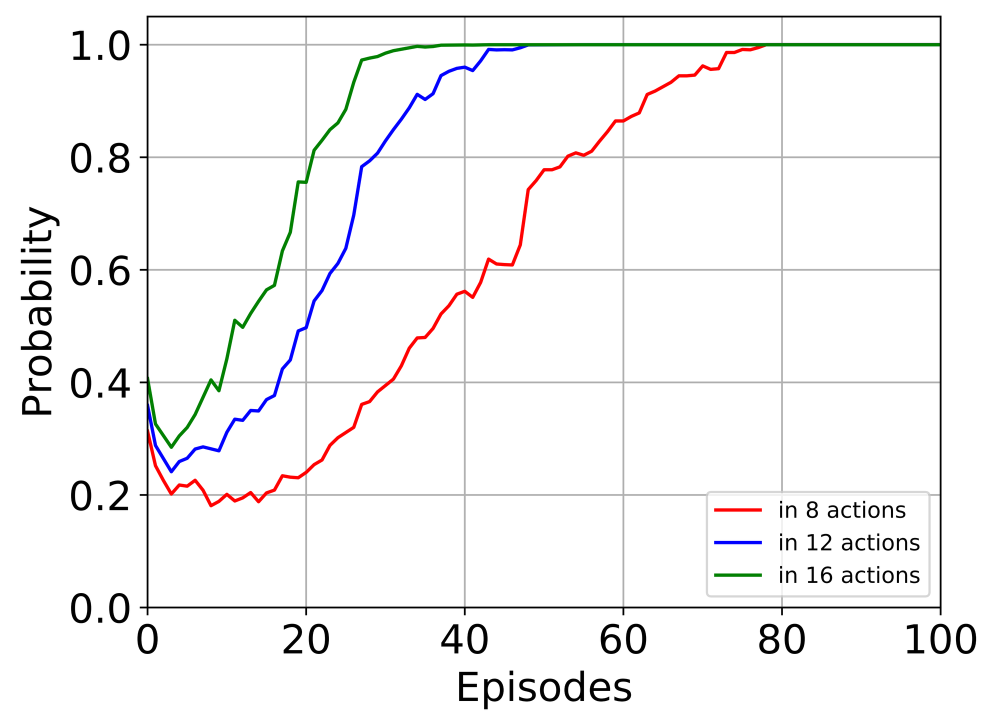
- It is essential that non-expert end-users can understand agents' intentions to obtain more details in case of a failure.
- From a non-expert end-user perspective, most relevant questions: 'why?' and 'why not?' [2, 3].
- To answer these questions we compute (i) artificial agent's probability of success (Ps) and (ii) number of transitions to reach the goal state (Nt).
- Memory-based explainable reinforcement learning approach with the on-policy method SARSA to compute the probability of success and the number of transitions to the goal state.

Initialize $Q(s,a)$, T_t, T_s, P_s, N_t
for each episode **do**
  Initialize $T_{List}[]$
  Choose an action using $a_t \leftarrow$ SELECTACTION(s_t)
  repeat
    Take action a_t
    Save state-action transition T_{List}.add(s, a)
    $T_t[s][a] \leftarrow T_t[s][a] + 1$
    Observe reward r_{t+1} and next state s_{t+1}
    Choose next action a_{t+1} using softmax action selection method
    $Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)]$
    $s_t \leftarrow s_{t+1}$; $a_t \leftarrow a_{t+1}$
  until s is terminal (goal or aversive state)
  if s is goal state **then**
    for each s,a $\in T_{List}$ **do**
      $T_s[s][a] \leftarrow T_s[s][a] + 1$
    end for
  end if
  Compute $P_s \leftarrow T_s/T_t$
  Compute N_t for each s $\in T_{List}$ as pos(s, T_{List}) + 1
end for

4. Experimental Results

- Why did you choose action down when in state 0? Using Ps: *I chose to go down because that has a 73.6% probability of successfully reaching the goal.*
- Why did you not choose to go left when in state 0? Using Ps: *I did not choose left because that has a zero probability of success, whereas by choosing down has a 73.6% probability of success, which was higher than other actions.*
- What is the probability of finishing the task in 8 movements starting from the state 0? Using Nt: (i) After 30 episodes: *I can finish the task in 8 movements with a probability of 39.4%*. (ii) After 60 episodes: *I can complete the task in 8 moves with a probability of 86.5%.*





References

[1] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: Bradford Book, 1998.

[2] Lim, B.Y., Dey, A.K., Avrahami, D. Why and why not explanations improve the intelligibility of context-aware intelligent systems. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. pp. 2119–2128, 2009.

[3] Madumal, P., Miller, T., Sonenberg, L., Vetere, F. *Explainable reinforcement learning through a causal lens*. arXiv preprint arXiv:1905.10958, 2019.