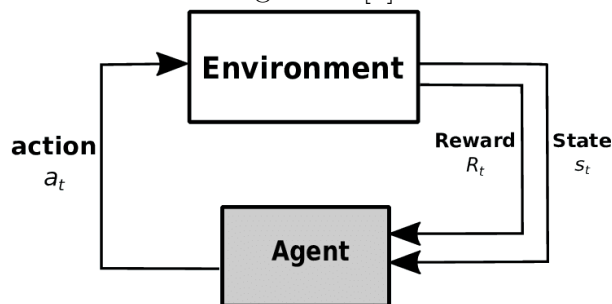


# Introduction to Reinforcement Learning in PyTorch[2]

---

## 1 Basics of Reinforcement Learning

Figure 1: [1]



RL algorithms are often modeled as Markov Decision Processes. Hence, at time step  $t$ , the **agent** (RL algorithm) is situated in **state**  $s_t$ . The agent interacts with an environment by taking an **action**  $a_t$ . This action results in a new state  $s_{t+1}$  and the transition  $(s_t, a_t)$  brings with it a **reward**  $r_t$ . Often times, there is a probability distribution over the transition  $(s_t, a_t)$  to a new state  $s_{t+1}$ . Additionally, there often exist **episode-ending states**, which corresponds to reaching a final goal. Your goal is to learn a **policy**  $\pi$  that maps states to actions.

Although, in an MDP, we assume that we can always tell which state  $s_t$  our agents is in, this isn't always the case. In these cases, we have observations  $o_t$ .

## 2 Notes

We always know the state of the agent  $\rightarrow$  no observations necessary.

### 3 Notation

$a$	Action
$r$	Reward
cell7	cell8

### 4 Definitions

**Episode:** The trajectory of going from start to finish of a task.

## References

- [1] Roohollah Amiri et al. “A Machine Learning Approach for Power Allocation in HetNets Considering QoS”. In: (Mar. 2018).
- [2] Harsh Panchal. *Introduction to Reinforcement Learning (RL) in PyTorch*. URL: <https://medium.com/analytics-vidhya/introduction-to-reinforcement-learning-rl-in-pytorch-c0862989cc0e>. (accessed: 26.08.2024).