Example:

\[ \text{Starting state} \]
\[ r = 0 \text{ elsewhere} \]
\[ r = -10 \]
\[ \text{Final state} \]
\[ r = 10 \]

\[ P = 0.8 \text{ act } \]
\[ P = 0.05 \text{ left} \]
\[ P = 0.05 \text{ right} \]
\[ P = 0.1 \text{ stay} \]
\[ \text{run into wall} = 0 \text{ stay} \]

MDP:

- Mathematical formulation of environment, what we want agent to learn

\[ t \in \{0, 1, \ldots \} \text{ time steps} \]

\[ S_t, A_t, R_t \]

state, action, reward respectively
MDP:

\[ M = (S, A, P, R, d, \delta) \]

- \( S \) = set of all possible states = state space
- \( A \) = action space
- \( P \) = transition function

\[ P : S \times A \times S \rightarrow [0,1] \]

\[ P(s,a,s') \triangleq \Pr(S_{t+1} = s' | S_t = s, A_t = a) \]

\[ S_t = s, A_t = a \]

\[ \forall s \in S, a \in A, s' \in S \]

Note: Deterministic \( P \Rightarrow P(s,a,s') \in \{0,1\} \)

\[ \sum_{s'} P(s,a,s') = 1 \]


\[ R = \text{Reward Function, determines how rewards are generated!} \]

\[ R : S \times A \times S \Rightarrow R \]

\[ R(s, a, s') = \mathbb{E}[R_t | s_t = s, a_t = a, s_{t+1} = s'] \]

**E.g.**

*In robot gridworld problem*

\[ R(20, \text{right}, 21) = -10 \]

\[ \begin{array}{c|cc}
19 & 20 & 21 \\
\hline
& r = -10 \\
\end{array} \]

\[ \begin{array}{c|c|c}
\leq & a & \leq' \\
\hline
20 & \text{right} & 21 \\
\end{array} \]

**Note:** From answering a question—not the notation we will use hereafter.

\[ R(s, a) = \text{deterministic reward value} \]

\[ \text{Prob.} \times \text{reward value} \]

\[ R(s, a, s') \text{ (for deterministic reward, not for stochastic reward)} \]
do = initial state distribution
\[ \text{do} \ni s \rightarrow [0,1] \]
\[ \text{do}(s) = p(s_0 = s) \]
\[ \gamma = \text{reward discount parameter} \]

Agent Formulation (for MDP)

- **Policy** - The mechanism contains the agent that determines which action to take in a state

\[ \text{Policy} = \pi : S \times A \Rightarrow [0, 1] \]
\[ \pi(s, a) \triangleq P_r(A_t = a \mid s_t = s) \]

- **Learning** - Corresponds to the agent changing its policy
if $|s|_1$ and $|A|$ are finite, if $R_t$ is bounded then an optimal policy exists

**Applying MDP for gridworld problem:**

- $s_0 \sim \pi (s_0, \cdot)$
- $s_1 \sim P(s_0, a_0, \cdot)$
- $R_0$ computed with $E[R_0] = R(s_0, a_0, s_1)$

**What agent want to do?**

**Goal:** Maximize expected reward

**Objective function:** $J: \pi \rightarrow \mathbb{R}$

$$J(\pi) = E \left[ \sum_{t=0}^{\infty} R_t \right]$$

**Optimal policy:** $\pi^* \in \text{arg} \max_{\pi \in \Pi} J(\pi)$