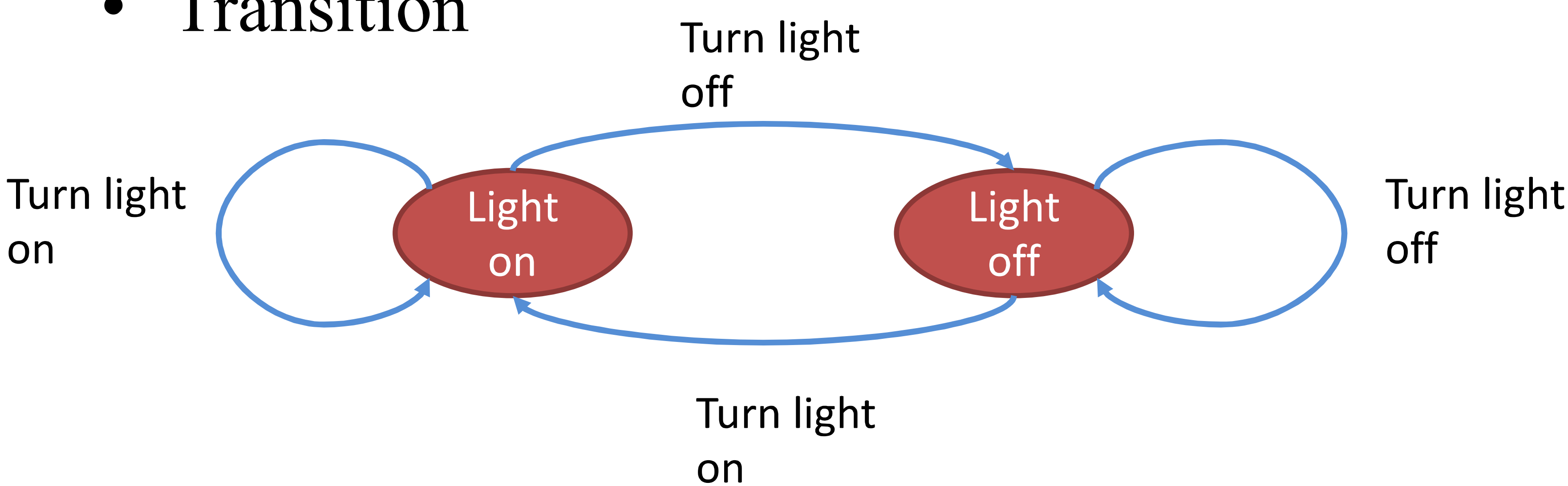
Markov Decision Process (MDP)

Finite State Machine

Definition

- State
  - Initial state
  - Final state
- Transition

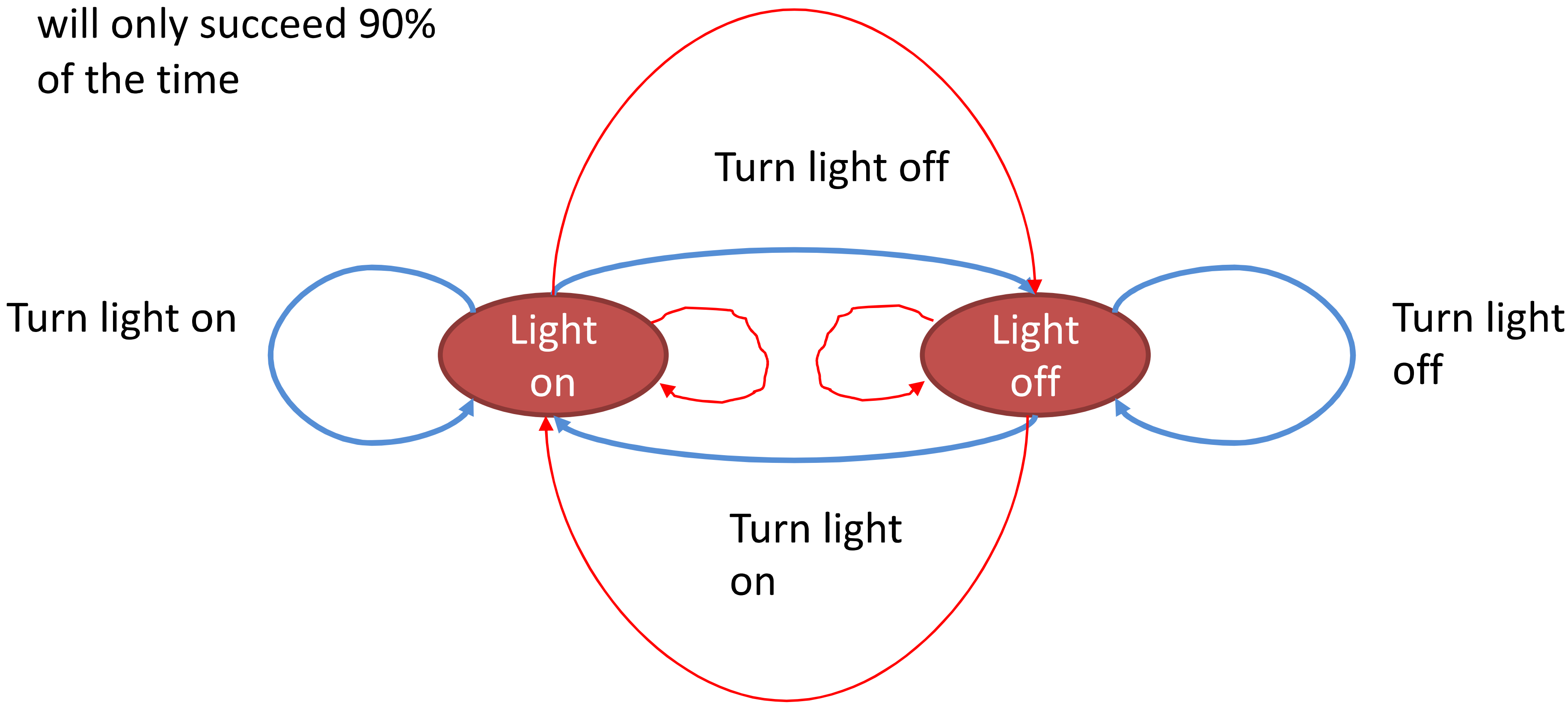


Markov Decision Process (MDP)

- Dealing with actions with uncertainty

Markov Decision Process (MDP)

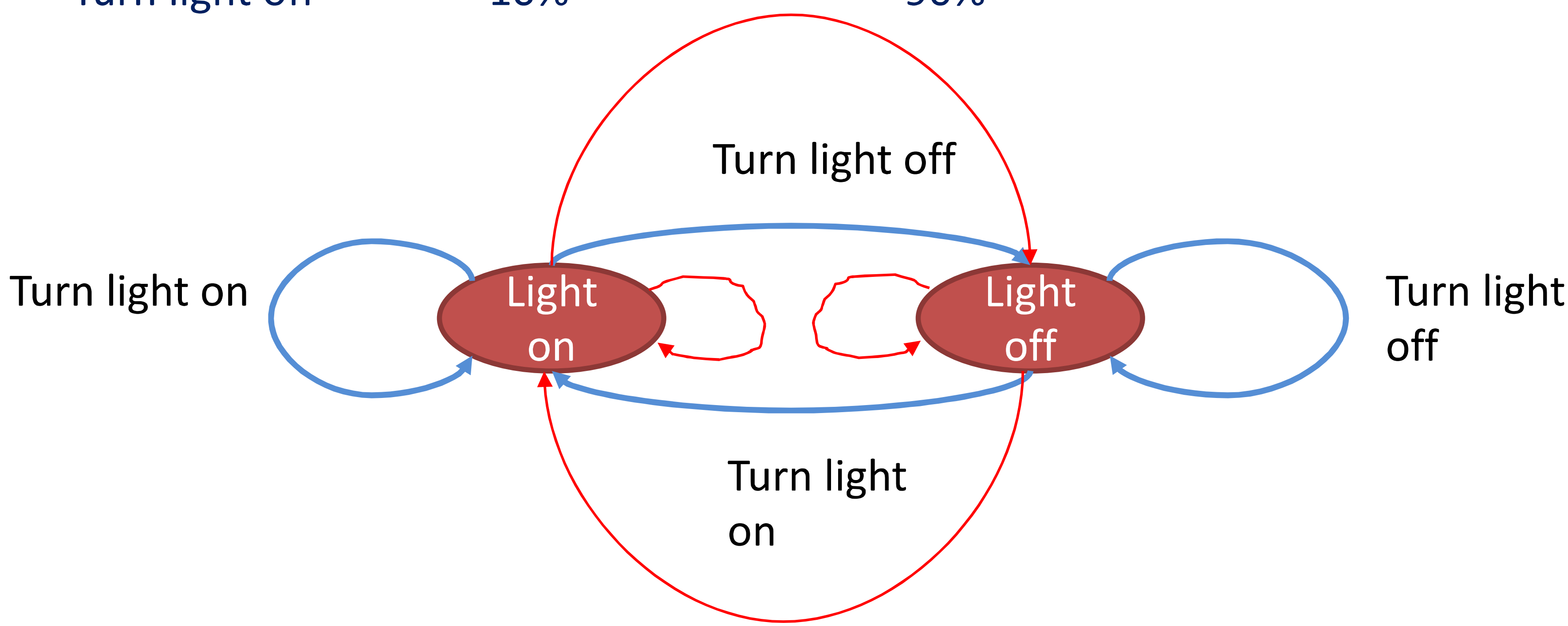
Assume all the actions will only succeed 90% of the time



Markov Decision Process (MDP)

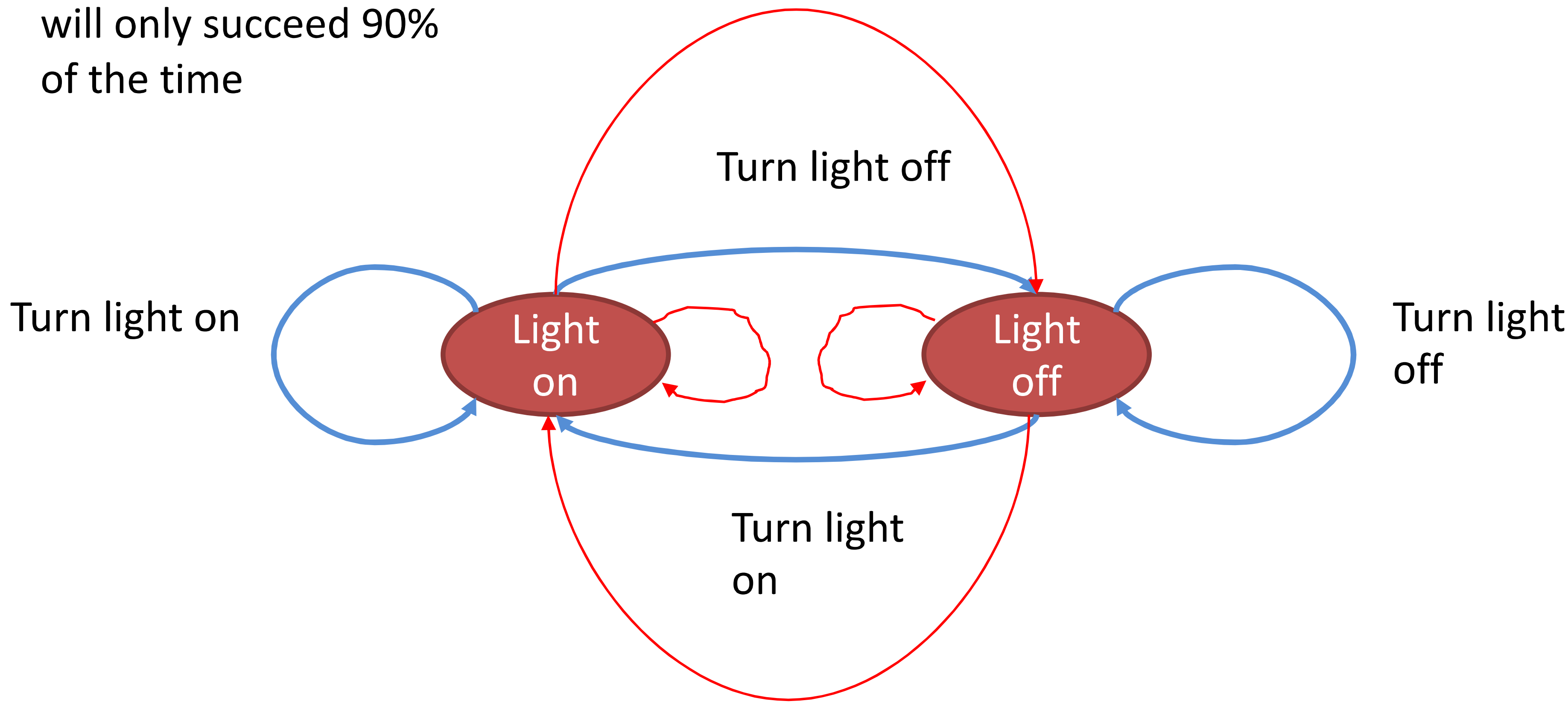
$P_a(s, s')$, $P(s, a, s')$

	Light on	Light off
Turn light on	90%	10%
Turn light off	10%	90%



Markov Decision Process (MDP)

Assume all the actions will only succeed 90% of the time



Transition function affects the probabilities of the system in each state:
starting with light on, then turn the light off

$$\begin{bmatrix}1\\0\end{bmatrix}\times\begin{bmatrix}.1 & .9\\.9 & .1\end{bmatrix}=\begin{bmatrix}.1\\.9\end{bmatrix}$$

Markov Decision Process

- R(s) = immediate reward if the state is s
- Define the goal of a MDP
  - For example, if we want the light to be off by the end of the game, we can define
    - R(light on) = 0
    - R(light off) = 1

Markov Decision Process

- Automated action selection: maximize expected rewards

  current state s

  for each action *a*:

  reward = sum (p(s,a,s')*r(s'))

  select the action with the highest reward

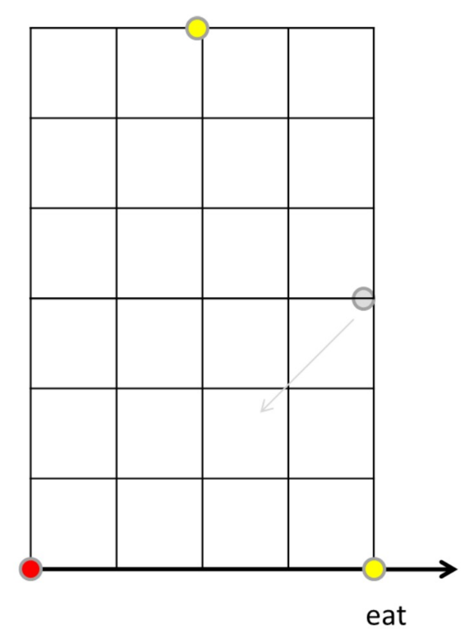
So if the light is on, what is the best action to do? R1step(S1)

Action1: turn the light on

Reward = (.9*0+.1*1) = .1

Action 2: turn the light off

Reward = (.1*0+.9*1) =.9



Markov Decision Process

- Tuple (S, A, P(.,.), R(.)))
  - S -> state space
  - A -> action space
  - $P_a(s, s') = Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$
  - R(s) = immediate reward if the current state is s

Markov Decision Process

- Automated action selection: maximize expected rewards

    current state s
    for each action *a*:
        reward = sum (p(s,a,s')*r(s'))
    select a with the highest reward

- Discount of future rewards:
  - Total reward after n steps= reward0 + $\gamma 1 *$ reward1 + $\gamma 2 *$ reward2+…+$\gamma(n-1) *$ reward(n-1)
  - I want to calculate rewards over two steps starting from S1

    A1
        0.9*R(S1) + 0.1*R(S2)   -- step1 reward, reward0
        0.9*(R1step(S1))+ 0.1*(R1step(S2))   -- step2 reward, reward1
    A2
        ??

Markov Decision Process

S1 R=0	S2 R=.1
S3 R=.5	S4 R= -.1

	Start State	End state with probability			
		S1	S2	S3	S4
a1	S1	.1	.8	.1	
a2	S2	.6		.3	.1
a3	S3	.1	.4	.2	.3
a4	S4		.3	.5	.2
a5	S3	.2	.2	.4	.2

- Starting from state S3
- Assuming you can move 3 steps, and we do not discount future rewards
- What is the best movement?
- Max(sum(p(s,a,s')*(r(s')+ γ*future_reward from s')))