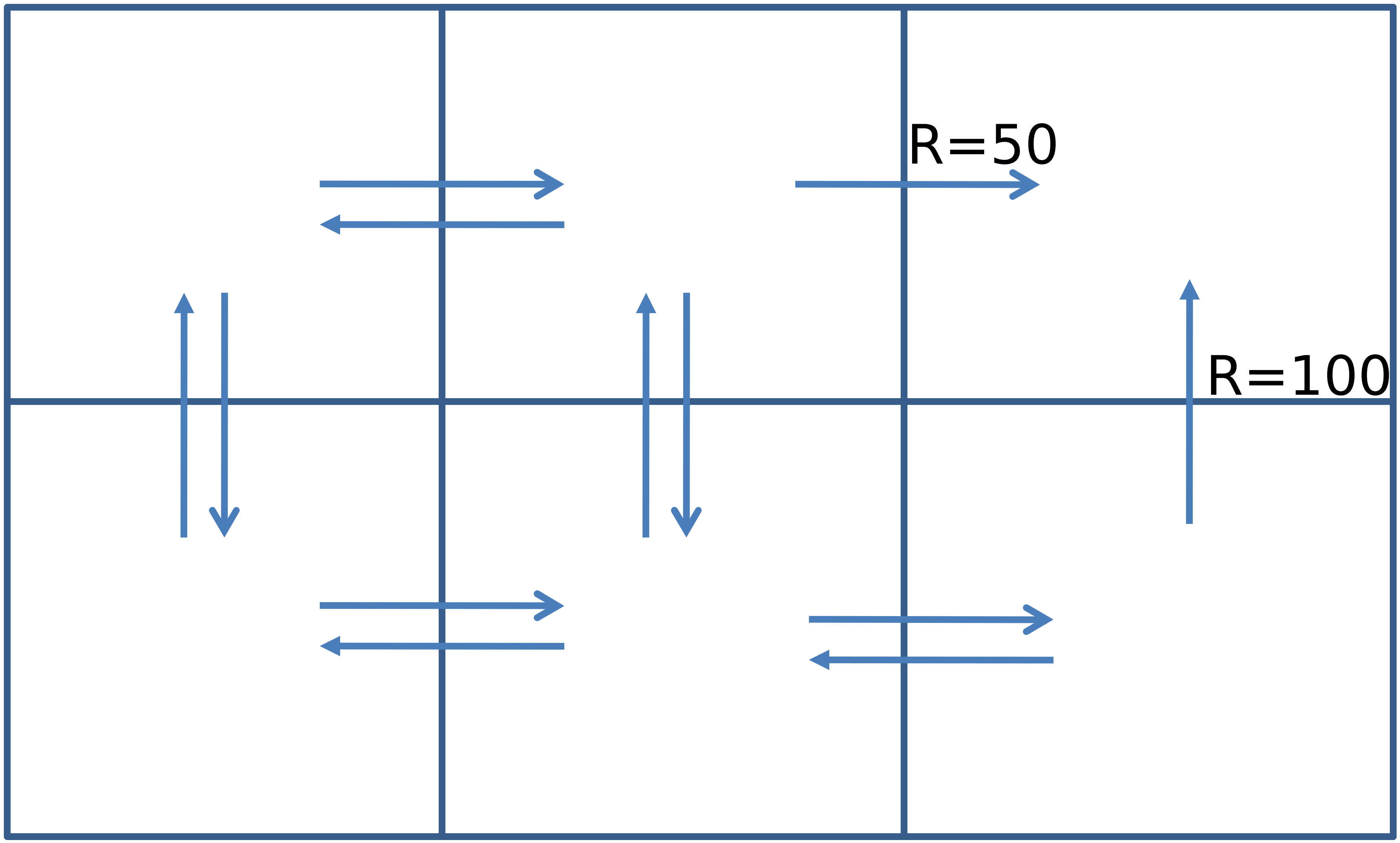
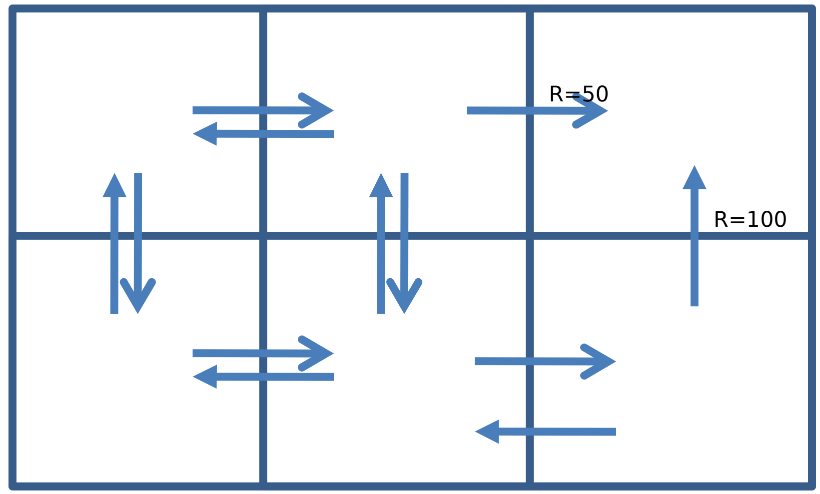
Reinforcement Learning

Some slides taken from previous 10701 recitations/lectures

A (Fully Deterministic) World



World



A **policy** is a mapping from **State => Action**. Normally denoted as π**(x)=a**

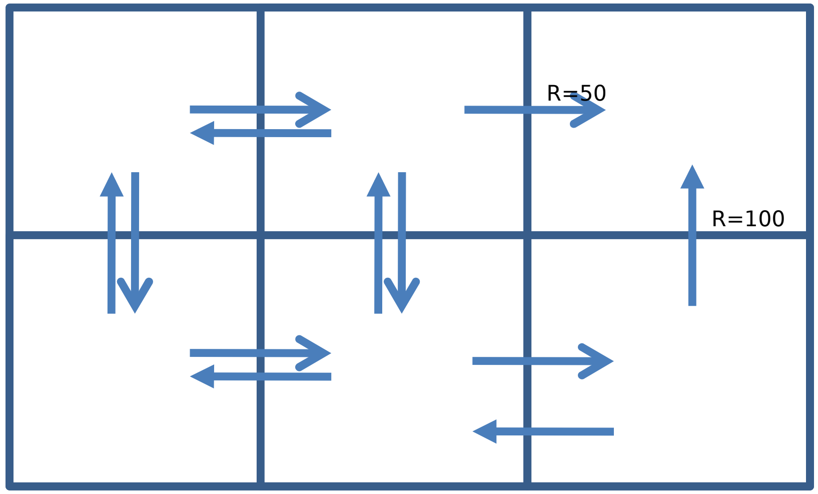
"What action do I make if I find myself in a particular place?"

Possible Questions.

1. If I am in state X. What is the value of following a particular policy?

2: What is the best policy?

World

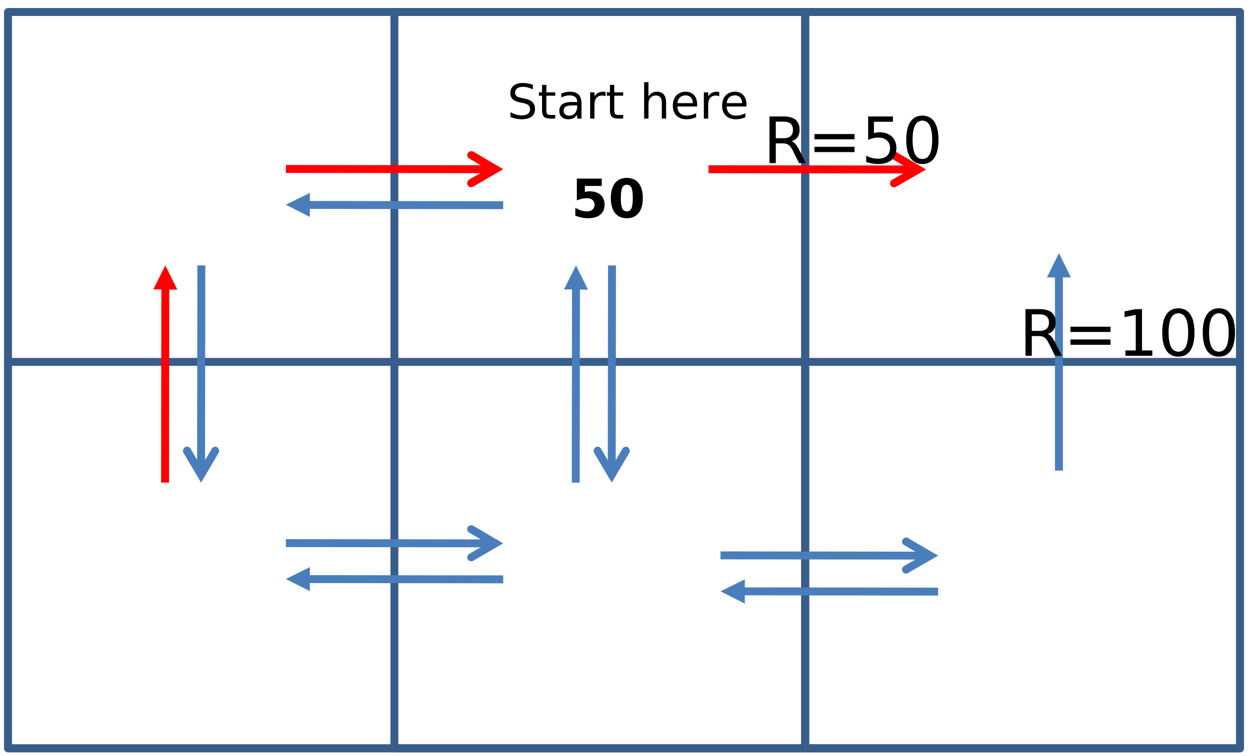


- Set of states S
- Set of actions A
- At each time, agent observes state $s_t \in S$, then chooses action $a_t \in A$
- Then receives reward r_t , and state changes to s_{t+1}
- Markov assumption: $P(s_{t+1} \mid s_t, a_t, s_{t-1}, a_{t-1}, ...) = P(s_{t+1} \mid s_t, a_t)$
- Also assume reward Markov: $P(r_t \mid s_t, a_t, s_{t-1}, a_{t-1},...) = P(r_t \mid s_t, a_t)$

Long Term Reward

Total Reward: Reward is discounted by the time I obtained it

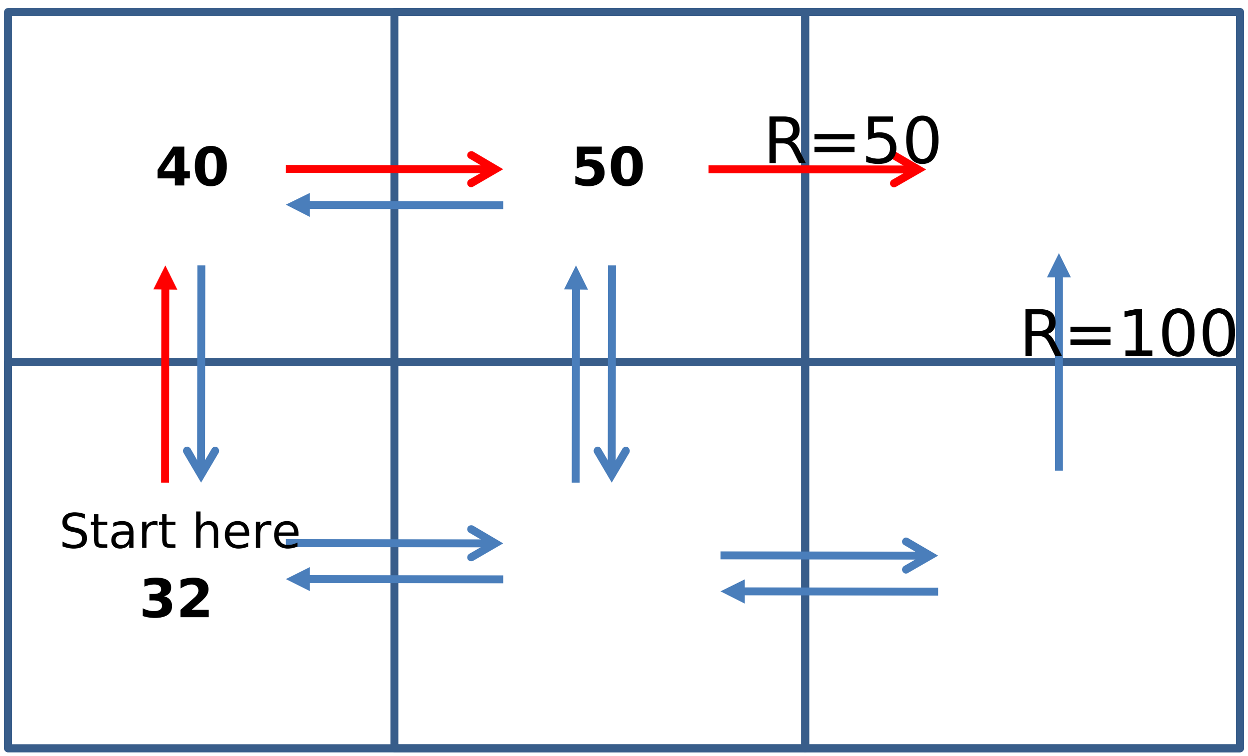
$$value = \sum_t \gamma^t r_t ; \gamma = 0.8$$



Long Term Reward

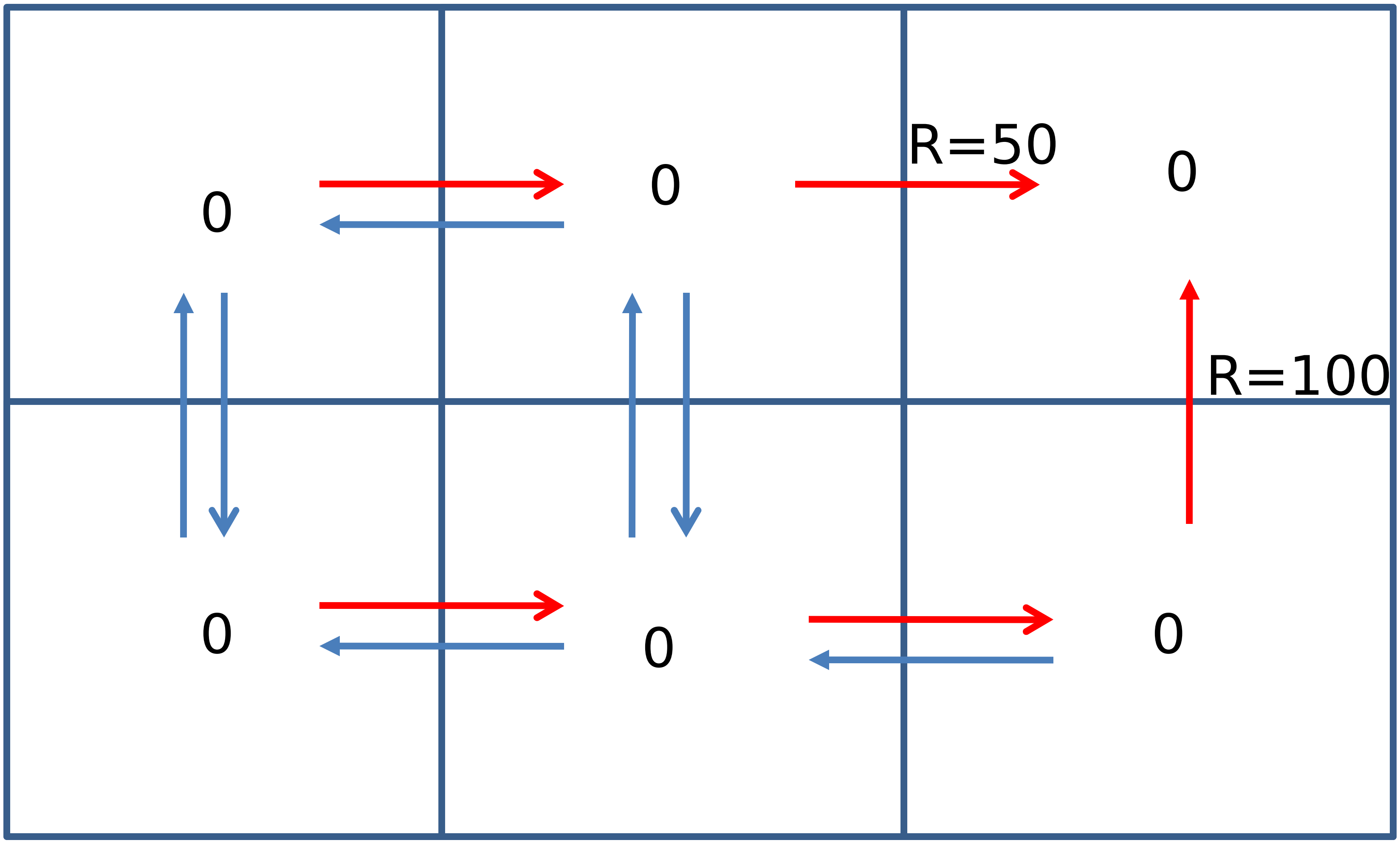
Total Reward: Reward is discounted by the time I obtained it

$$value = \sum_t \gamma^t r_t ; \gamma = 0.8$$



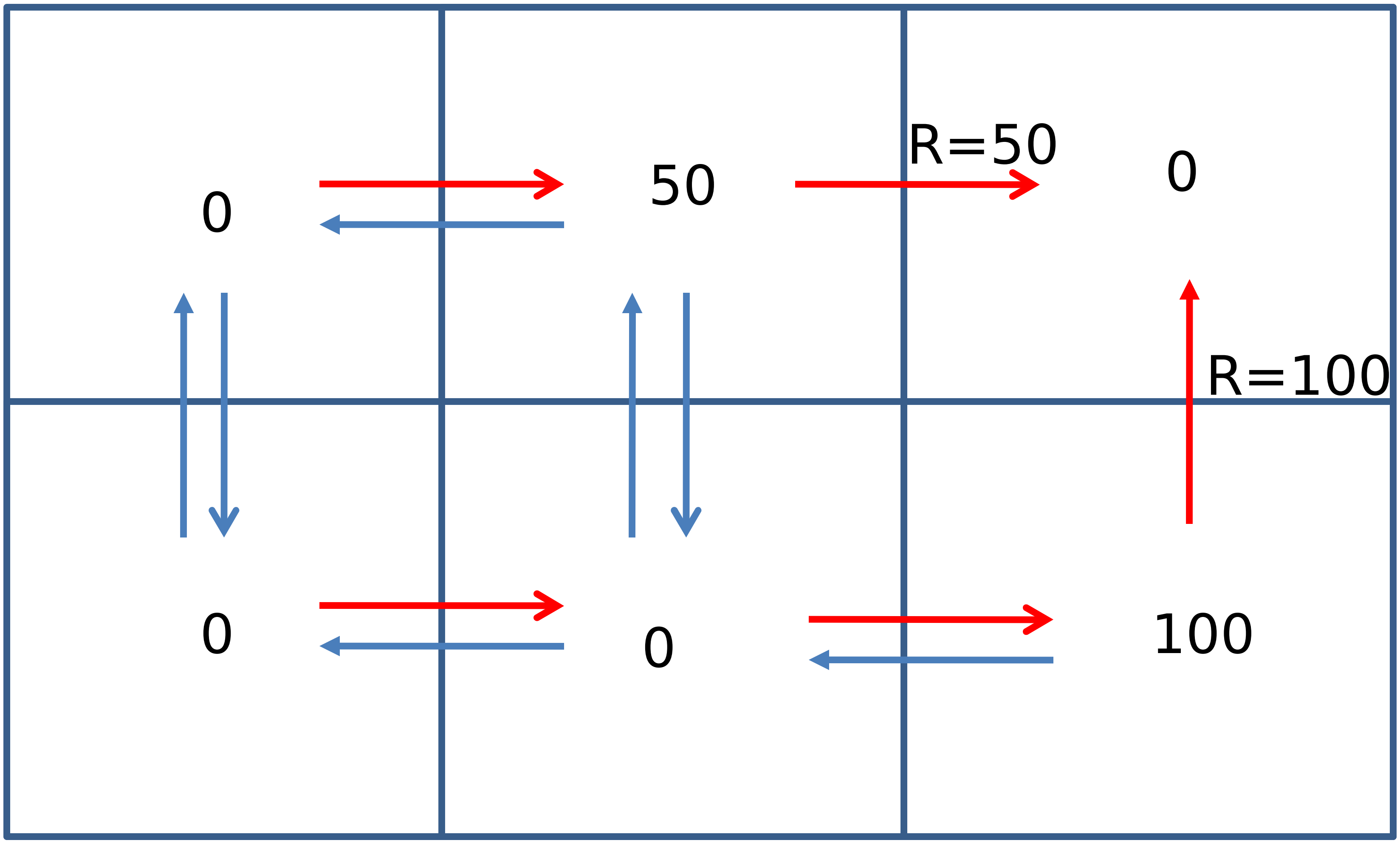
We can Reuse Computation!

Value of a Policy if I run for 0 time steps



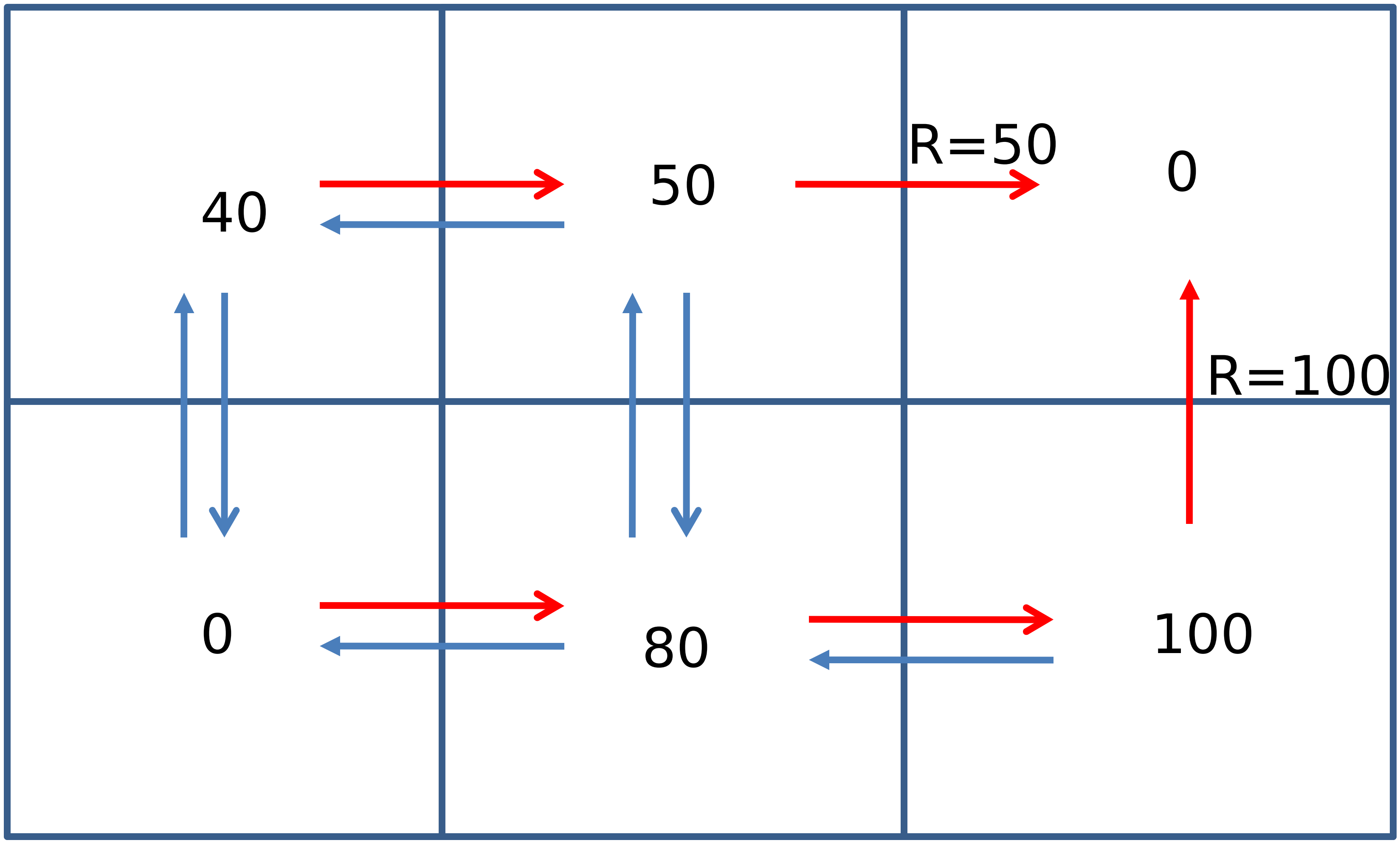
V_0

Value of a Policy if I run for 1 time step



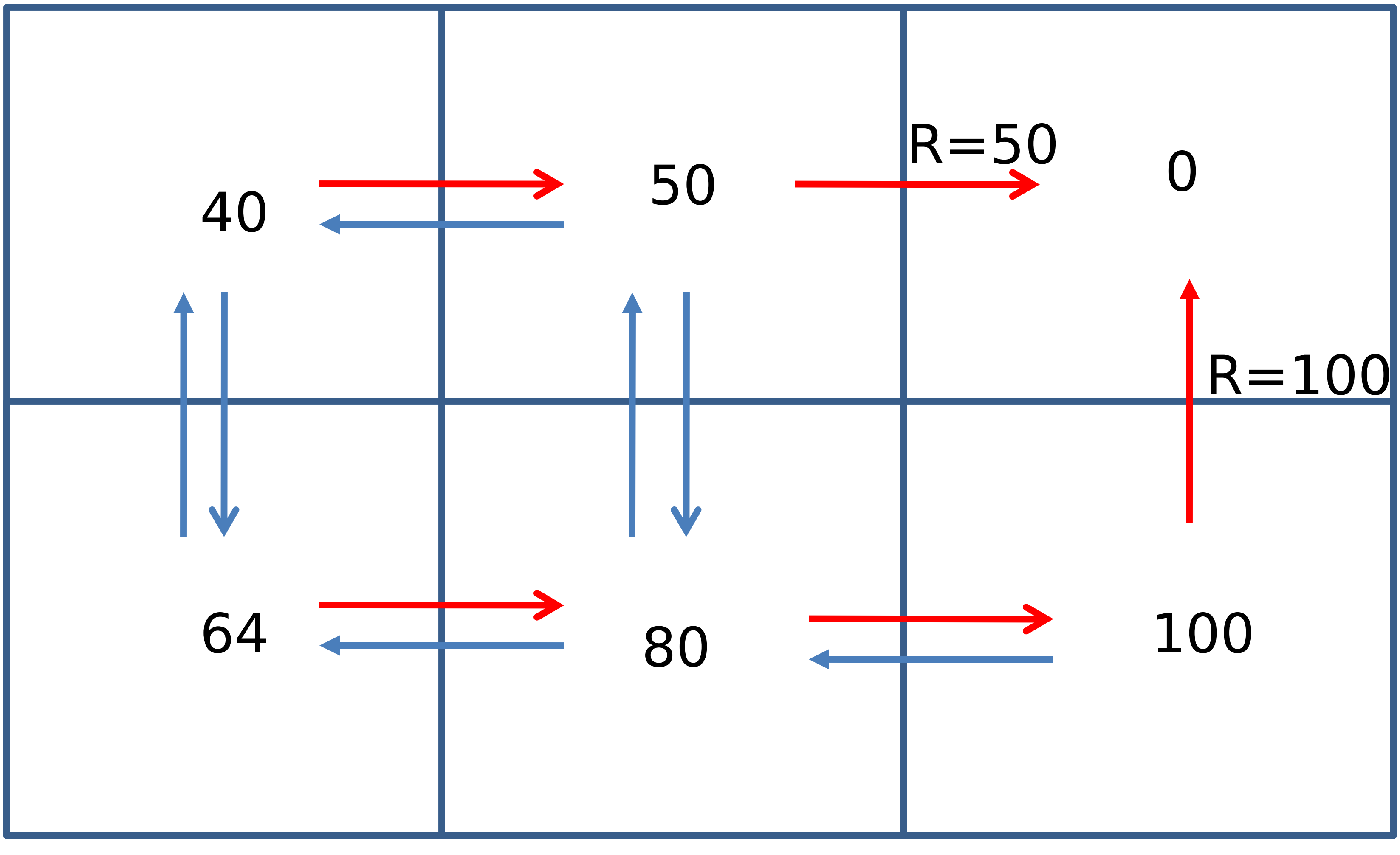
V_1

Value of a Policy if I run for 2 time steps



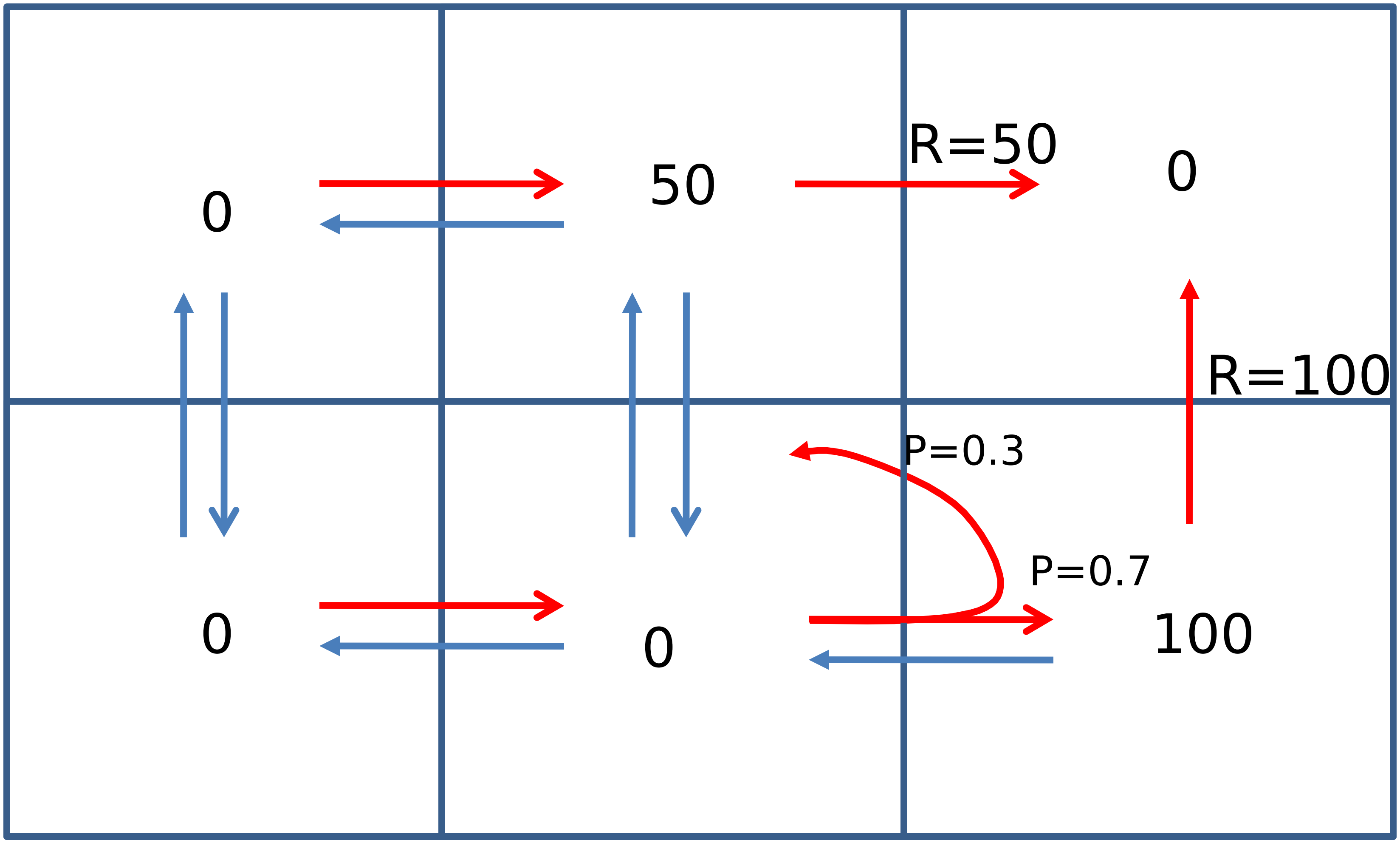
$\mathbf{V_2}$

Value of a Policy if I run for 3 time steps



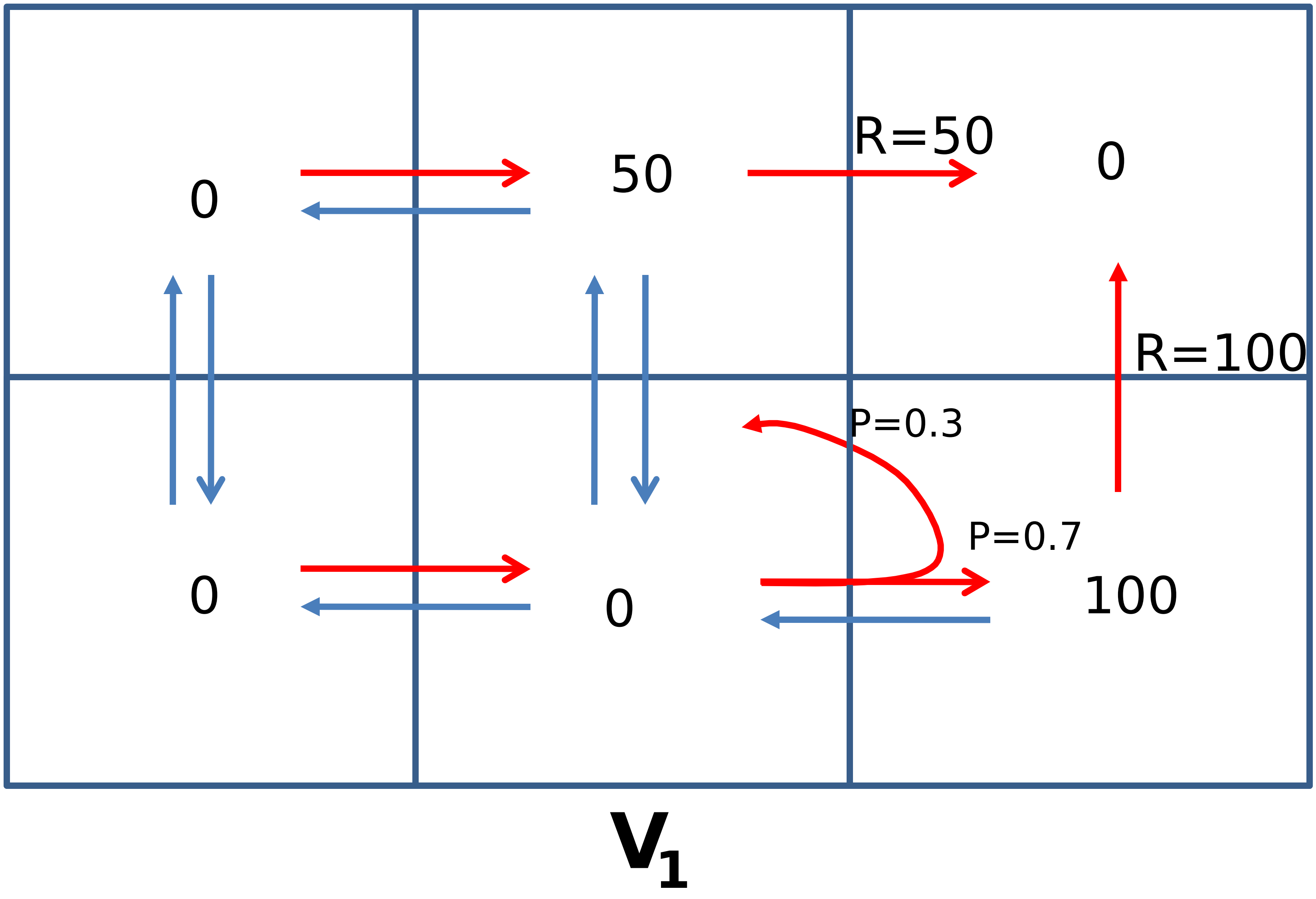
V_3

Non-deterministic World

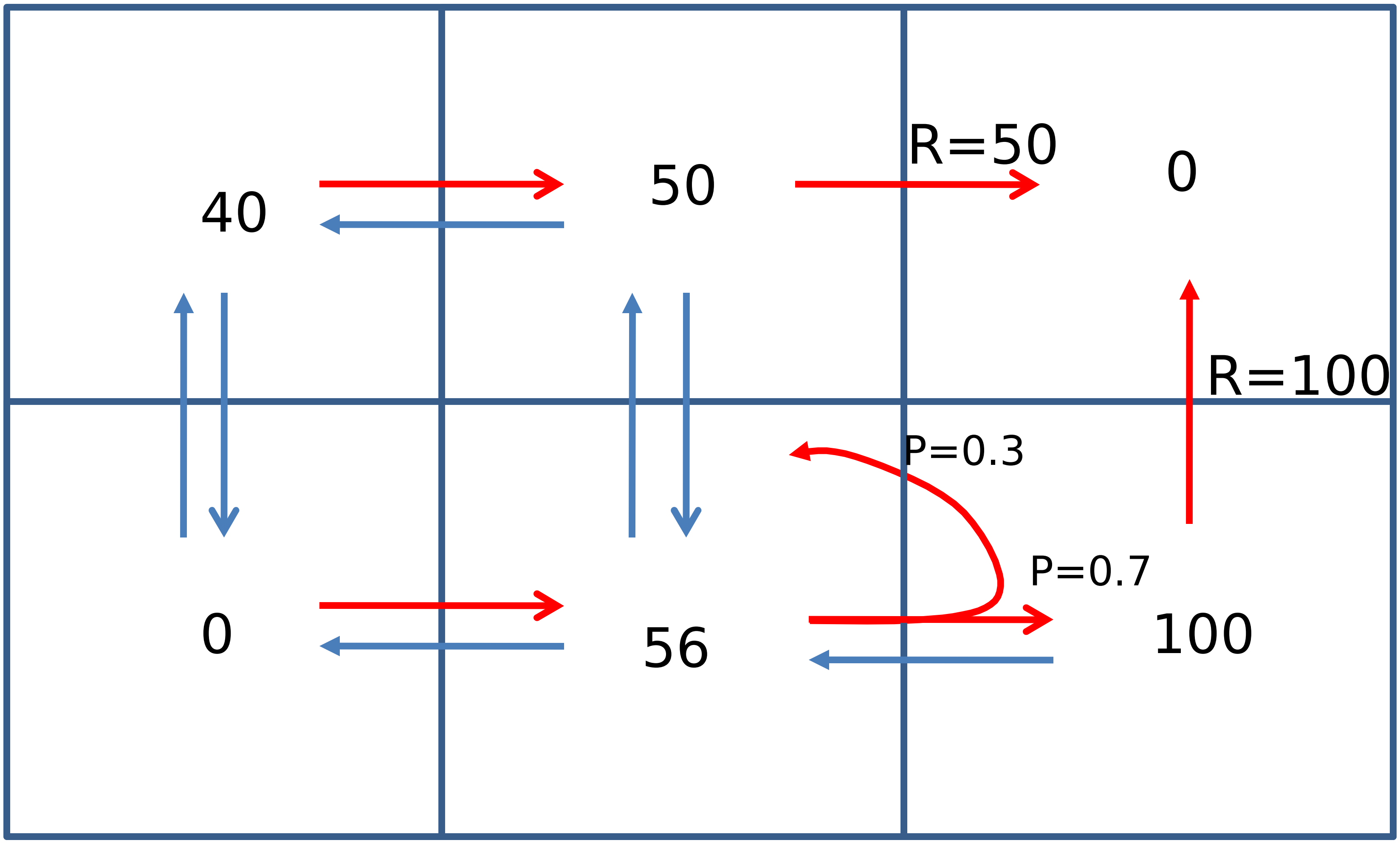


$\mathbf{V_1}$

Non-deterministic World

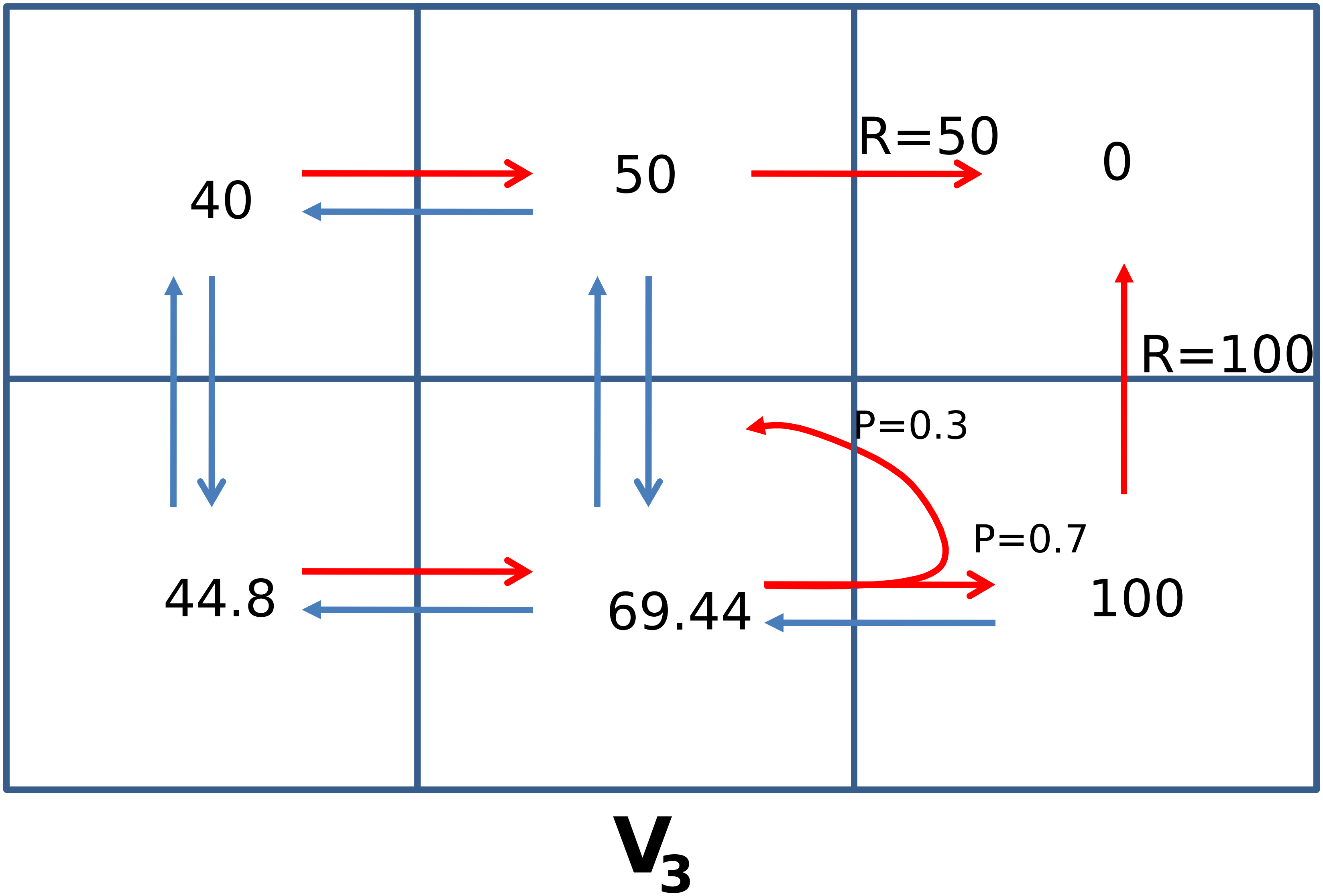


Non-deterministic World



V_2

Non-deterministic World



Value Iteration

$$V_\pi^{t+1}(x) = \underbrace{R(x, \pi(x))}_{\text{Immediate reward of following policy}} + \underbrace{\gamma \sum_{x'} P(x'|x, a = \pi(x)) V_\pi^t(x')}_{\text{Discounted future reward}}$$

Find BEST Policy

Ask the question in a slightly different way.
What is the Value of the Best Policy?

$$V_\pi^{t+1}(x) = \underbrace{R(x, \pi(x))}_{\text{Immediate reward of following policy}} + \underbrace{\gamma \sum_{x'} P(x' | x, a = \pi(x)) V_\pi^t(x')}_{\text{Discounted future reward}}$$

$$V^\star(x) = \max_{a'} \underbrace{R(x, a')}_{\text{Immediate reward of following policy}} + \underbrace{\gamma \sum_{x'} P(x' | x, a = a') V^\star(x')}_{\text{Discounted future reward}}$$

Find BEST Policy

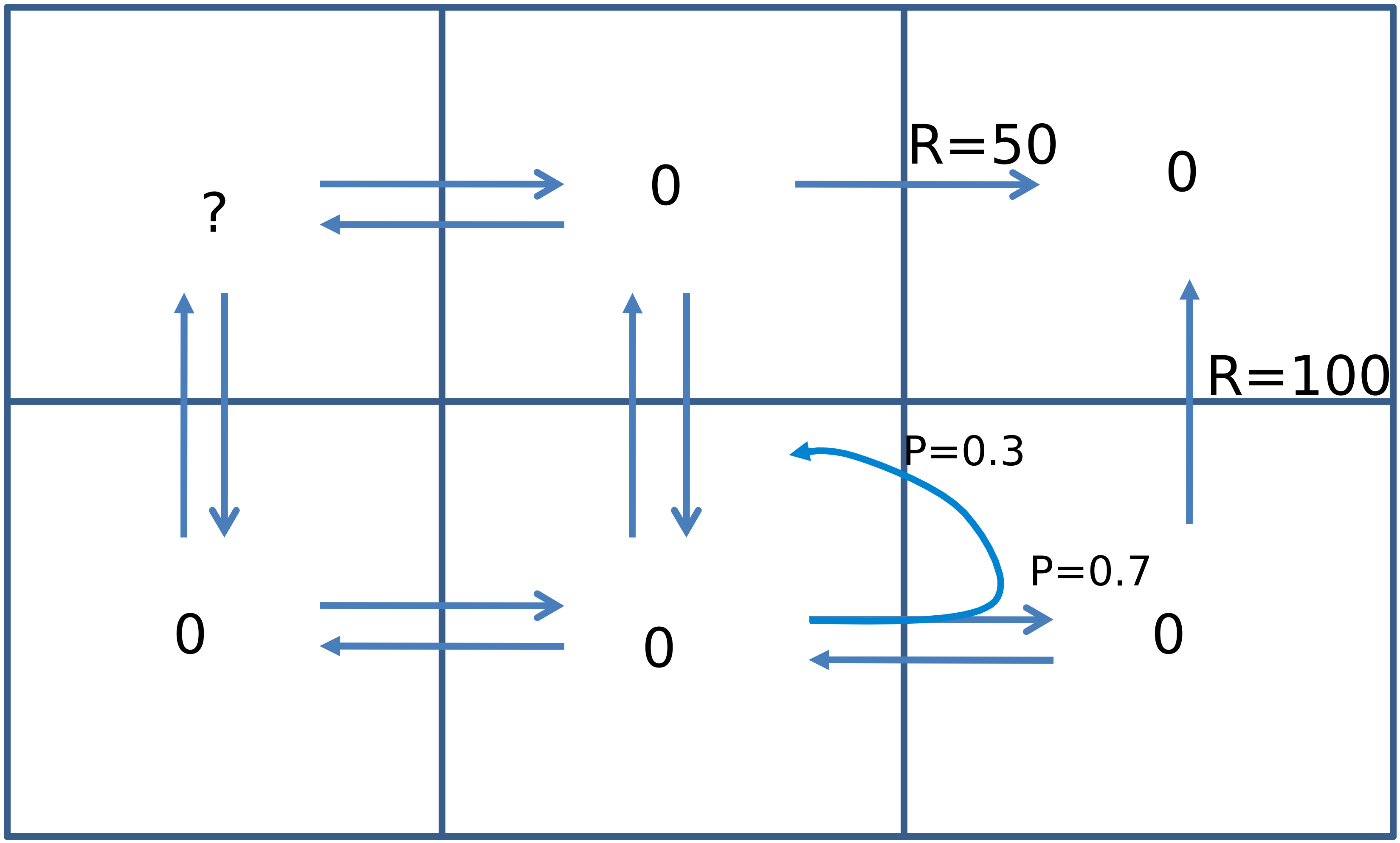
What is the Value of the Best Policy?

$$V^\star(x) = \max_{a'} \underbrace{R(x,a')}_{\text{Immediate reward of following policy}} + \underbrace{\gamma \sum_{x'} P(x'|x,a=a')V^\star(x')}_{\text{Discounted future reward}}$$

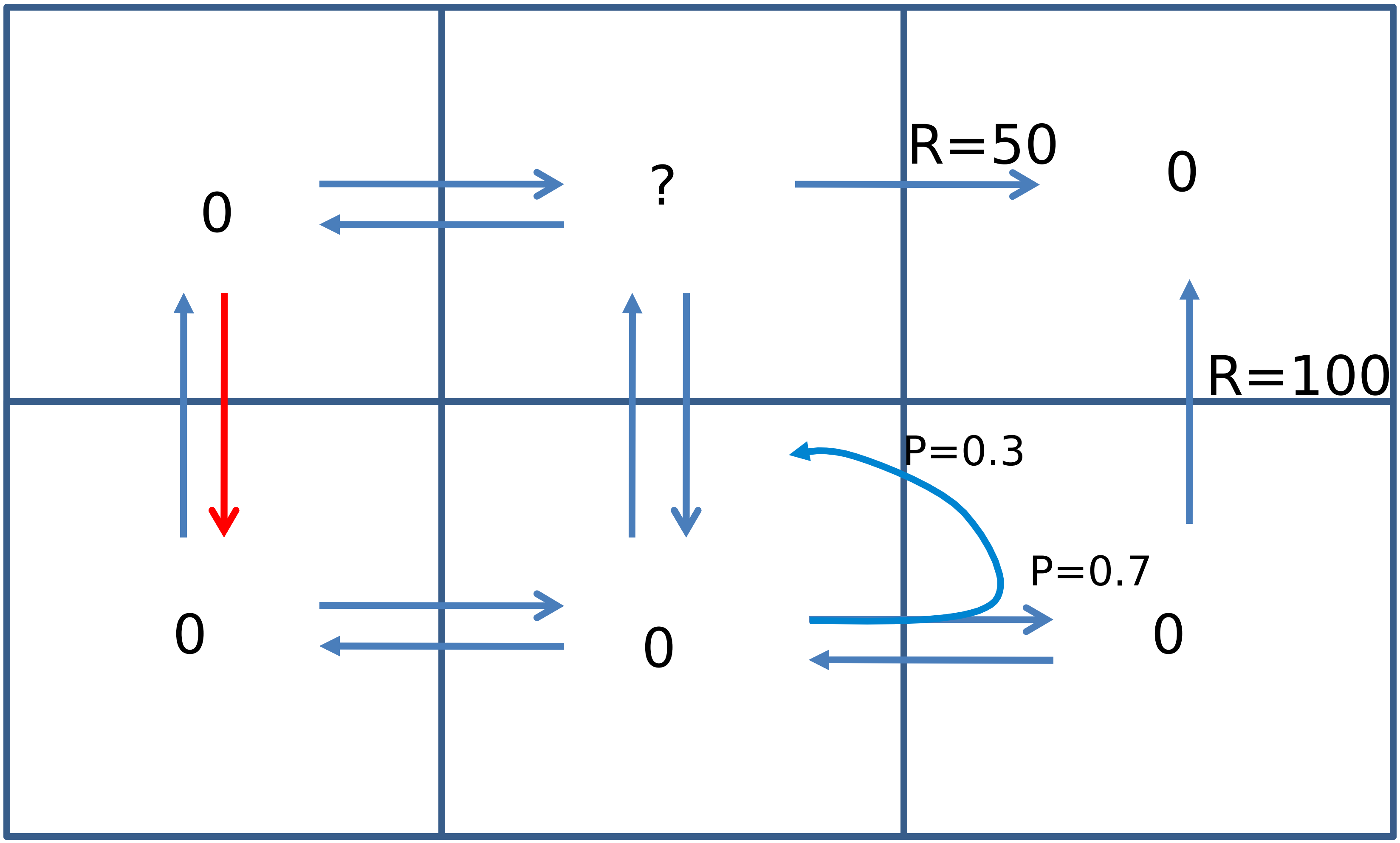
$$\pi^\star(x) = argmax_{a'} R(x,a') + \gamma \sum_{x'} P(x'|x,a=a')V^\star(x')$$

The optimal policy is optimal at every state!

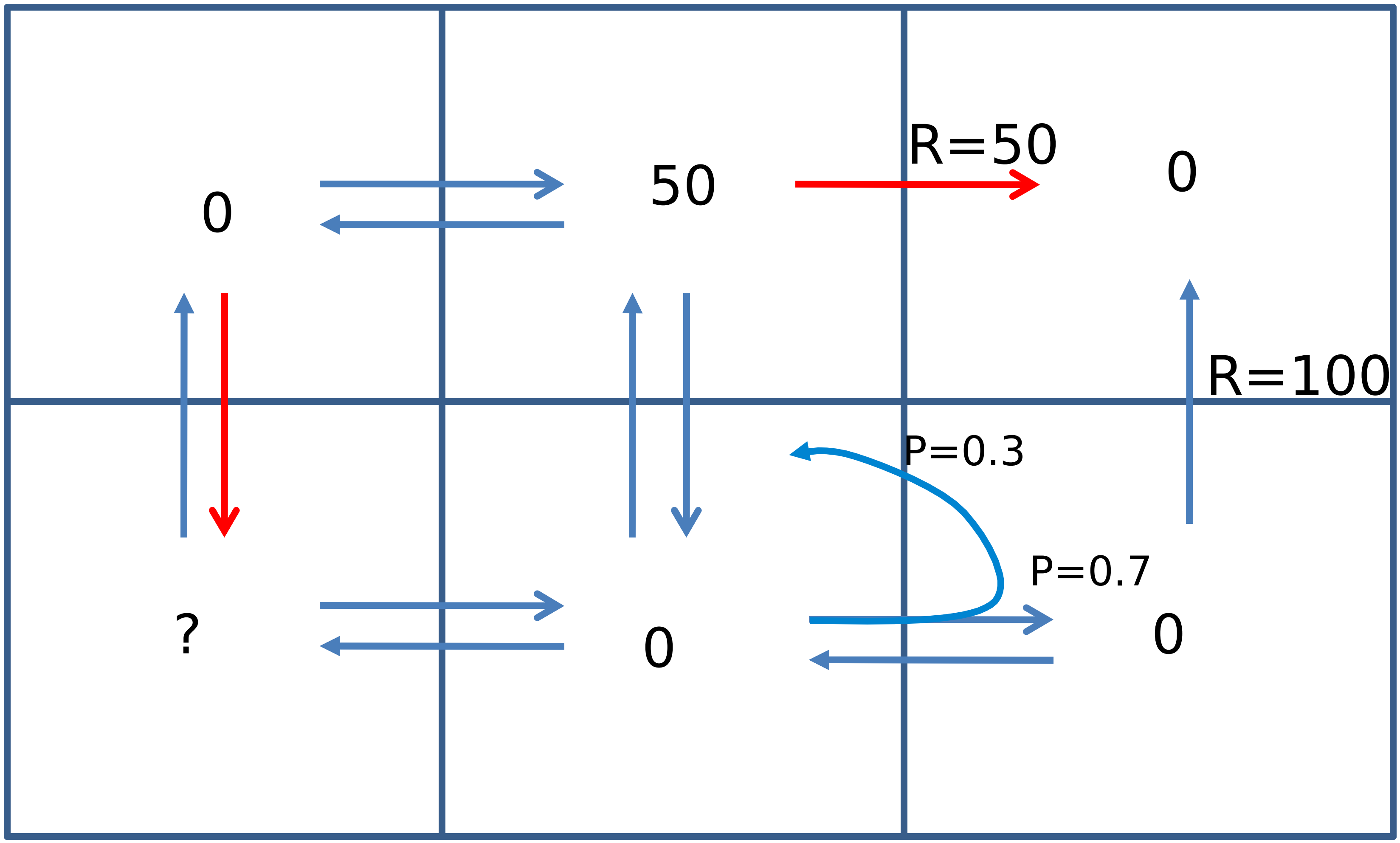
Policy Learning Example



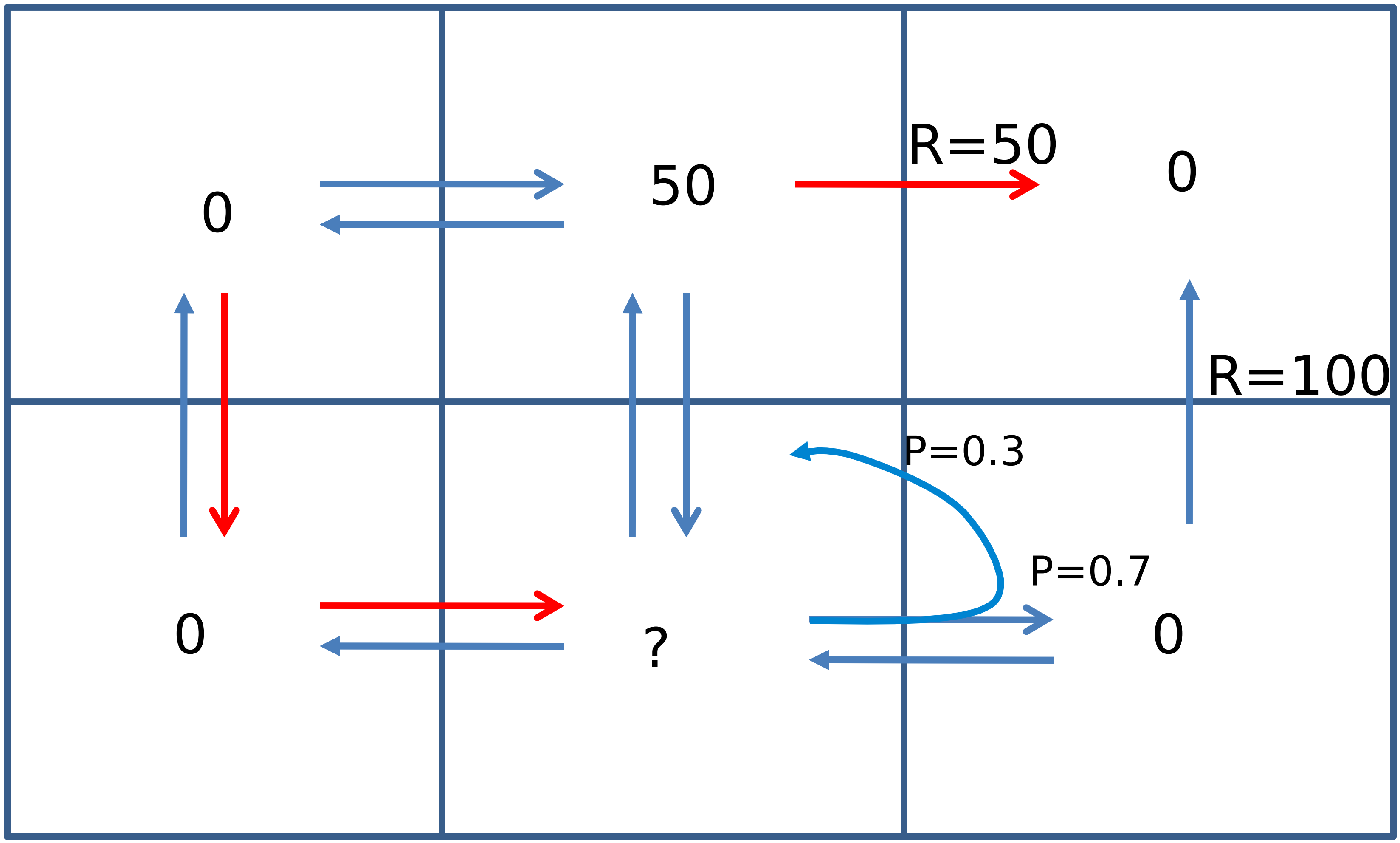
Policy Learning Example



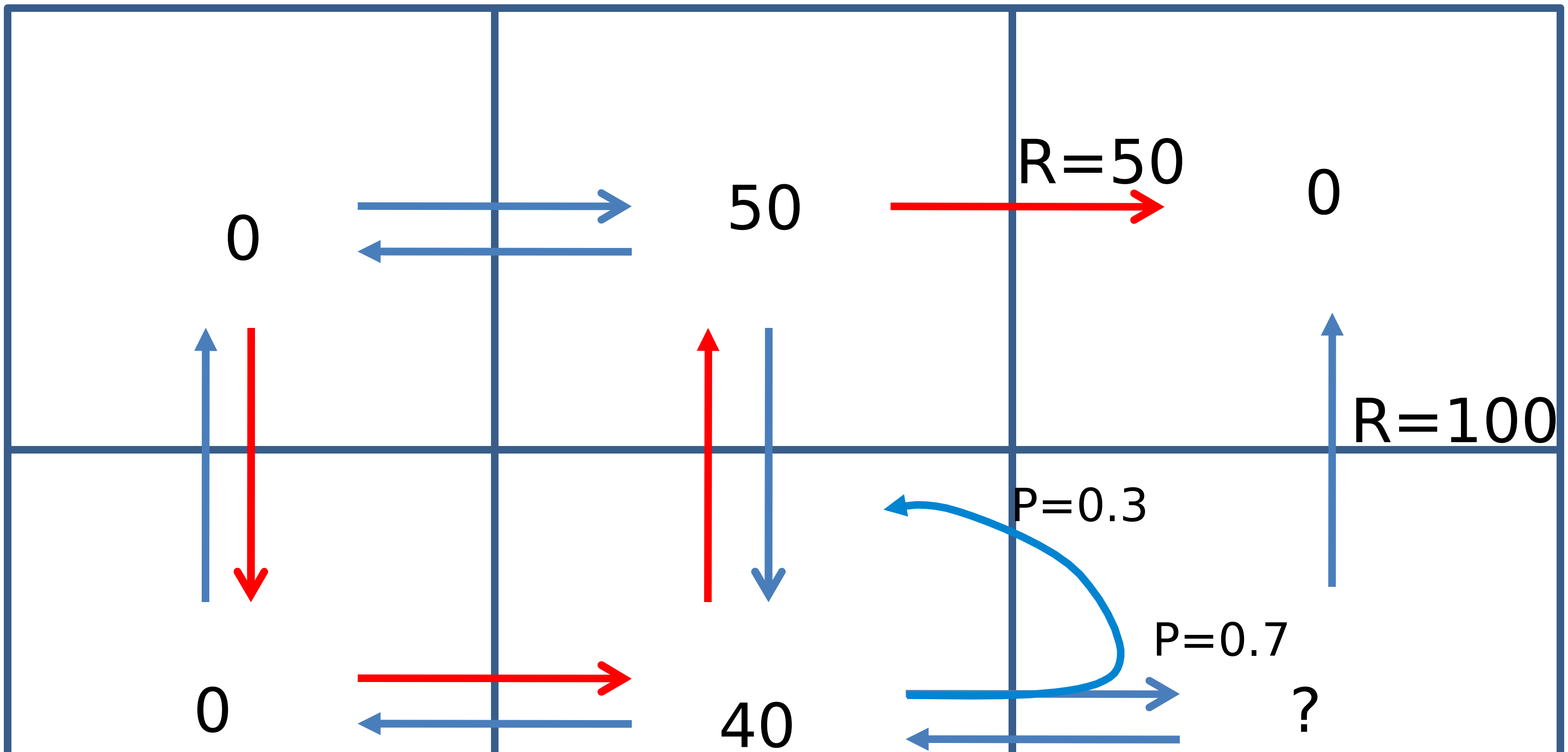
Policy Learning Example



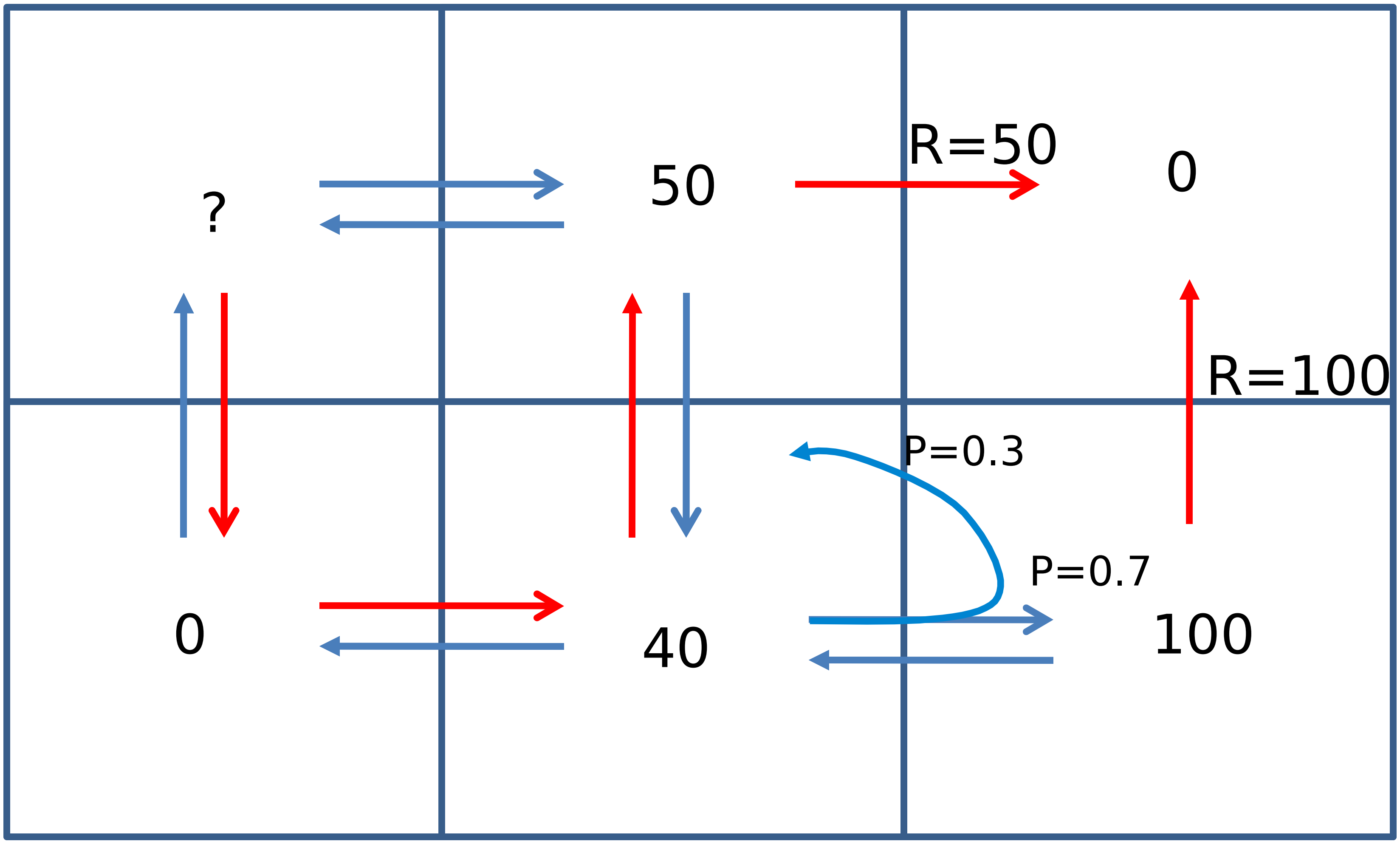
Policy Learning Example



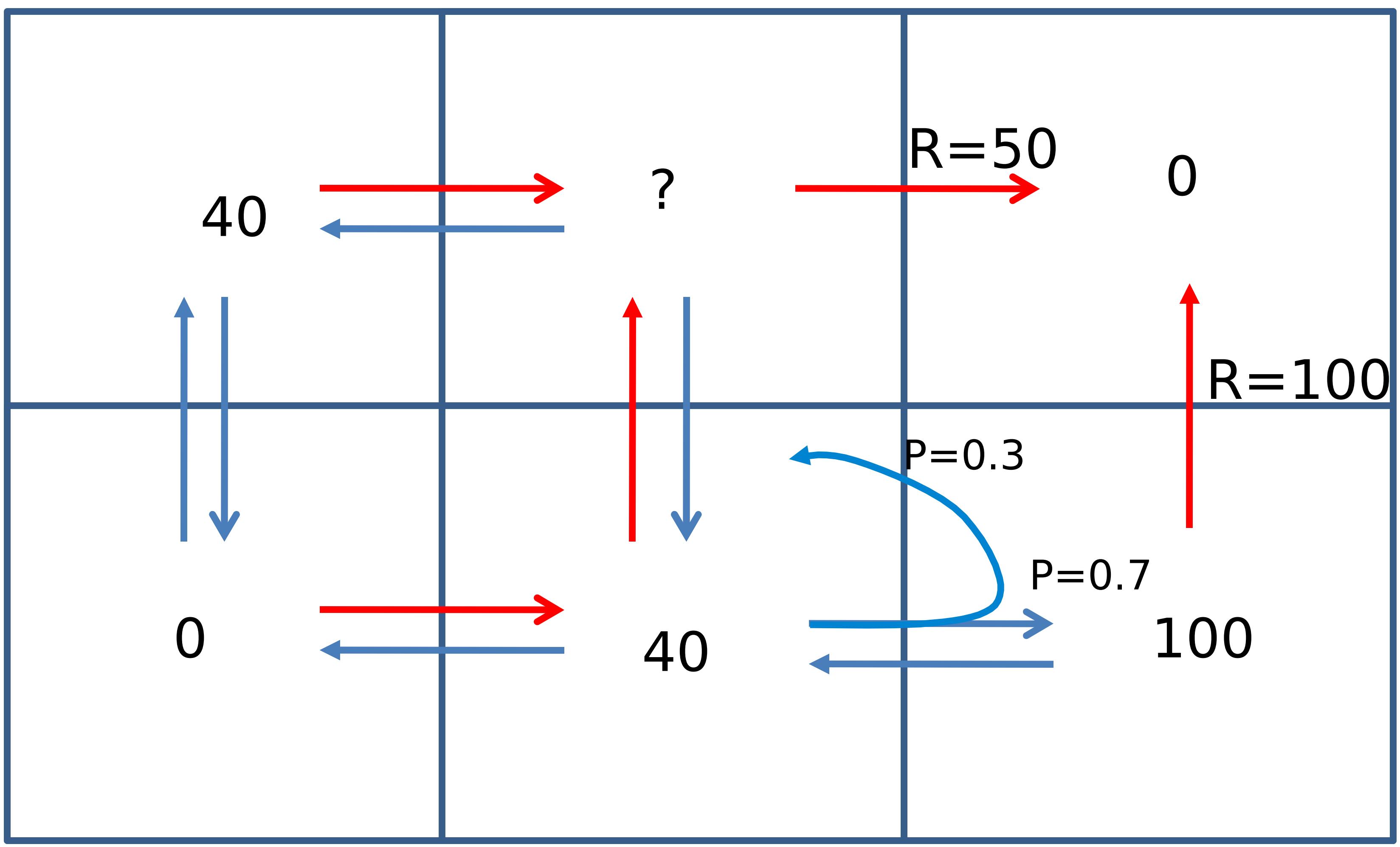
Policy Learning Example



Policy Learning Example



Policy Learning Example



Backgammon

Learning task:

- chose move at arbitrary board states

Training signal:

- final win or loss

Training:

- played 300,000 games against itself

Algorithm:

- reinforcement learning + neural network

Result:

- World-class Backgammon player

Something is wrong here...

Backgammon

Dealing with huge state spaces

Estimate V*(x) instead of *Π*(x)

Approximate V*(x) using a neural net

$$V^\star(x) = \max_{a'} R(x, a') + \gamma \sum_{x'} P(x'|x, a = a')V^\star(x')$$

0 except when you win or lose

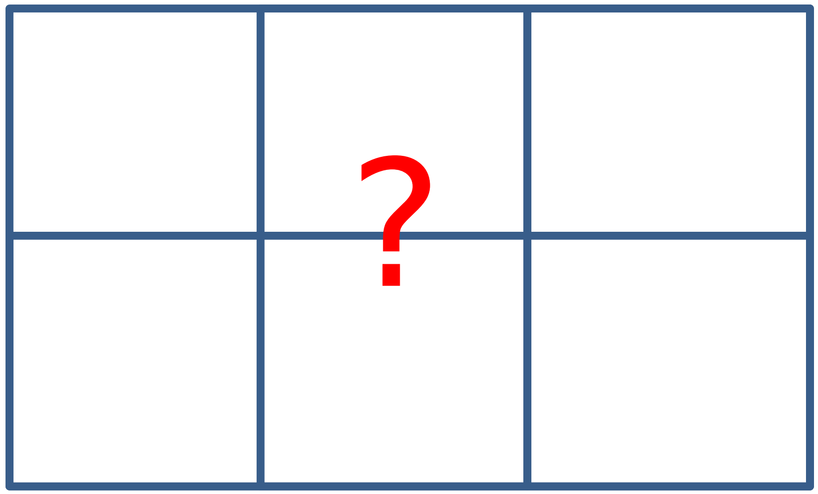
Can be estimated from our current network
In this case, $P(x'|x,a=a')$ is 0 or 1 for all x'

Since V* is a neural net, we can't 'set' the value V*(x)

Instead, use target V*(x) as a training example for the NN

Can't visit every state, so instead play games against yourself to visit the most likely ones.

Unknown World

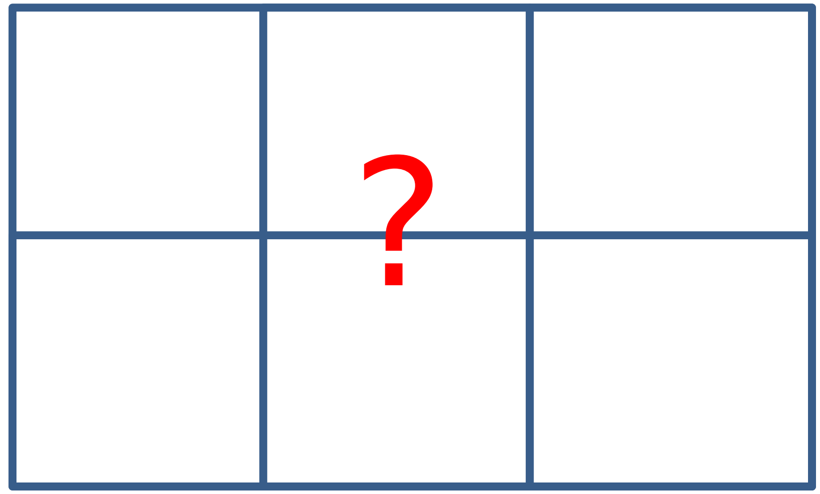


Do not know the transitions.
Do not know the probabilities.
Do not know the rewards.

Only know a state when we actually get there!

Possible Questions.
1: I am in state X. What is the value of following a particular policy?
2: What is the best policy?

Value of Policy



If I know the rewards:

$$V_{\pi}^{t+1}(x) = R(x, \pi(x)) + \gamma \sum_{x'} P(x' | x, a = \pi(x)) V_{\pi}^{t}(x')$$

If I do not know the rewards:

$$V_{\pi}^{t+1}(x_t) = \alpha(r_t + \gamma(V_{\pi}^{t}(x_{t+1}))) + (1-\alpha) V_{\pi}^{t}(x_t)$$

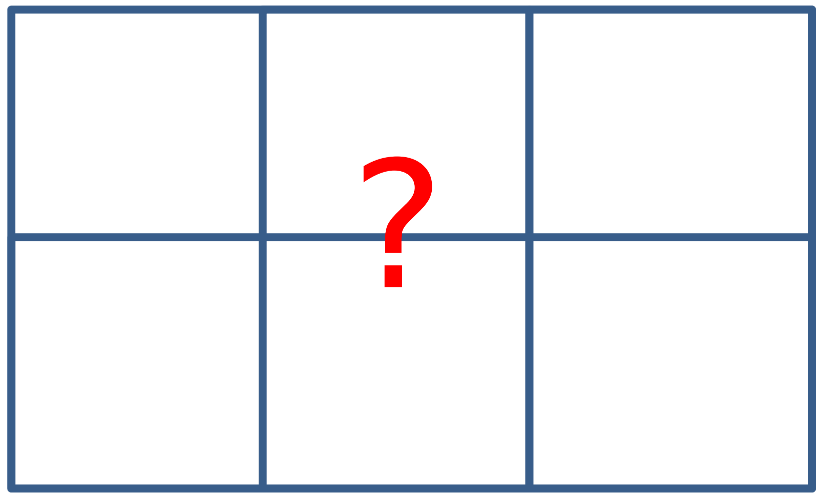
Learning a Policy: Q Learning

Define *Q* which estimates both values and rewards:

$$Q(s,a) \equiv r(s,a) + \gamma V^*(\delta(s,a))$$

Where $\delta(s,a)$ **is the result of taking action** *a* **in state** *s*

Learning a Policy: Q Learning



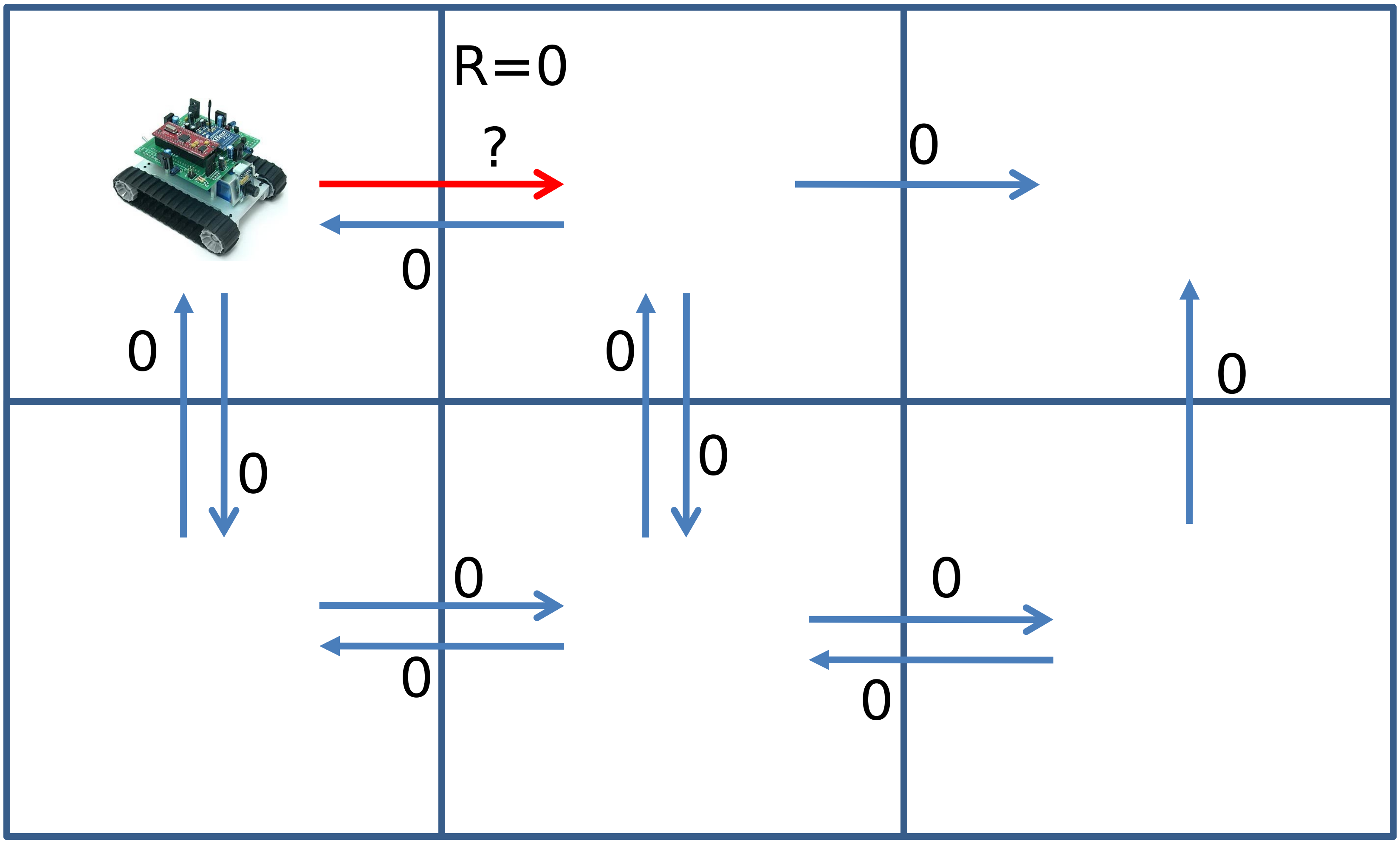
Estimate Q the same way we estimated V

$$V_{\pi}^{t+1}(x_t) = \alpha(r_t + \gamma(V_{\pi}^{t}(x_{t+1}))) + (1-\alpha)V_{\pi}^{t}(x_t)$$

$$Q^{t+1}(x_t, a_t) = \alpha(r_t + \gamma(max_{a'} Q^t(x_{t+1}, a'))) + (1-\alpha)Q^t(x_t, a_t)$$

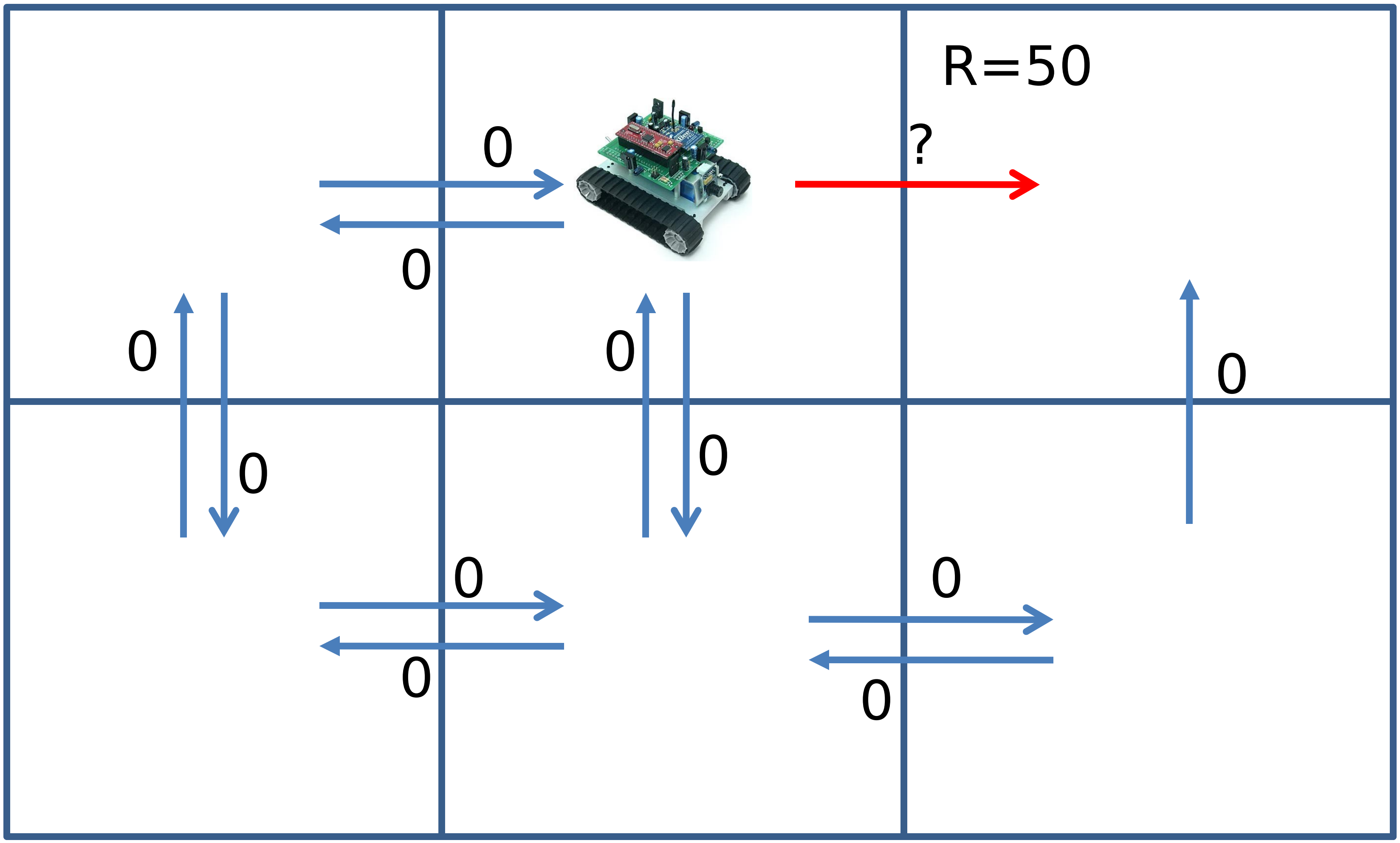
Q Learning Example

$$\gamma = .8, \alpha = .5$$



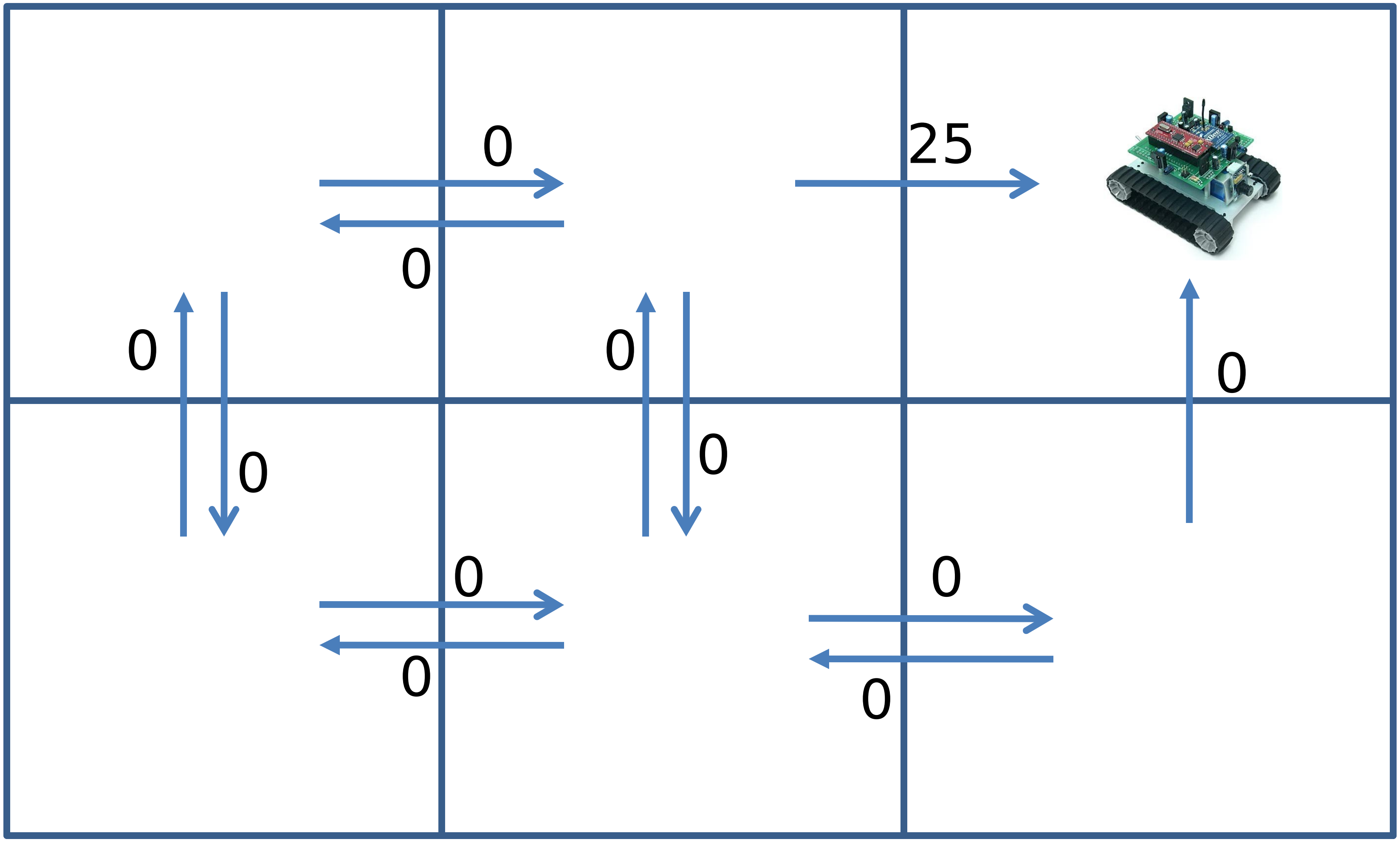
Q Learning Example

$$\gamma = .8, \alpha = .5$$



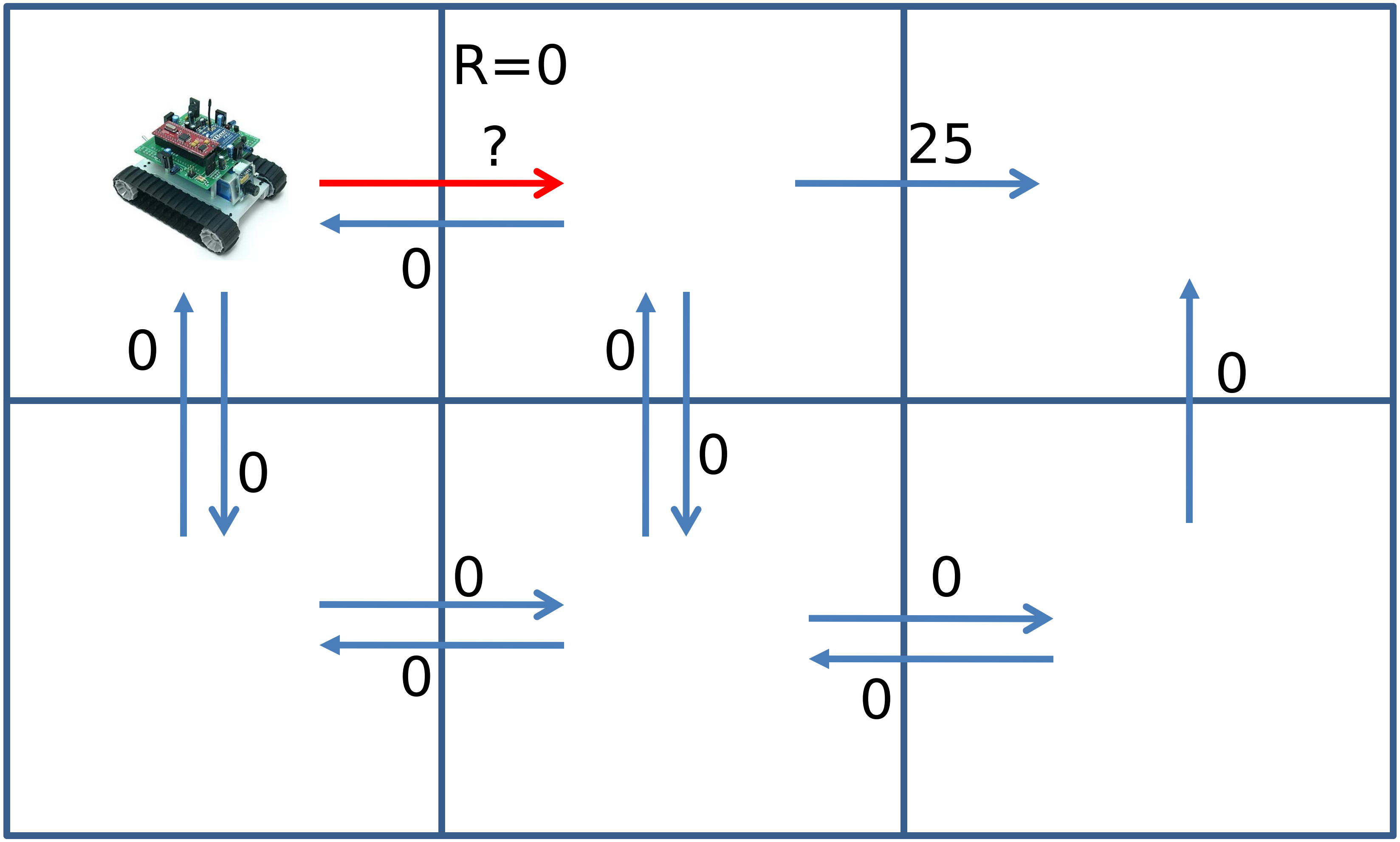
Q Learning Example

$$\gamma = .8, \alpha = .5$$



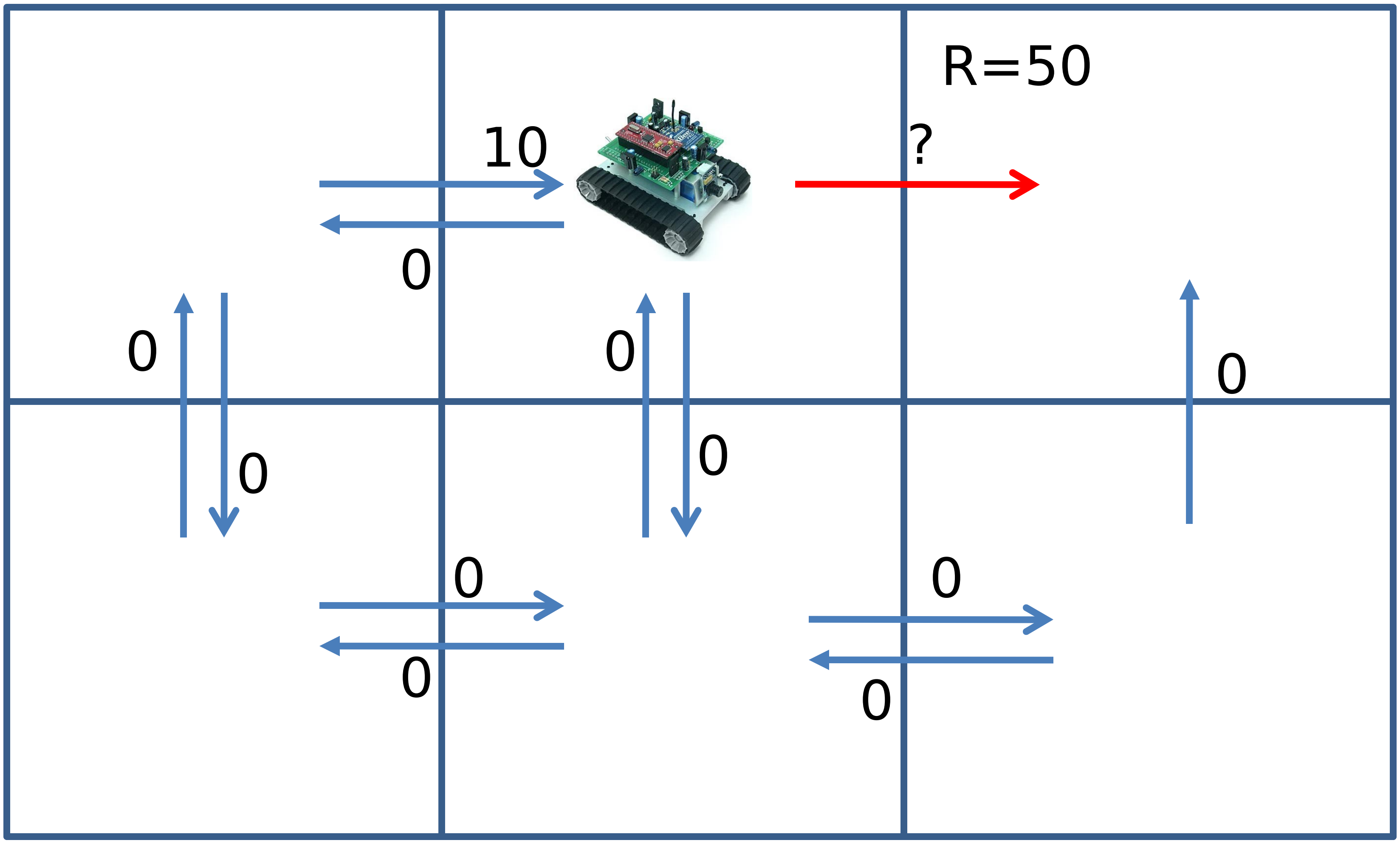
Q Learning Example

$$\gamma = .8, \alpha = .5$$



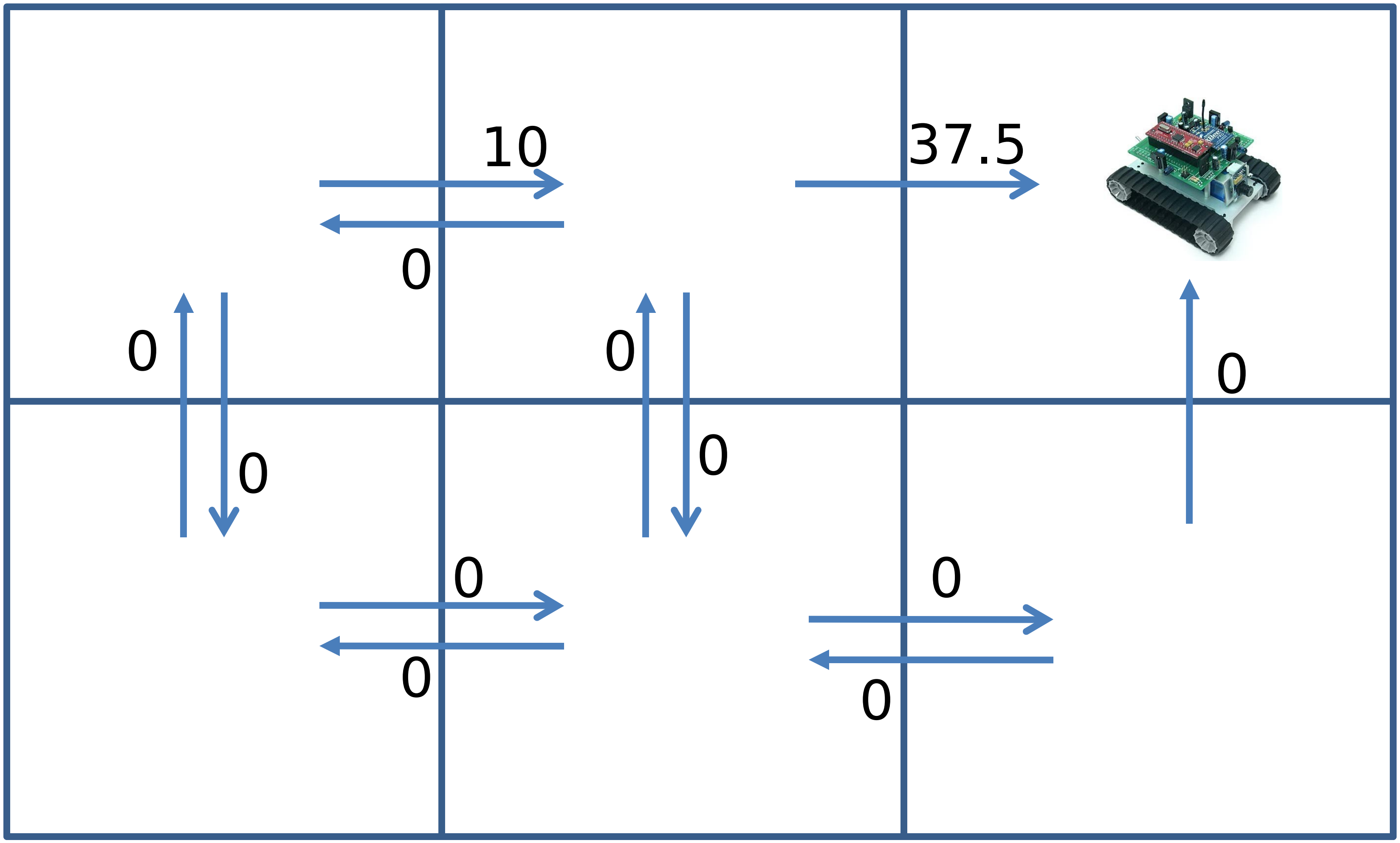
Q Learning Example

$$\gamma = .8, \alpha = .5$$



Q Learning Example

$$\gamma = .8, \alpha = .5$$



$\hat{Q}$ converges to Q. Consider case of deterministic world where see each $\langle s, a \rangle$ visited infinitely often.

Proof: Define a full interval to be an interval during which each $\langle s, a \rangle$ is visited. During each full interval the largest error in $\hat{Q}$ table is reduced by factor of γ

Let $\hat{Q}_n$ be table after n updates, and Δ_n be the maximum error in $\hat{Q}_n$; that is

$$\Delta_n = \max_{s,a} |\hat{Q}_n(s,a) - Q(s,a)|$$

For any table entry $\hat{Q}_n(s,a)$ updated on iteration $n+1$, the error in the revised estimate $\hat{Q}_{n+1}(s,a)$ is

$$\begin{aligned}
|\hat{Q}_{n+1}(s,a) - Q(s,a)| &= |(r + \gamma \max_{a'} \hat{Q}_n(s',a')) \\
&\qquad -(r + \gamma \max_{a'} Q(s',a'))| \\
&= \gamma |\max_{a'} \hat{Q}_n(s',a') - \max_{a'} Q(s',a')| \\
&\leq \gamma \max_{a'} |\hat{Q}_n(s',a') - Q(s',a')| \\
&\leq \gamma \max_{s'',a'} |\hat{Q}_n(s'',a') - Q(s'',a')| \\
|\hat{Q}_{n+1}(s,a) - Q(s,a)| &\leq \gamma \Delta_n
\end{aligned}$$

Use general fact:

$$|\max_a f_1(a) - \max_a f_2(a)| \leq \max_a |f_1(a) - f_2(a)|$$