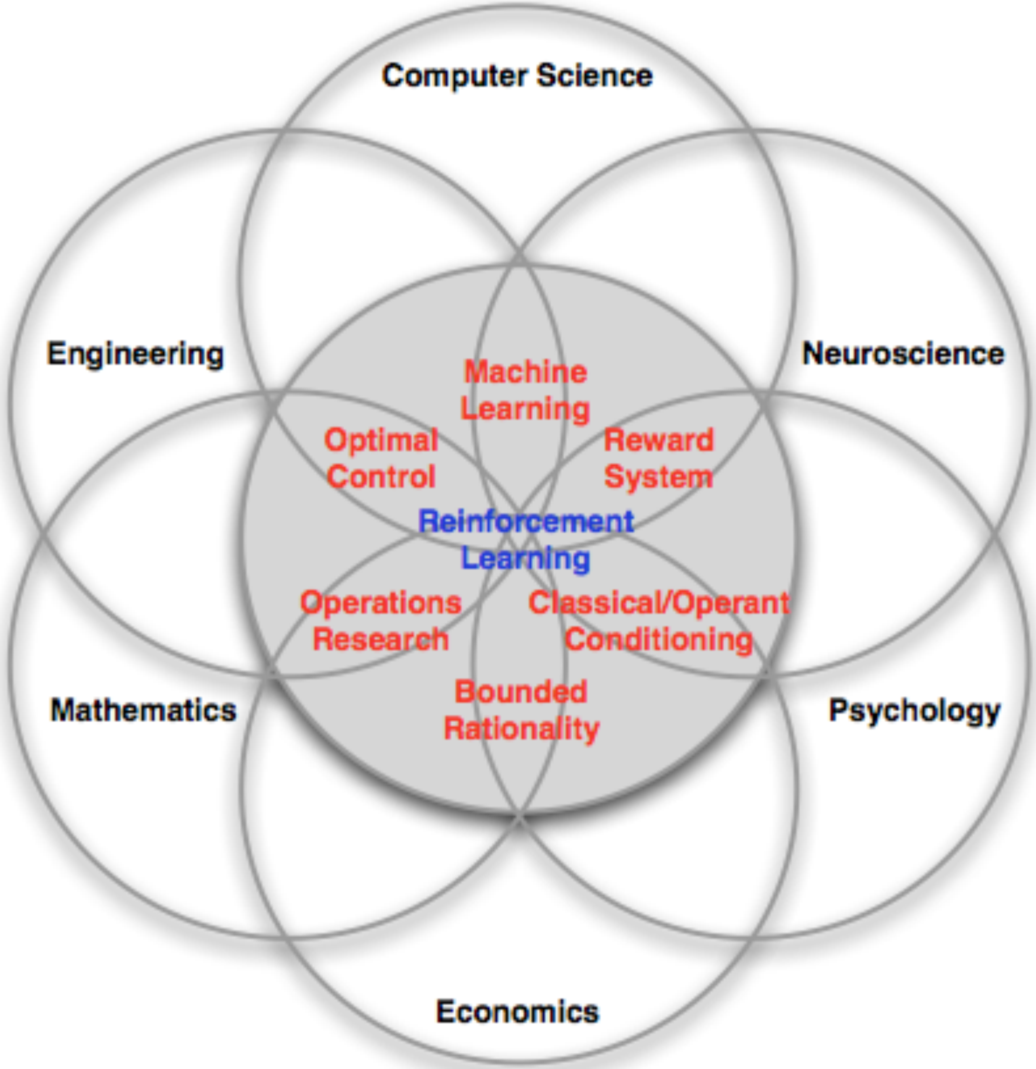
Introduction to Reinforcement Learning

RL

Overview of topics

- About Reinforcement Learning
- The Reinforcement Learning Problem
- Inside an RL agent
- Temporal difference learning

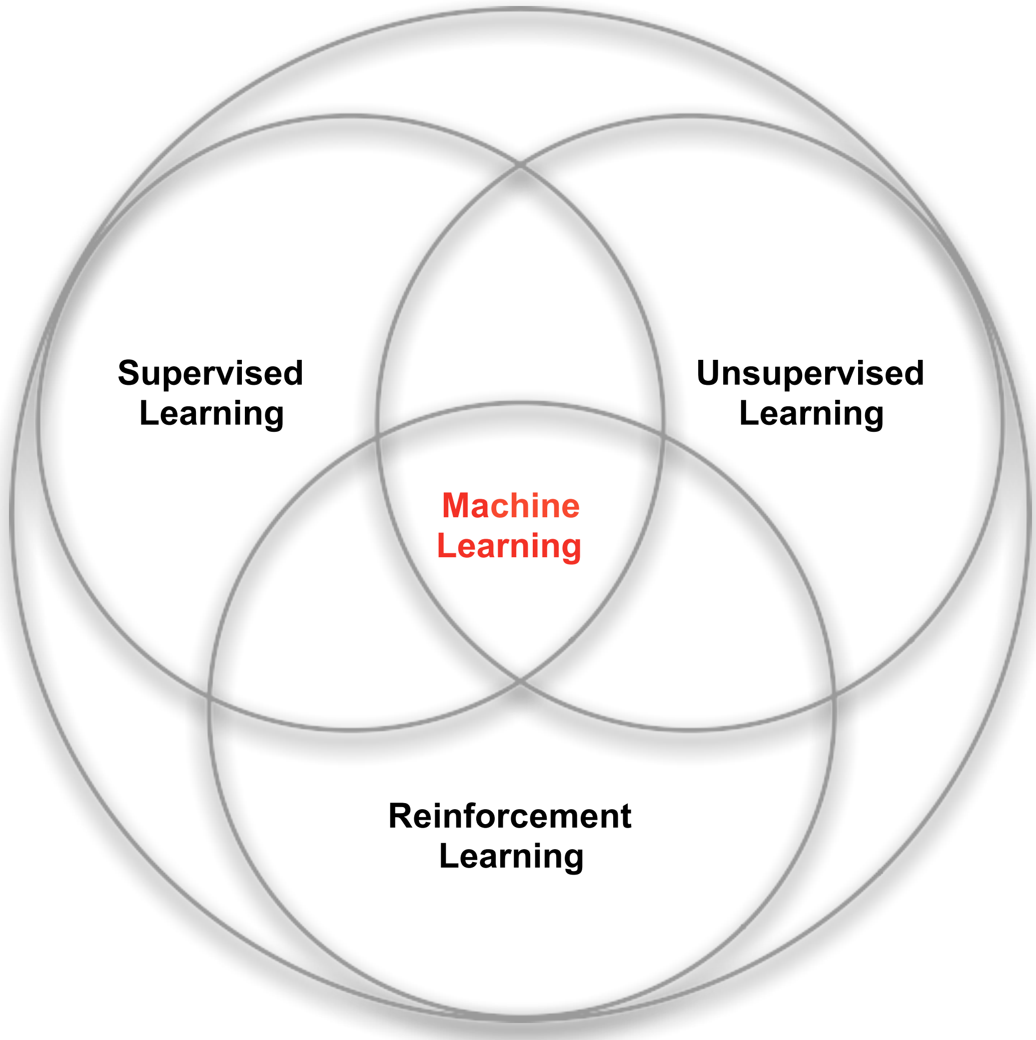
Many faces of Reinforcement Learning



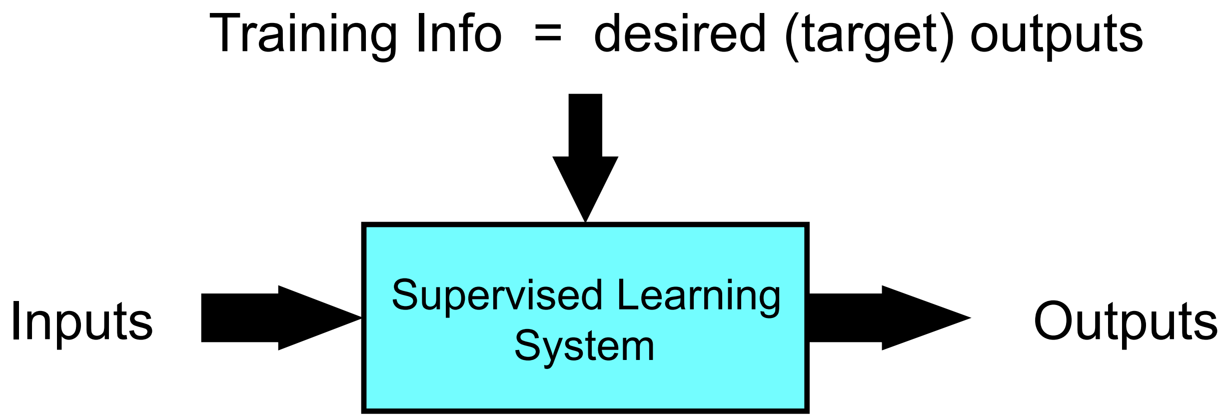
What is Reinforcement Learning?

- Learning from interaction
- Goal-oriented learning
- Learning about, from, and while interacting with an external environment
- Learning what to do—how to map situations to actions—so as to maximize a numerical reward signal

Branches of AI

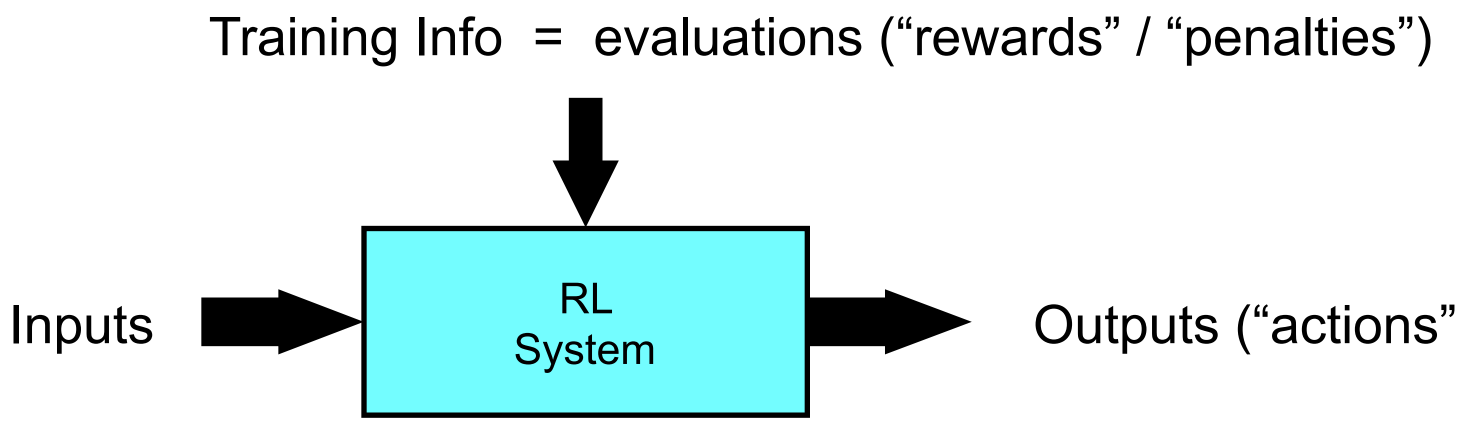


Supervised Learning



Error = (target output – actual output)

Reinforcement Learning



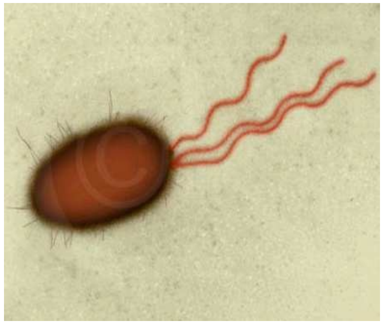
Objective: get as much reward as possible

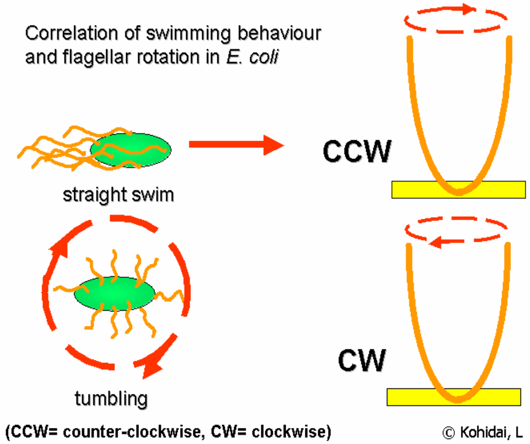
Recipe for creative behavior: explore & exploit

- Creativity: finding a new approach / solution / …
  - Exploration (random / systematic / …)
  - Evaluation (utility = expected rewards)
  - Selection (ongoing behavior and learning)

Coli bacteria and creativity

- Escherichia Coli searches for food using trial and error:
  - Choose a random direction by tumbling and then start swimming straight
  - Evaluate progress
  - Continue longer or cancel earlier depending on progress

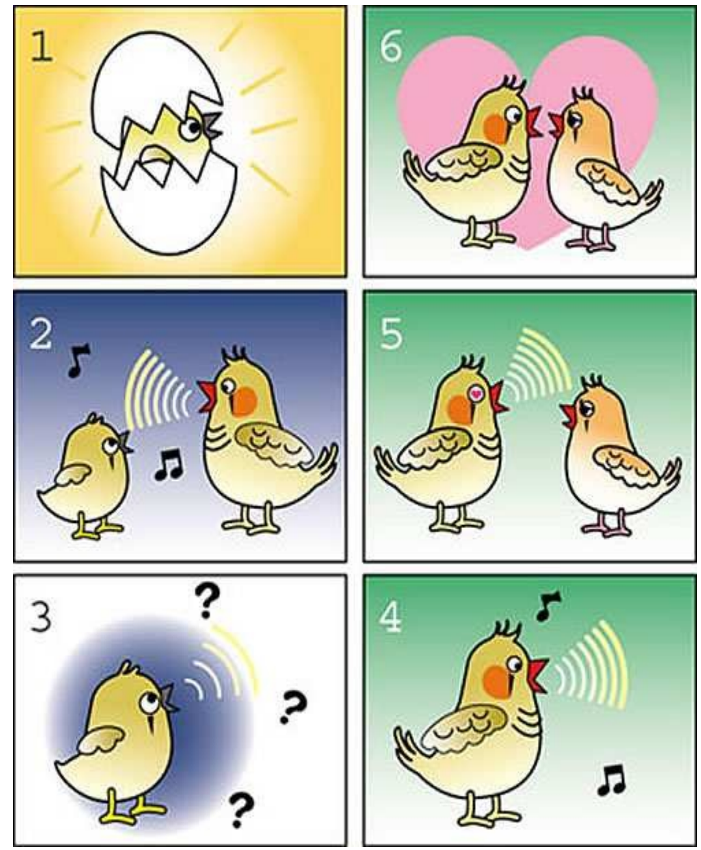




http://biology.about.com/library/weekly/aa081299.htm

Zebra finch: from singing in the shower to performing artist

1. A newborn zebra finch can’ t sing
2. The baby bird listens to father’s song
3. The baby starts to “babble” father’s song as a target template
4. The song develops through trial and error – “singing in the shower”
5. No exploration when singing to a female



http://www.brain.riken.jp/bsi-news/bsinews34/no34/speciale.html

Zebra finch: from singing in the shower to performing artist

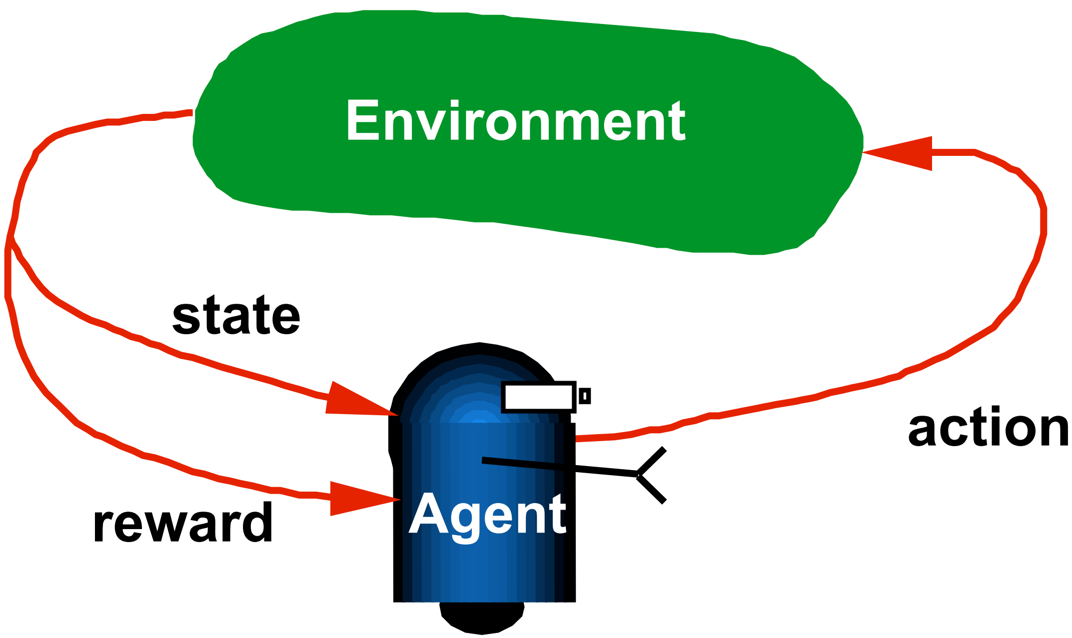
- http://www.youtube.com/watch?v=Md6bsvkauPg

Key Features of RL

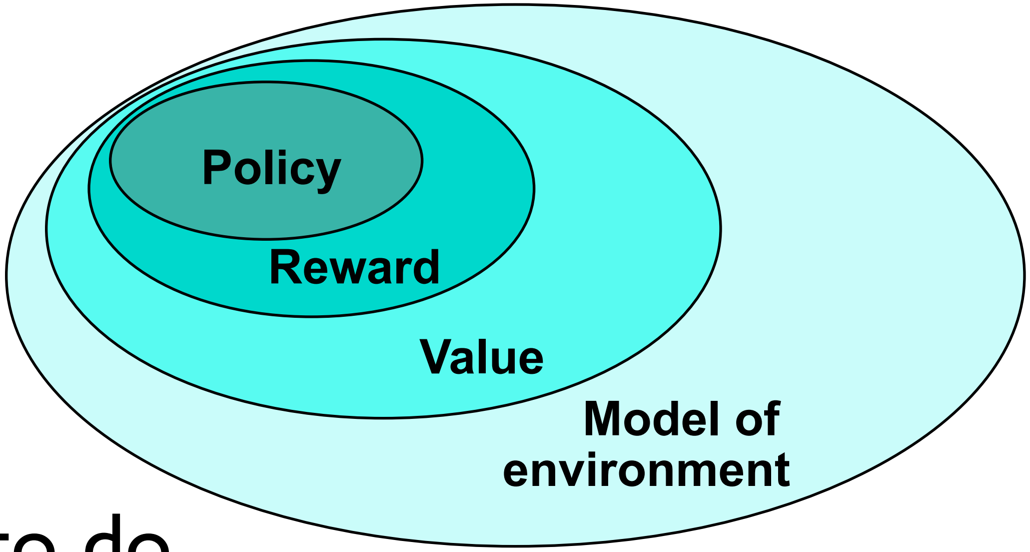
- Learner is not told which actions to take
- Trial-and-Error search
- Possibility of delayed reward (sacrifice short-term gains for greater long-term gains)
- The need to ***explore*** and ***exploit***
- Considers the whole problem of a goal-directed agent interacting with an uncertain environment

Complete Agent

- Temporally situated
- Continual learning and planning
- Object is to ***affect*** the environment
- Environment is stochastic and uncertain

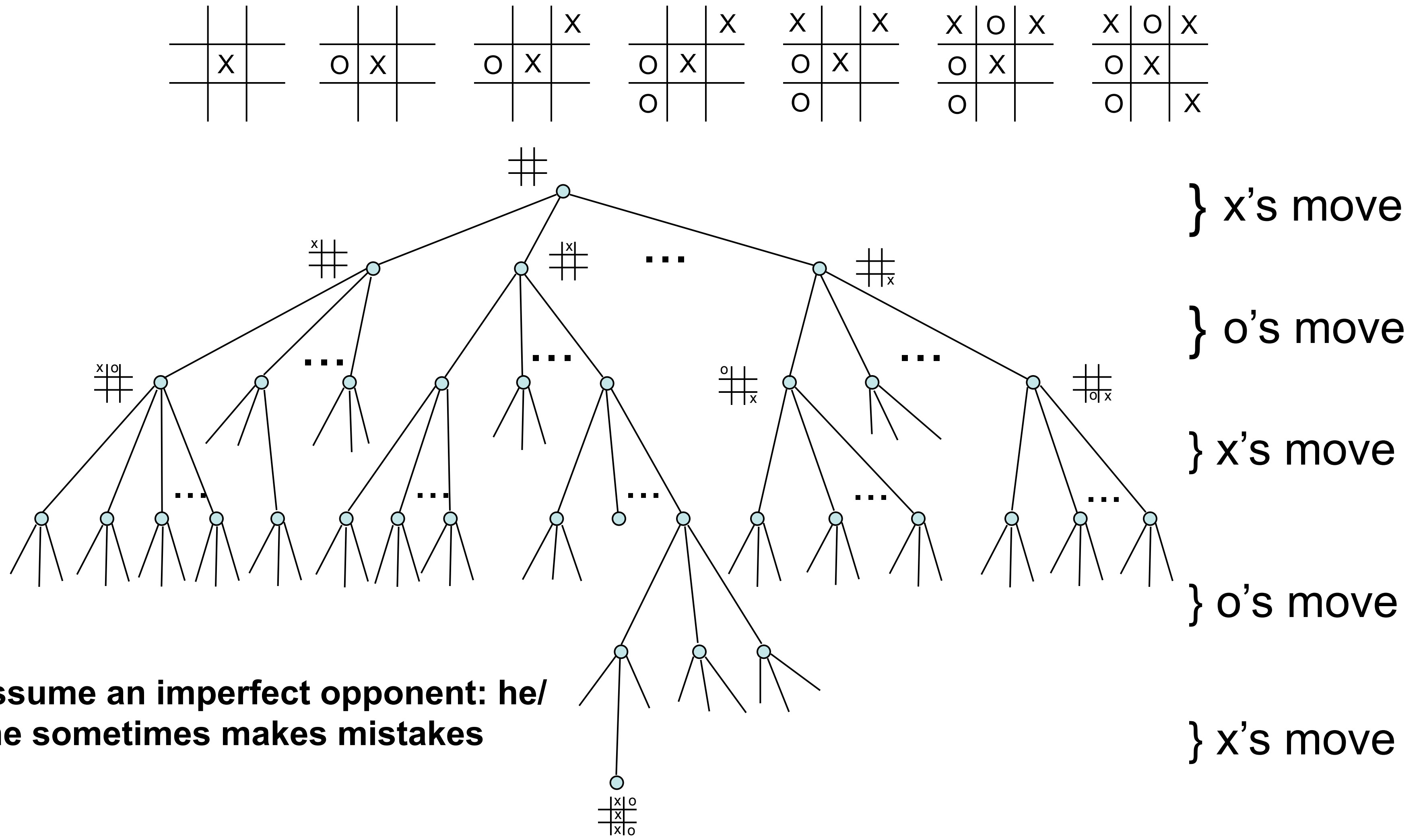


Elements of RL



- **Policy**: what to do
- **Reward**: what is good
- **Value**: what is good because it *predicts* reward
- **Model**: what follows what

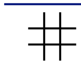
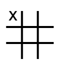
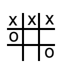
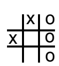
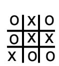
An Extended Example: Tic-Tac-Toe



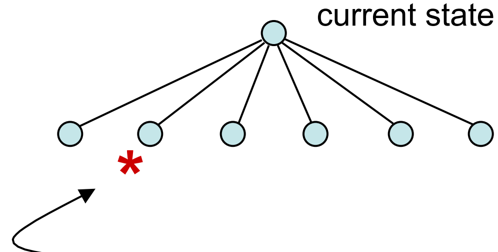
Assume an imperfect opponent: he/ she sometimes makes mistakes

An RL Approach to Tic-Tac-Toe

1. Make a table with one entry per state:

State	$V(s)$ – estimated probability of winning	
	.5	?
	.5	?
⋮	⋮	
	1	win
⋮	⋮	
	0	loss
⋮	⋮	
	0	draw

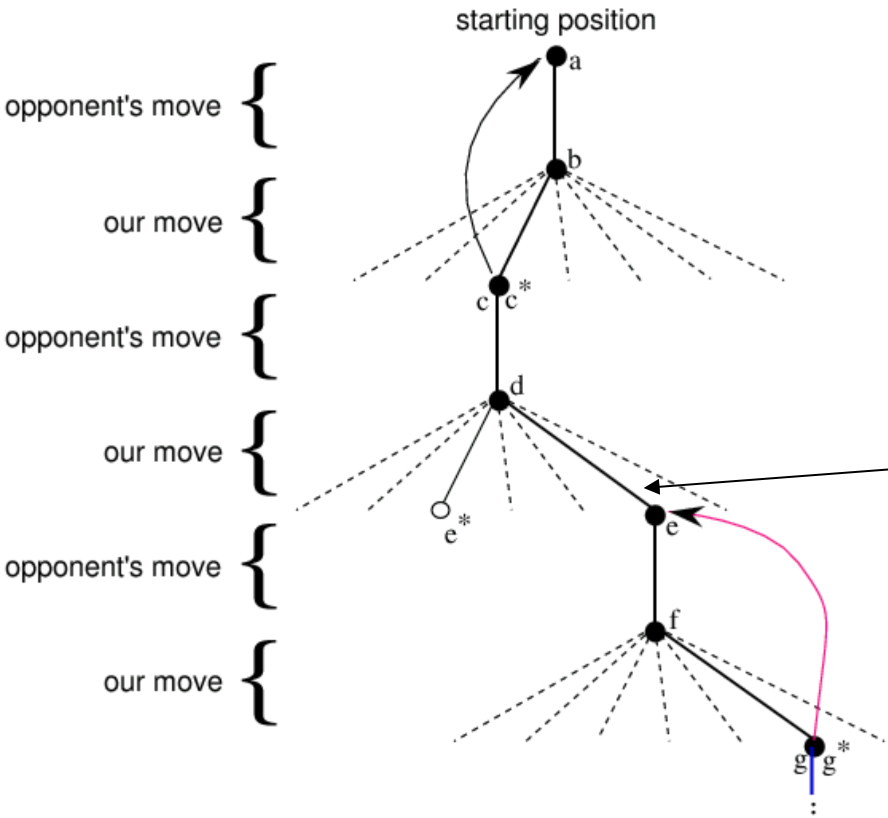
2. Now play lots of games. To pick our moves, look ahead one step:



Just pick the next state with the highest estimated prob. of winning — the largest $V(s)$; a ***greedy*** move.

But 10% of the time pick a move at random; an ***exploratory move***.

RL Learning Rule for Tic-Tac-Toe



s – *the state before our greedy move*

s' – *the state after our greedy move*

We increment each $V(s)$ *toward* $V(s')$ – *a* ***backup*** *:*

$$V(s) \leftarrow V(s) + \alpha\left[V(s') - V(s)\right]$$

a small positive fraction, e.g., $\alpha = .1$

the ***step - size parameter***

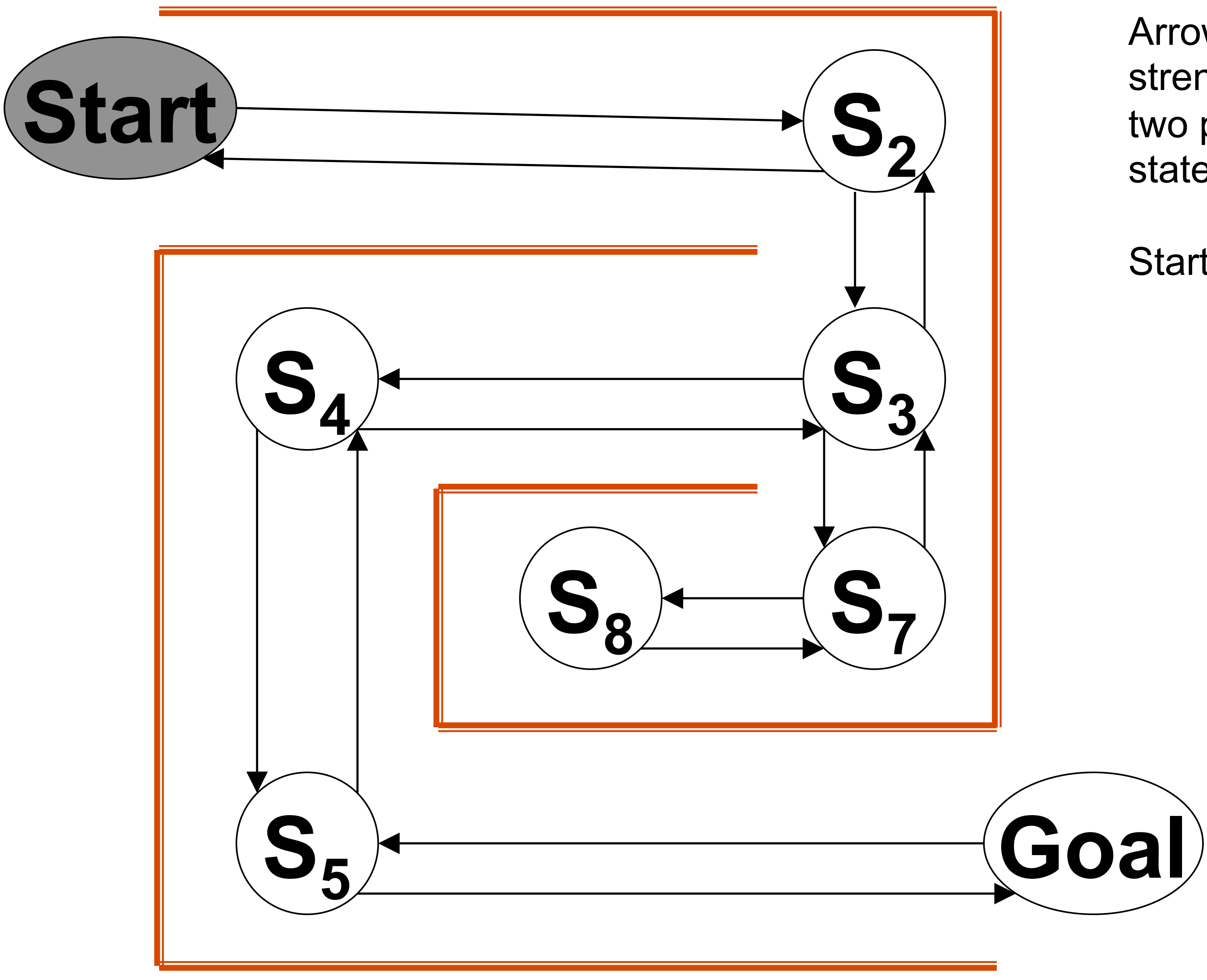
How can we improve this T.T.T. player?

- Take advantage of symmetries
  - representation/generalization
- Do we need “random” moves? Why?
  - Do we always need a full 10%?
- Can we learn from “random” moves?
- …

Temporal difference learning

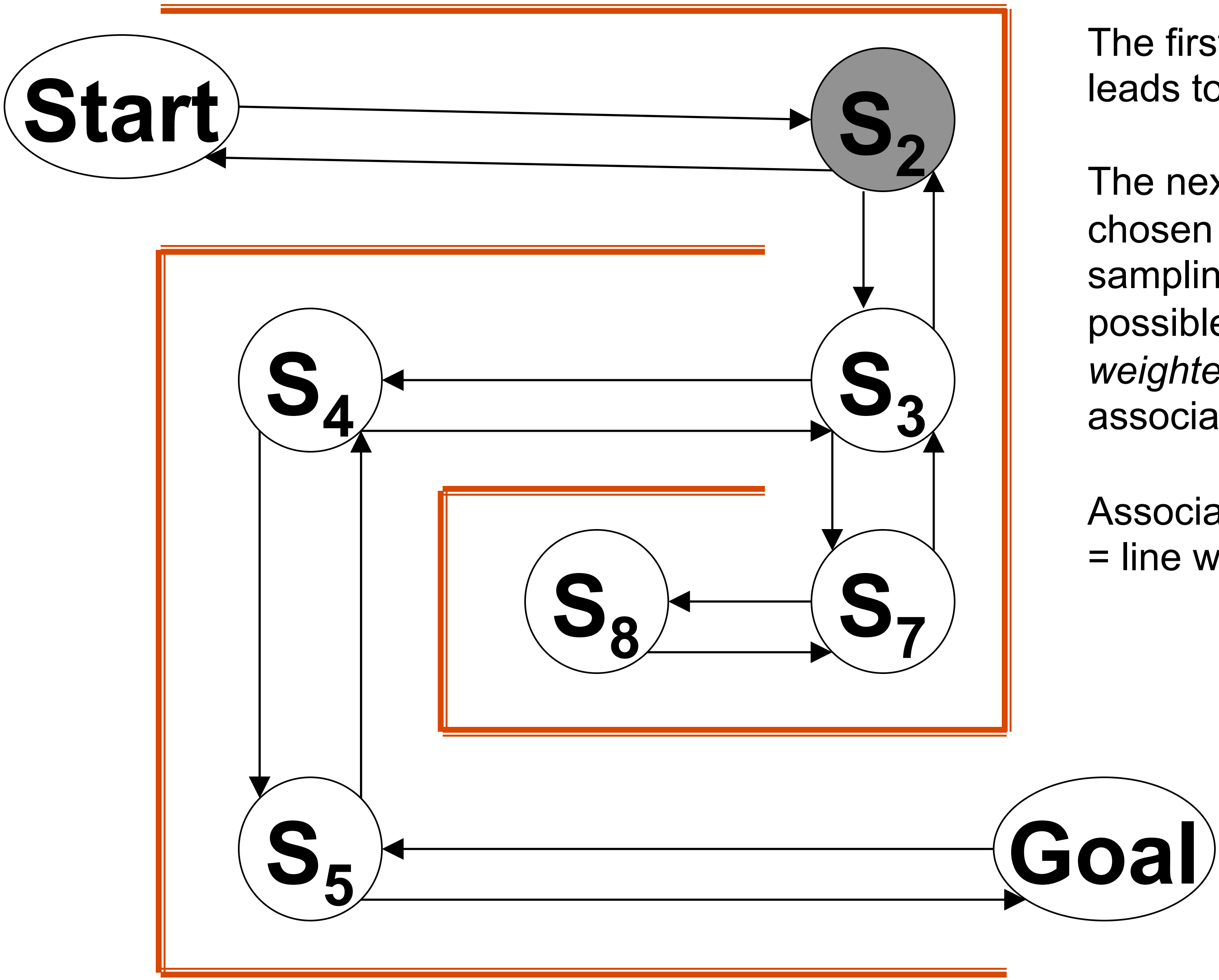
- Solution to temporal credit assignment problem
- Replace the reward signal by the change in expected future reward
  - Prediction moves the rewards from the future as close to the actions as possible
  - Primary reward such as sugar replaced with secondary (or higher order) rewards such as money
  - In the brain, **dopamine** ≈ temporal difference signal
  - Supervised learning is used for channelling information in predictive stimuli to learning

Reinforcement learning example



Arrows indicate strength between two problem states

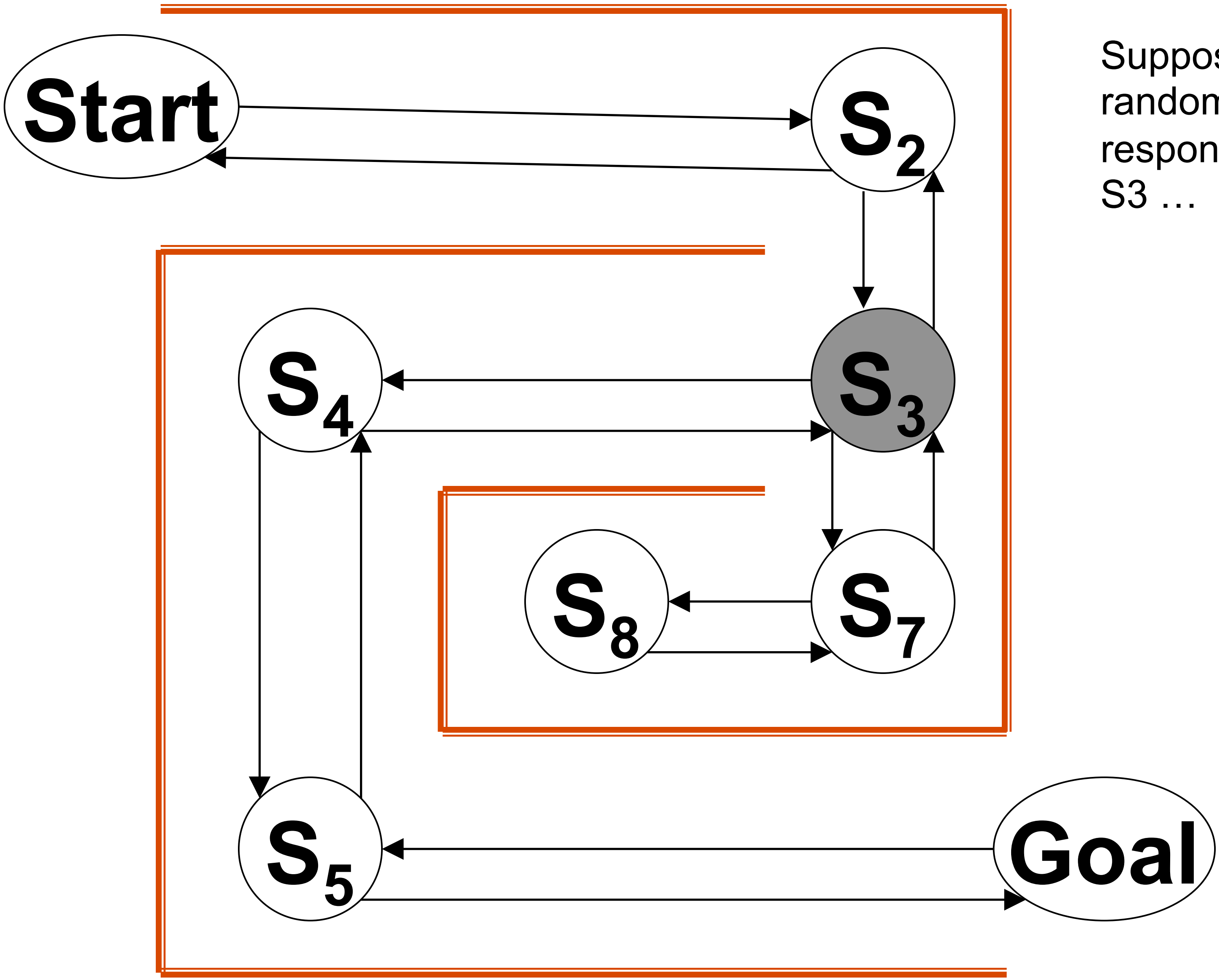
Start maze …



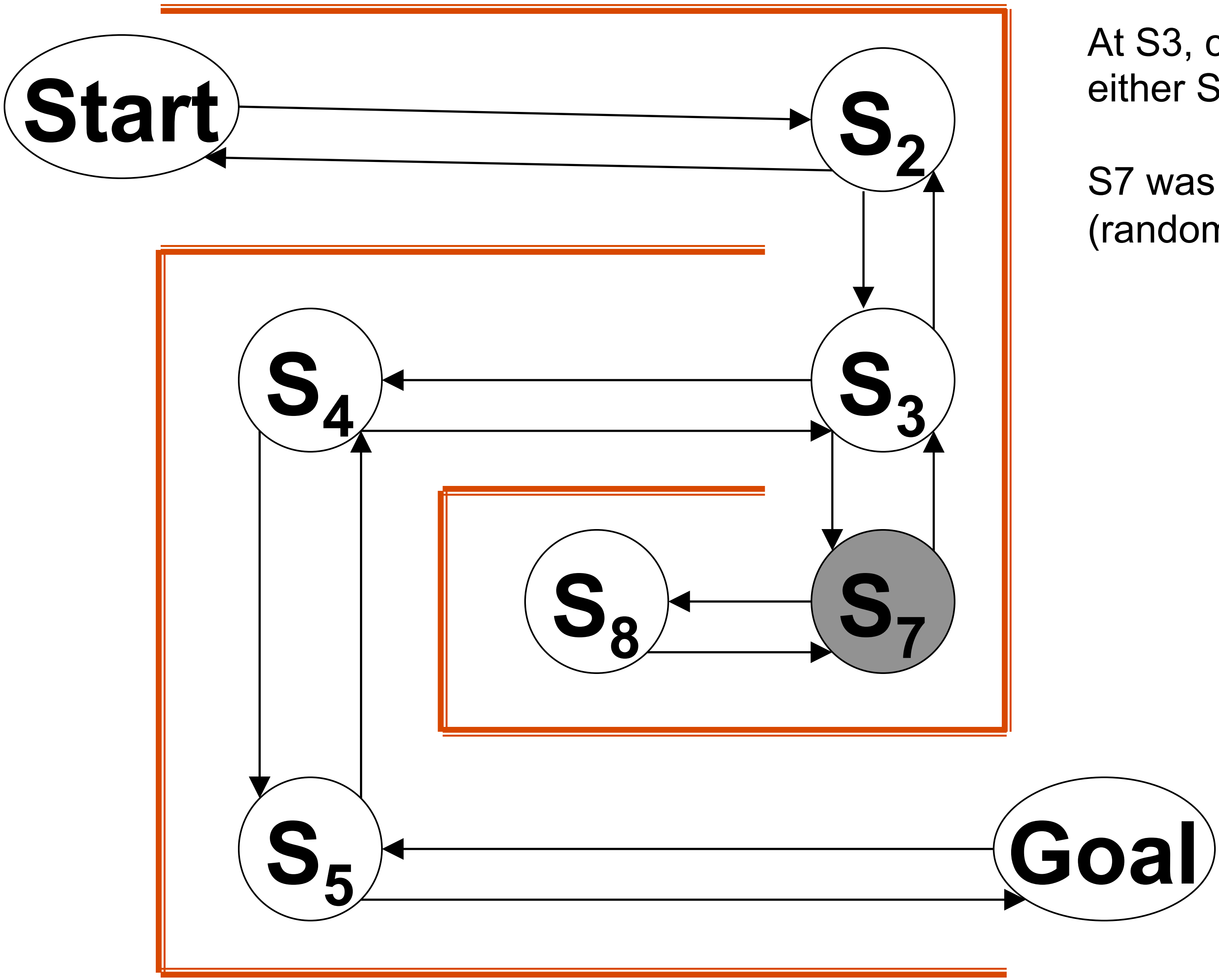
The first response leads to S2 …

The next state is chosen by randomly sampling from the possible next states *weighted* by their associative strength

Associative strength = line width

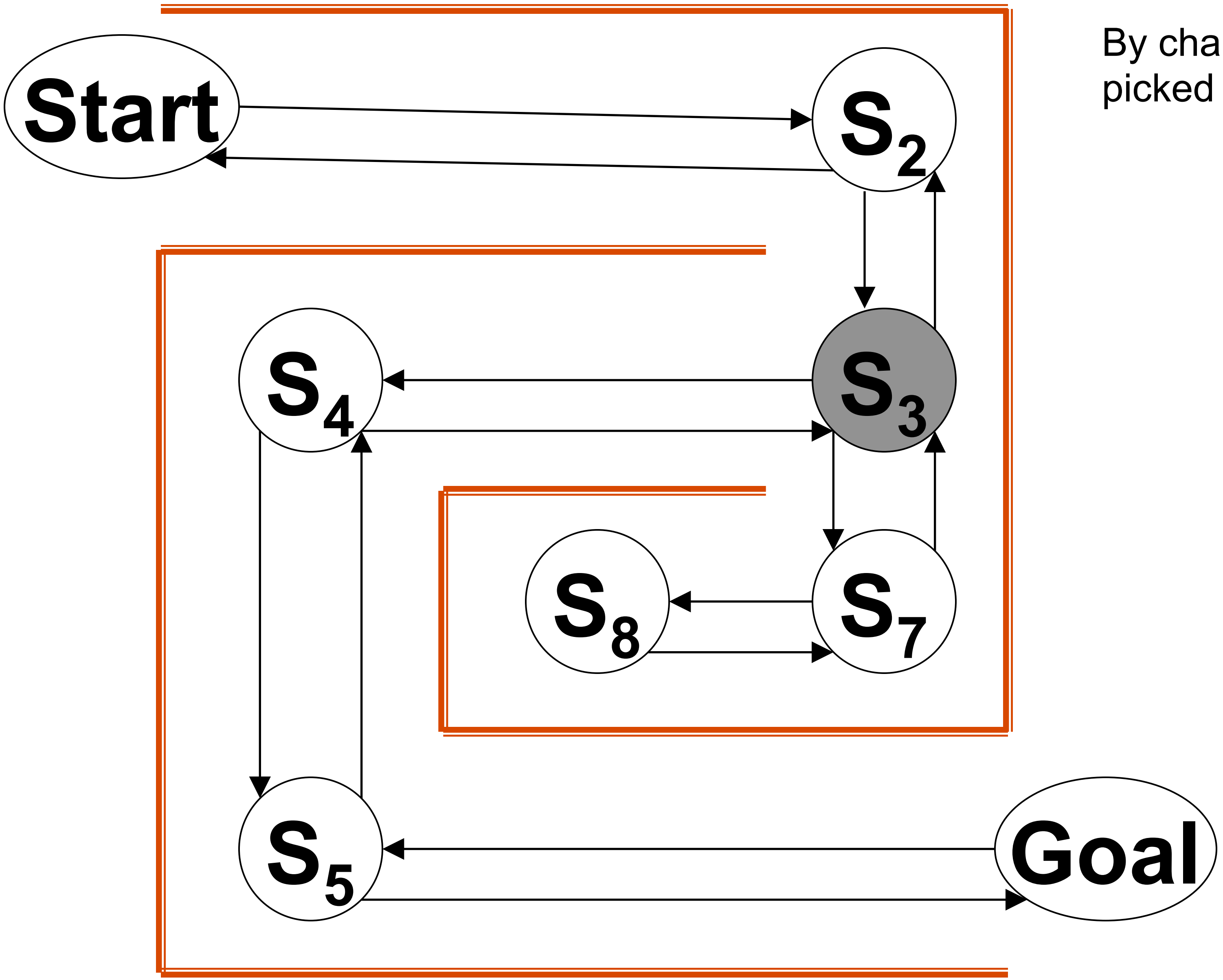


Suppose the randomly sampled response leads to S3 …



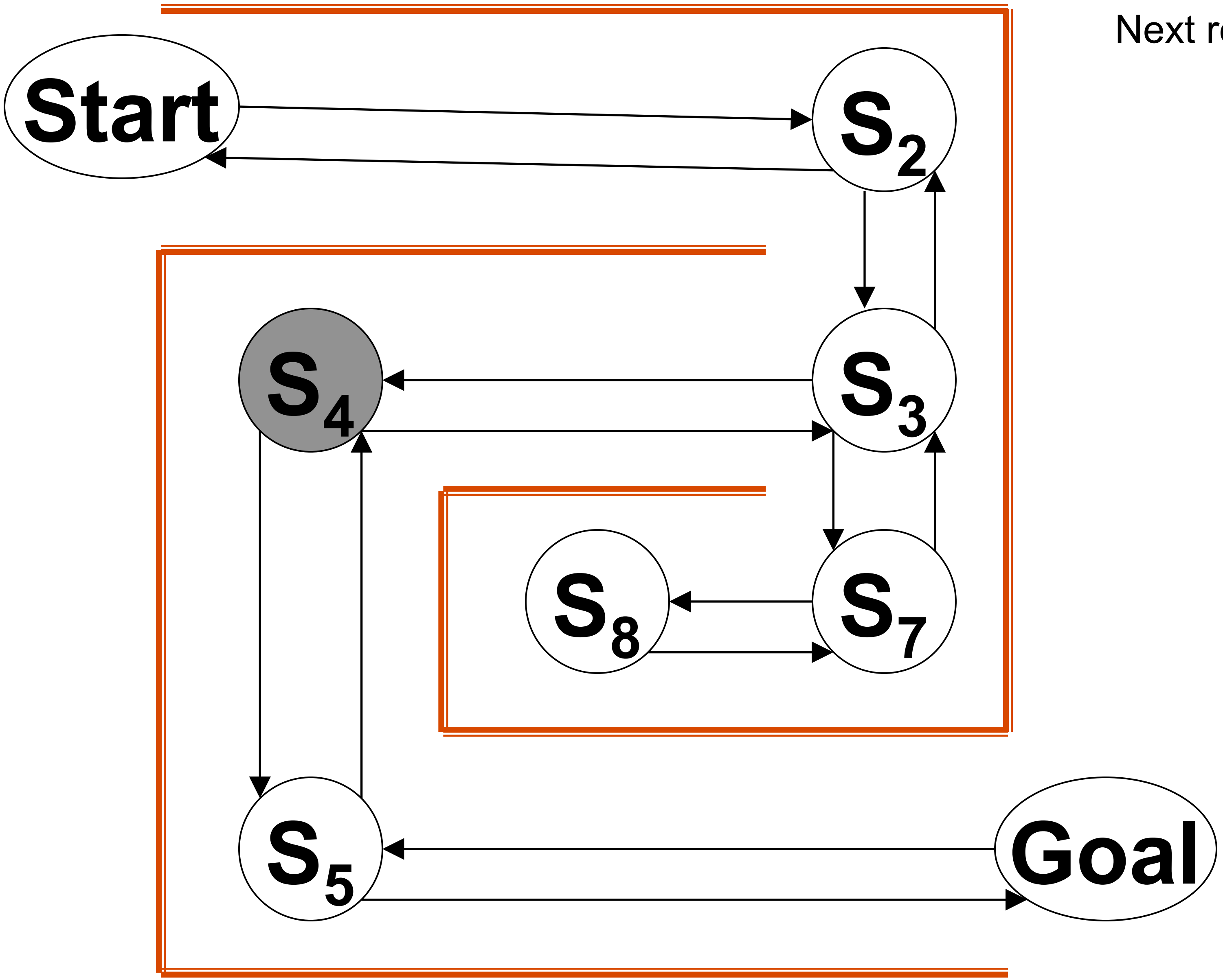
At S3, choices lead to either S2, S4, or S7.

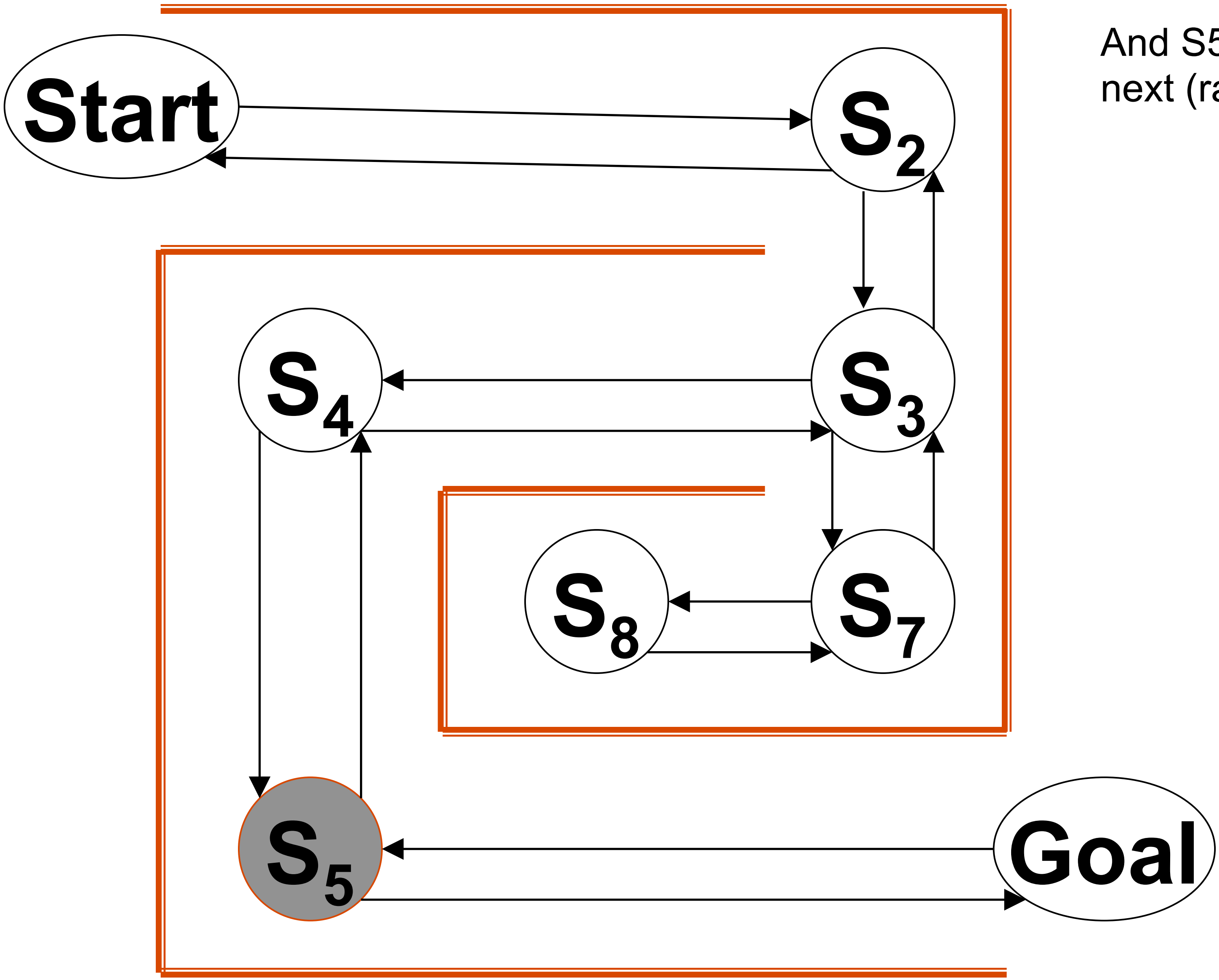
S7 was picked (randomly)



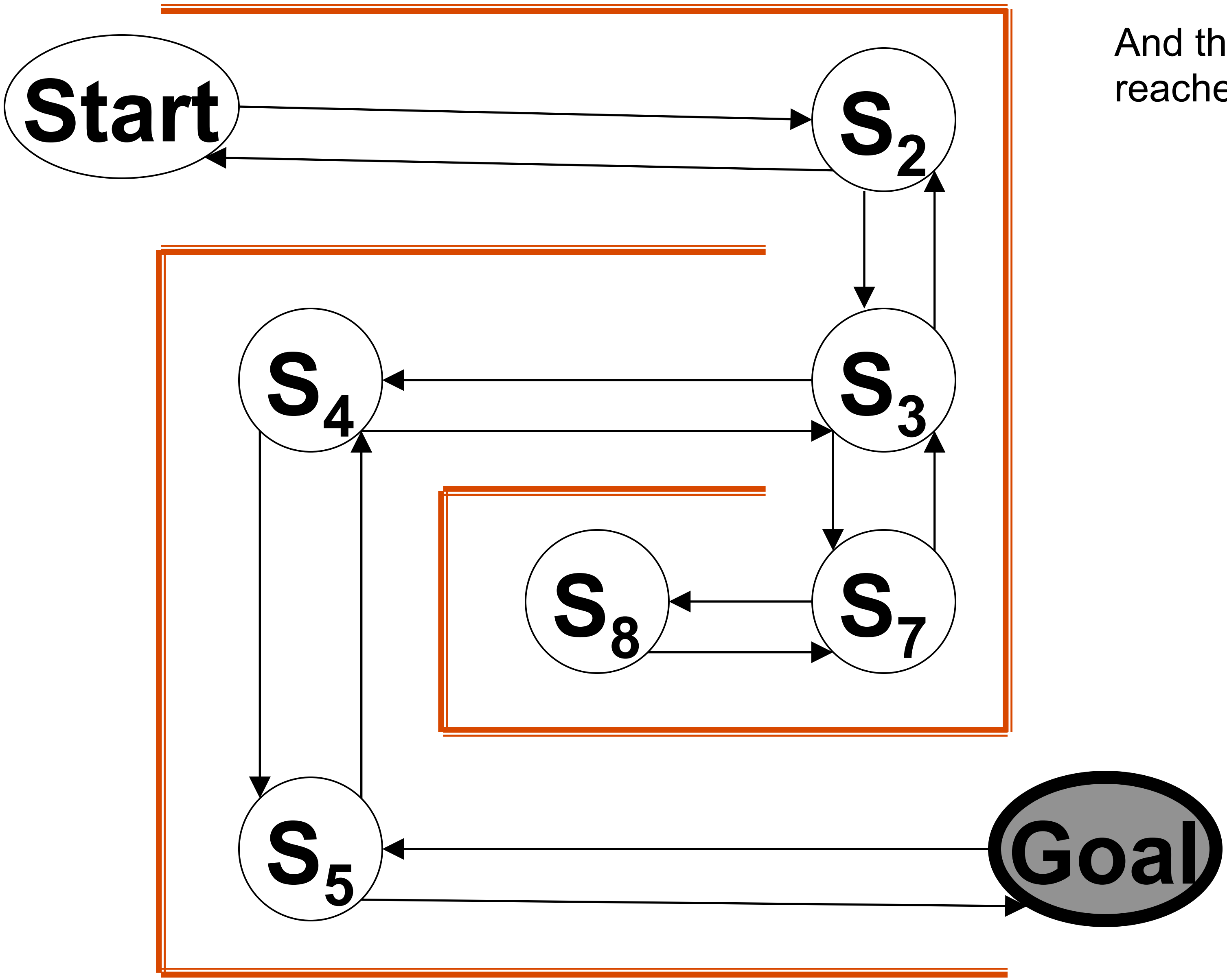
By chance, S3 was picked next…

Next response is S4

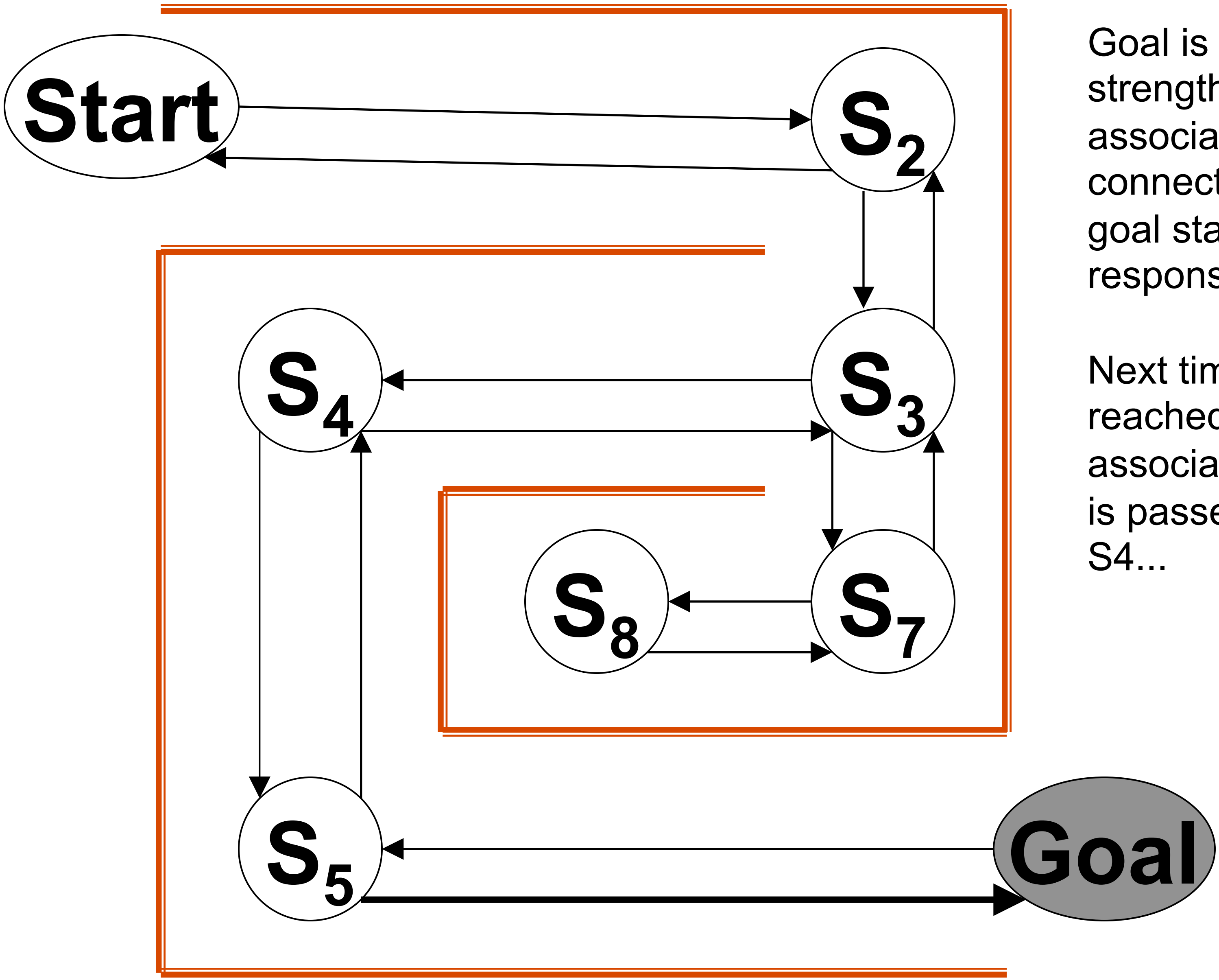




And S5 was chosen next (randomly)



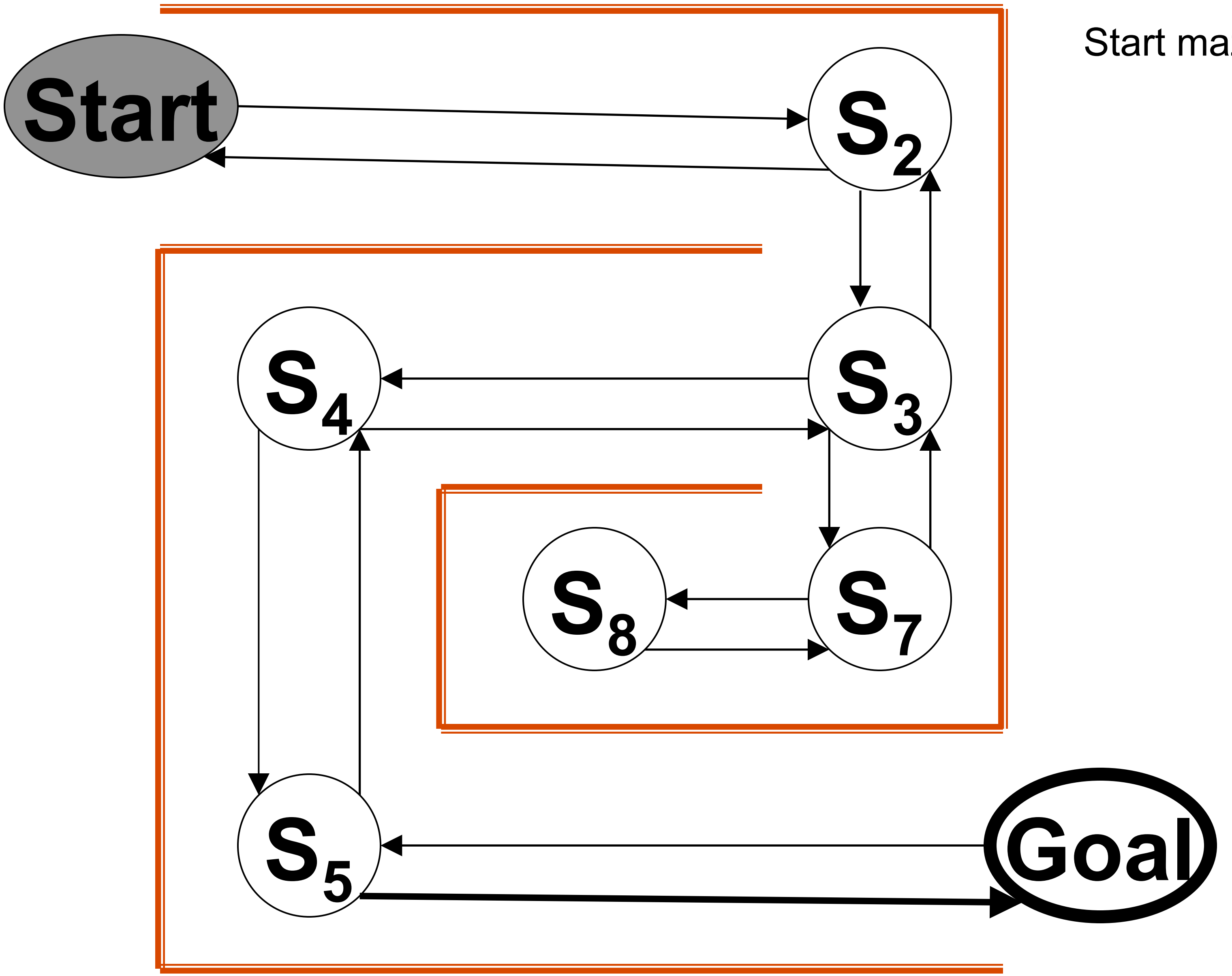
And the goal is reached …

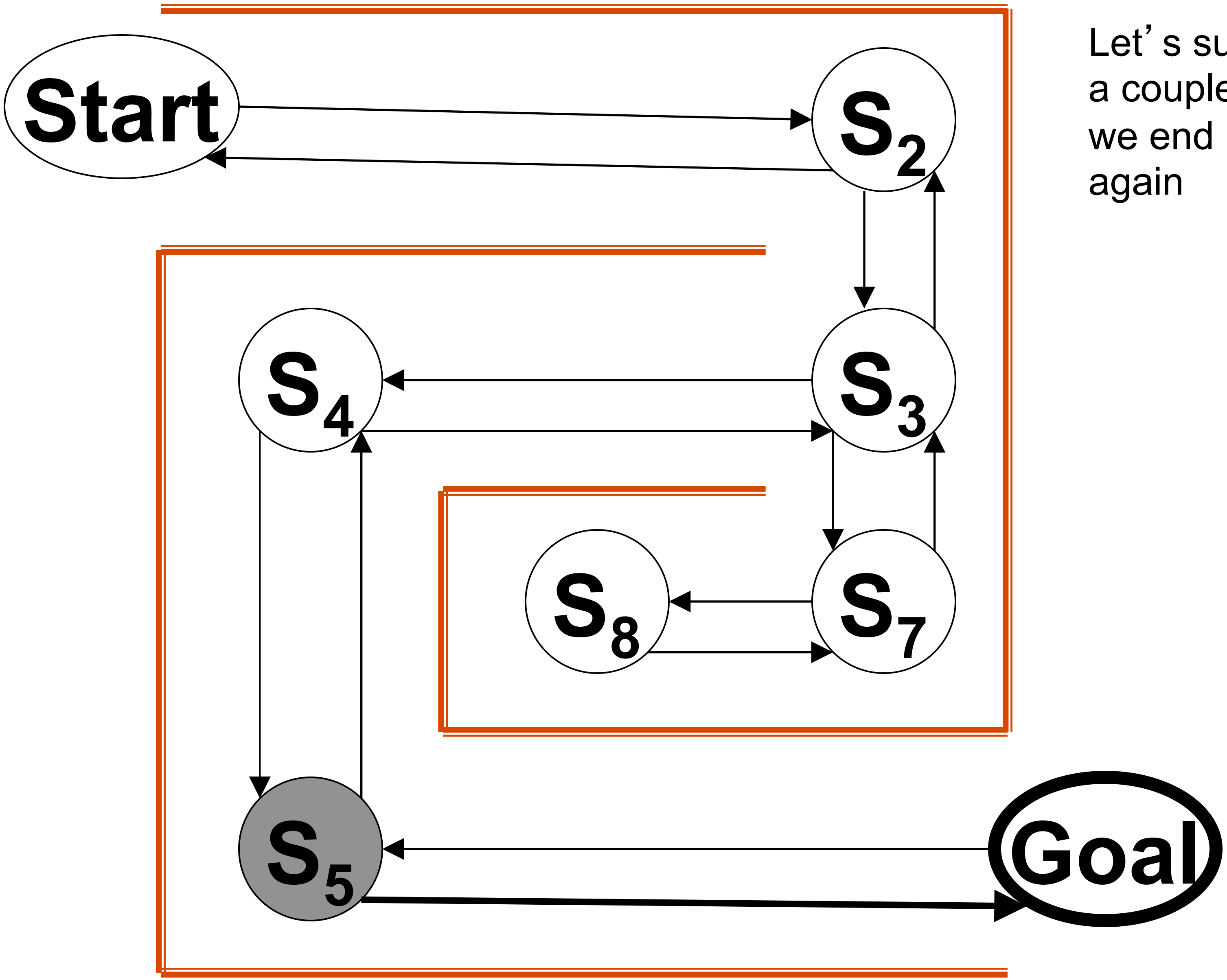


Goal is reached, strengthen the associative connection between goal state and last response

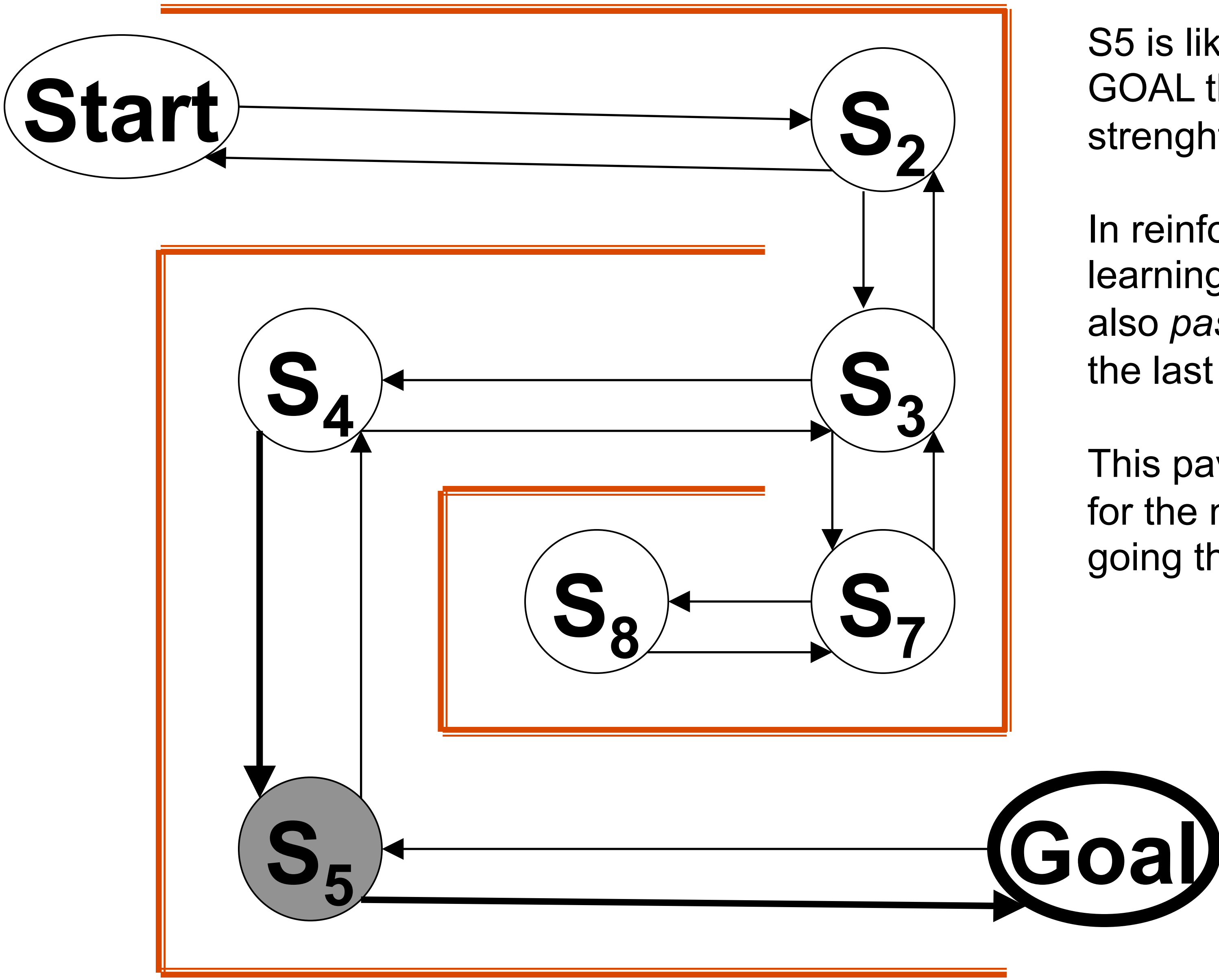
Next time S5 is reached, part of the associative strength is passed back to S4...

Start maze again…





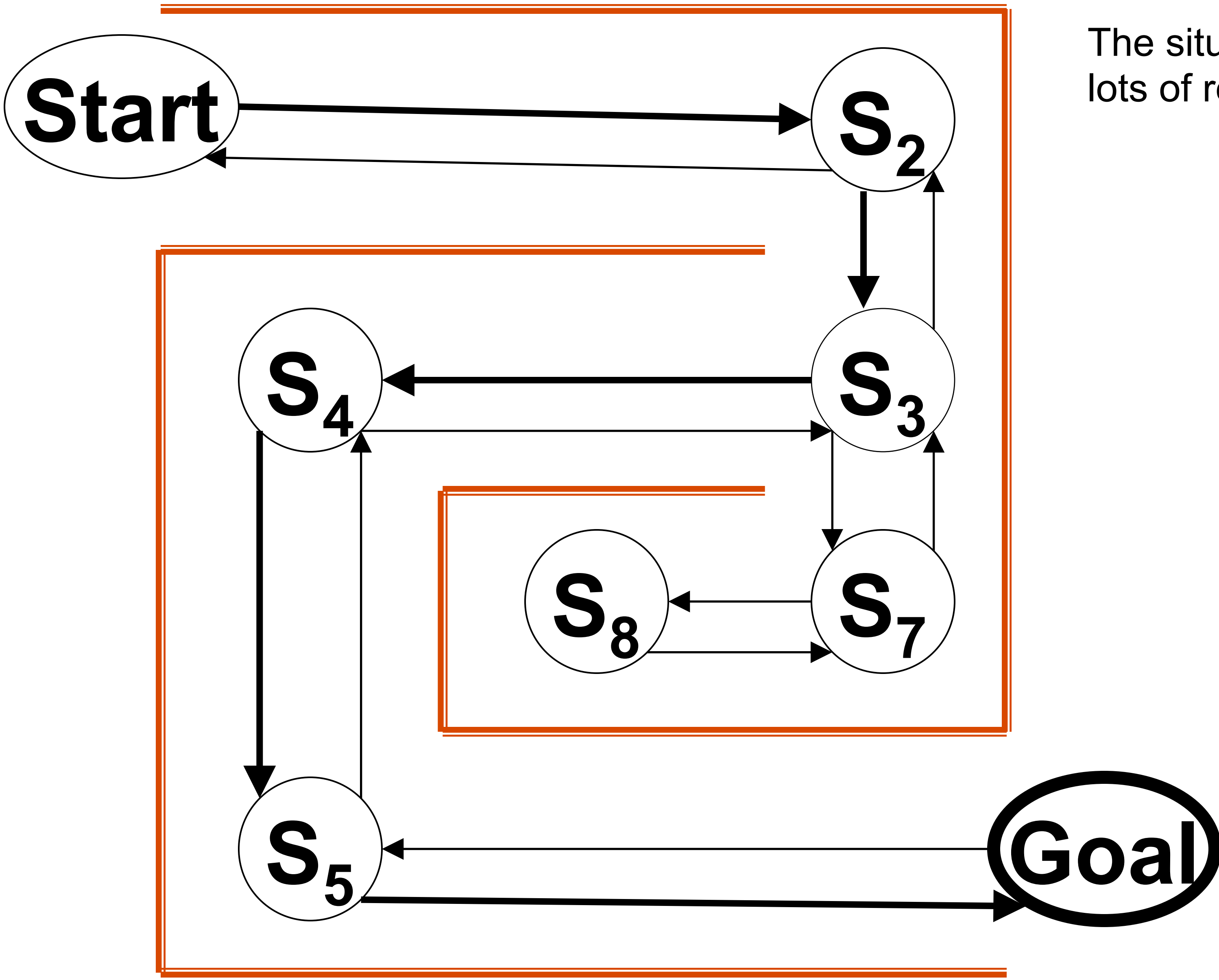
Let’s suppose after a couple of moves, we end up at S5 again



S5 is likely to lead to GOAL through strenghtened route

In reinforcement learning, strength is also *passed back* to the last state

This paves the way for the next time going through maze



The situation after lots of restarts …

Stanford autonomous helicopter

- https://www.youtube.com/watch?v=VCdxqn0fcnE

(a) (b)

Figure 1: (a) Autonomous helicopter. (b) Helicopter hovering under control of learned policy.

RL applications in robotics

- Robot Learns to Flip Pancakes
- Autonomous spider learns to walk forward by reinforcement learning
- Reinforcement learning for a robitic soccer goalkeeper

Conclusion

- The Reinforcement Learning Problem
- Inside an RL agent
  - Policy
  - Reward
  - Value
  - Model
- Temporal difference learning