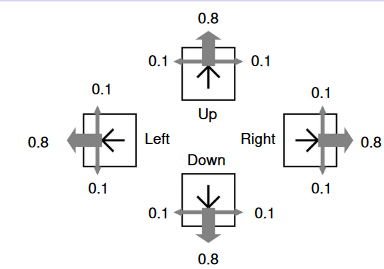
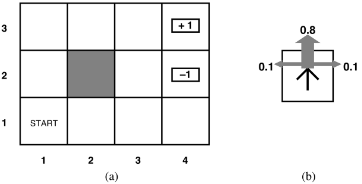
CMP9132M – Markov Decision Process

**Motion Model**



80% of the time the agent moves as intended

20% of the time the agent moves perpendicular to the intended direction. Half the time to the left, half the time to the right.

The agent doesn’t move if it hits a wall.

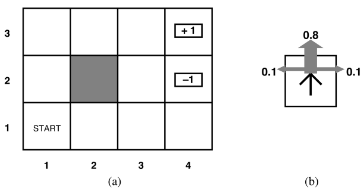
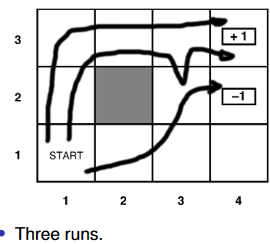
**Transition model**

We can write a transition model to describe these actions. Since the actions are stochastic, the model looks like: 

Where a is the action that takes the agent s to s’.

* Transitions are assumed to be (first order) Markovian.
* They only depend on the current and next states.
* So, we could write a large set of probability tables that would  
  describe all the possible actions executed in all the possible  
  states.  
  This would completely specify the actions.

Runs



* The reward for non-terminal states is ́0.04.
* We will assume that the utility of a run is the sum of the  
  rewards of states, so the ́0.04 is an incentive to take fewer  
  steps to get to the terminal state.  
  (You can also think of it as the cost of an action).

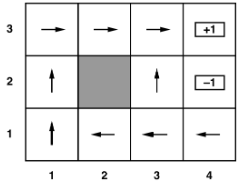
Markov decision process

The overall problem the agent faces here is a Markov decision  
process (MDP).  
Mathematically we have:

* A set of states s Є S with an initial state s0.
* A set of actions A (s) in each state.
* A transition model P(s’|s,a); and
* A reward function R(s).

Captures any fully observable non-deterministic environment  
with a Markovian transition model and additive rewards.

A solution is a policy, which we write as pi

**

This is a choice of actions for every state



That wasy if we get off track, we still knoiw what to do. In any state s, pi(s) identifies what action to take.

Naturally we’d prefer not just any policy but the optimum  
policy.

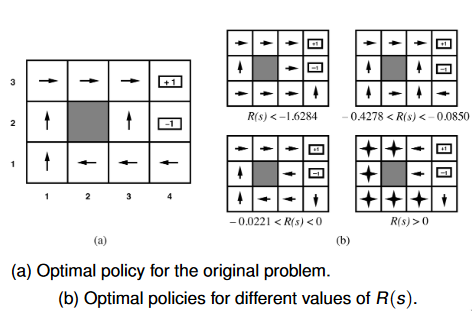
* But how to find it?

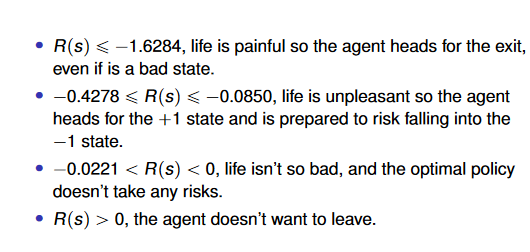
Firstly we compare policies by the utility they generate. As actions are stochastic, policies won’t give the same utility every time.  
Therefore we compare the expected utility.  
The optimum policy π ̊ is the policy with the highest expected  
utility.  
At every stage the agent should do **π \*(s)**

**π \*(s)** is the right thing.

But now is the best action in the best run, not the best myopic single action.

*Markov assumption* – also referred to as the first order assumption. The state previous to the one the robot is currently at.





**How utilities are calculated**