Timing Sheet

- 3: Trees/Utility Lecture
- 3: Tree Alone
- 5: Tree Together
- 5: Tree Over
- 5 Utils (Together)
- 5 Util (Over)

- 5: MDP Lecture
- 5 MDP Alone
- 5 MDP Together
- 9 MDP Go Over

New Thing!

Join section mailing list:

http://sniyaz.weebly.com/cs188.html

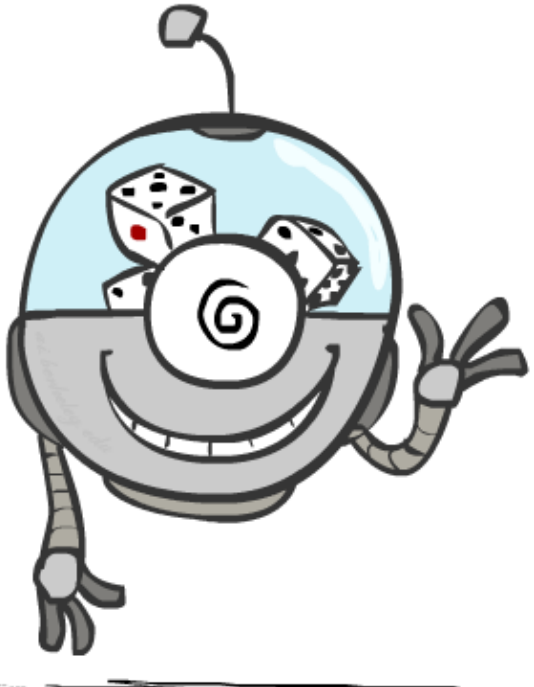
CS 188: Artificial Intelligence

Discussion 3:

Game Trees, Utilities, MDPs

TA: Sherdil Niyaz

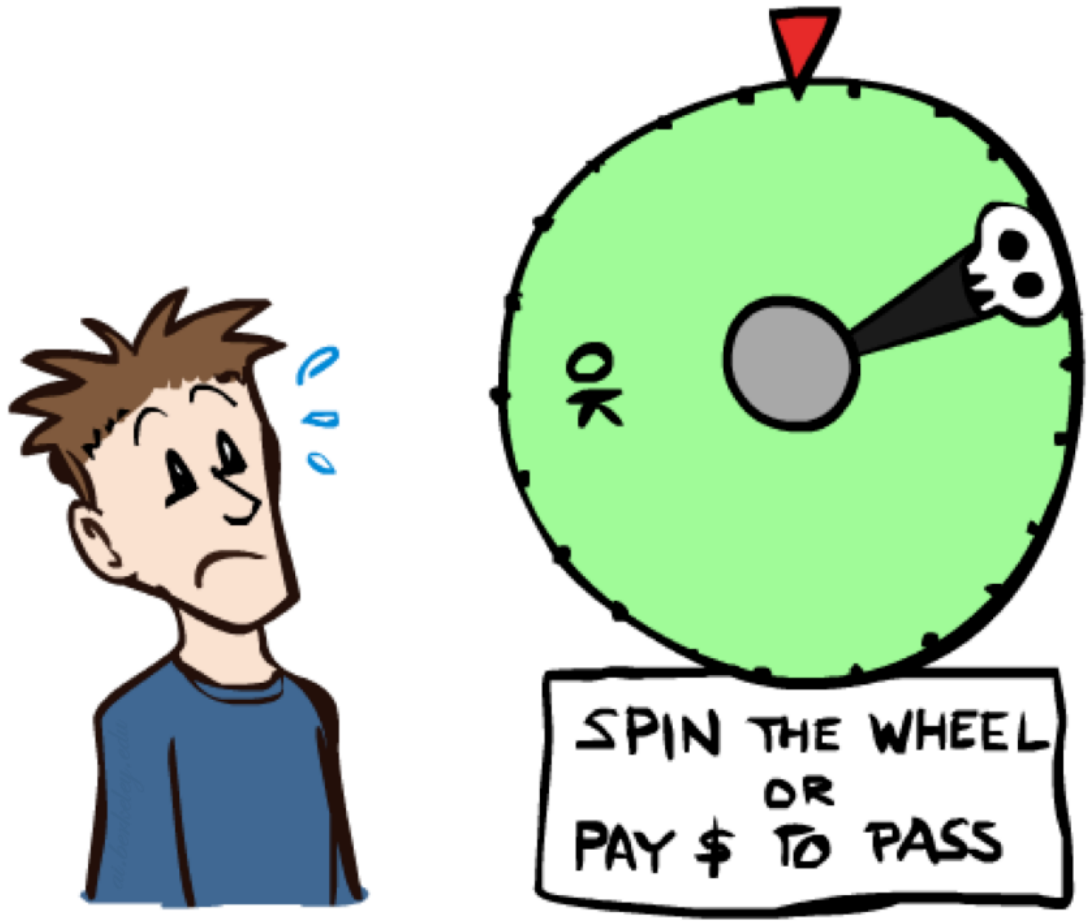
University of California, Berkeley



Announcements

- Homework 4: MDPs (today's topic)
  - Due Monday 9/26 at 11:59pm.
- Contest 1: Search
  - Due tonight at 11:59 PM.
  - Extra credit! See Piazza for Caryn's getting started video.
- Project 2: Multi-Agent Pacman
  - Has been released, due Friday 9/30 at 5:00pm.

Probability and Utilities



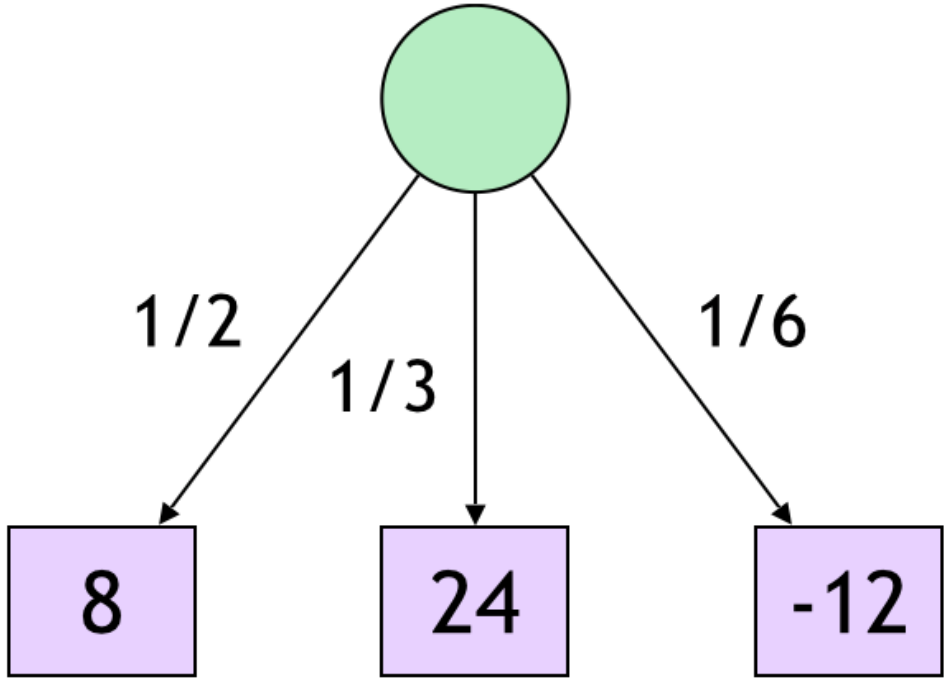
CS70 Review: Expectation

$$E(x) = \sum_{i=1}^{N} x_i p_i$$

- Essentially weight each outcome by its probability!
- Also called the **mean.**

Expecti-Max

- Value of state is expected value of the outcome!



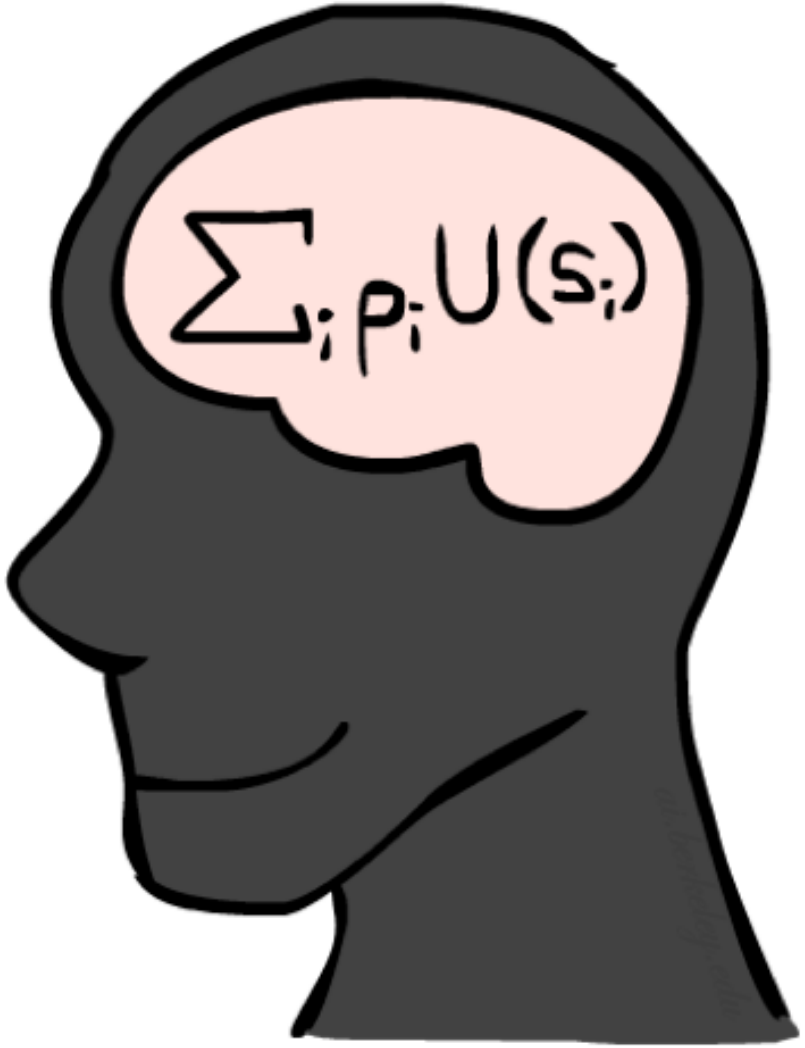
Utility

- Utility function: U(x)
- Tells us how important x is to us.

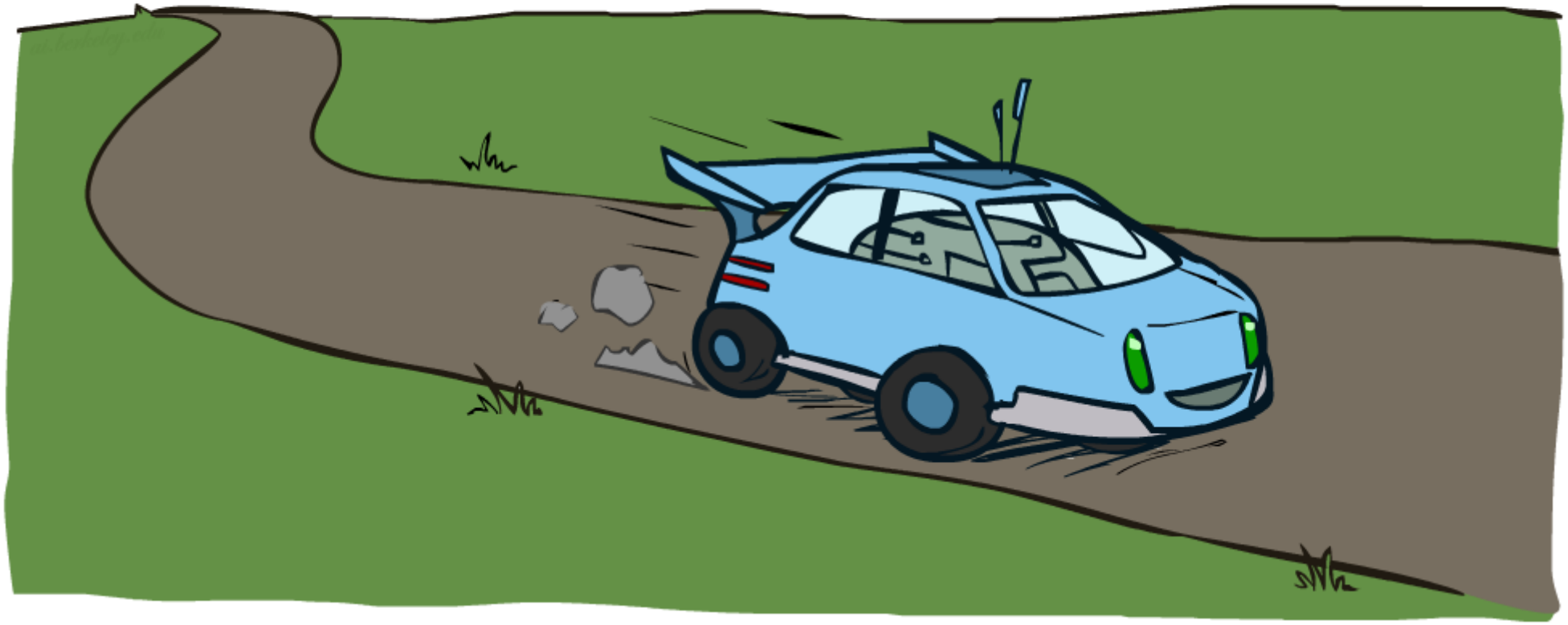
- Lottery: we may receive each outcome with a certain probability:

$$(s_1, p_1), (s_2, p_2), \ldots, (s_n, p_n)$$

- Utility of a lottery: see picture.



MDPs



MDPs

- A **new** type of search problem!

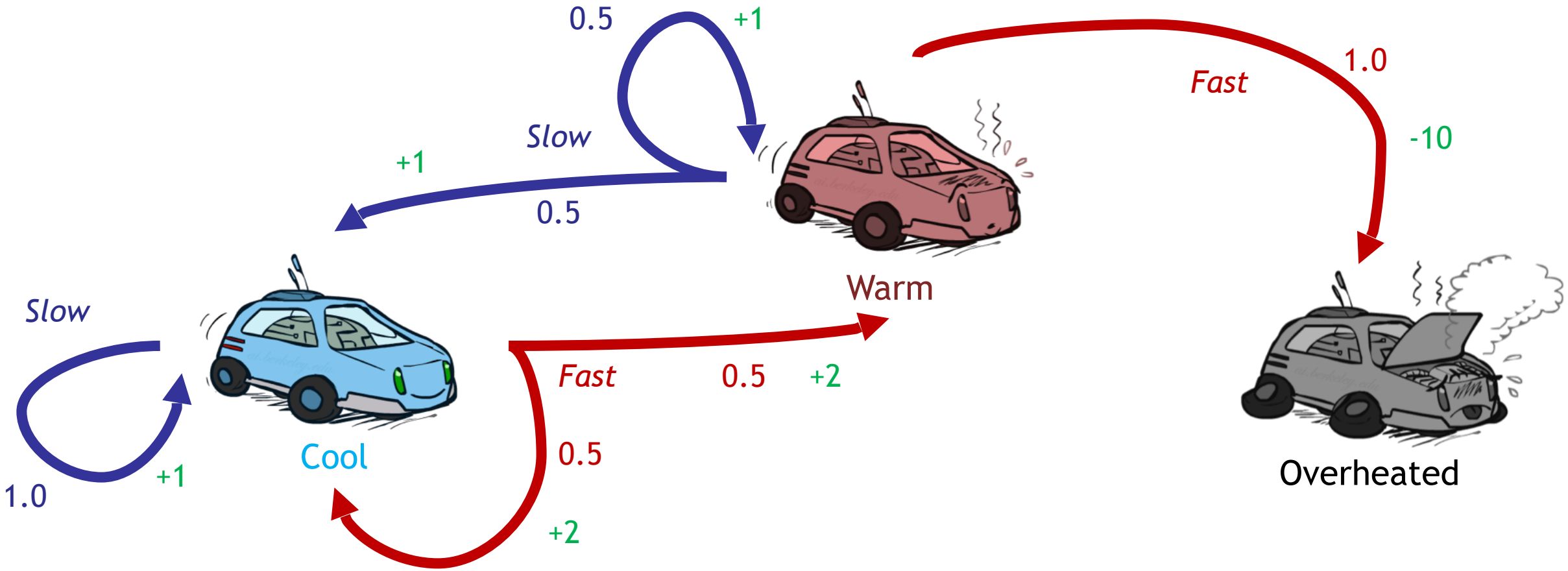
- There is a set of states that we are constantly moving between. We collect **rewards** as we move though different states.
- We still have actions available at each state, but these actions won't always take us the same place!!
- This is highly related to **Markov Chains** if you've seen those before.

Markov?

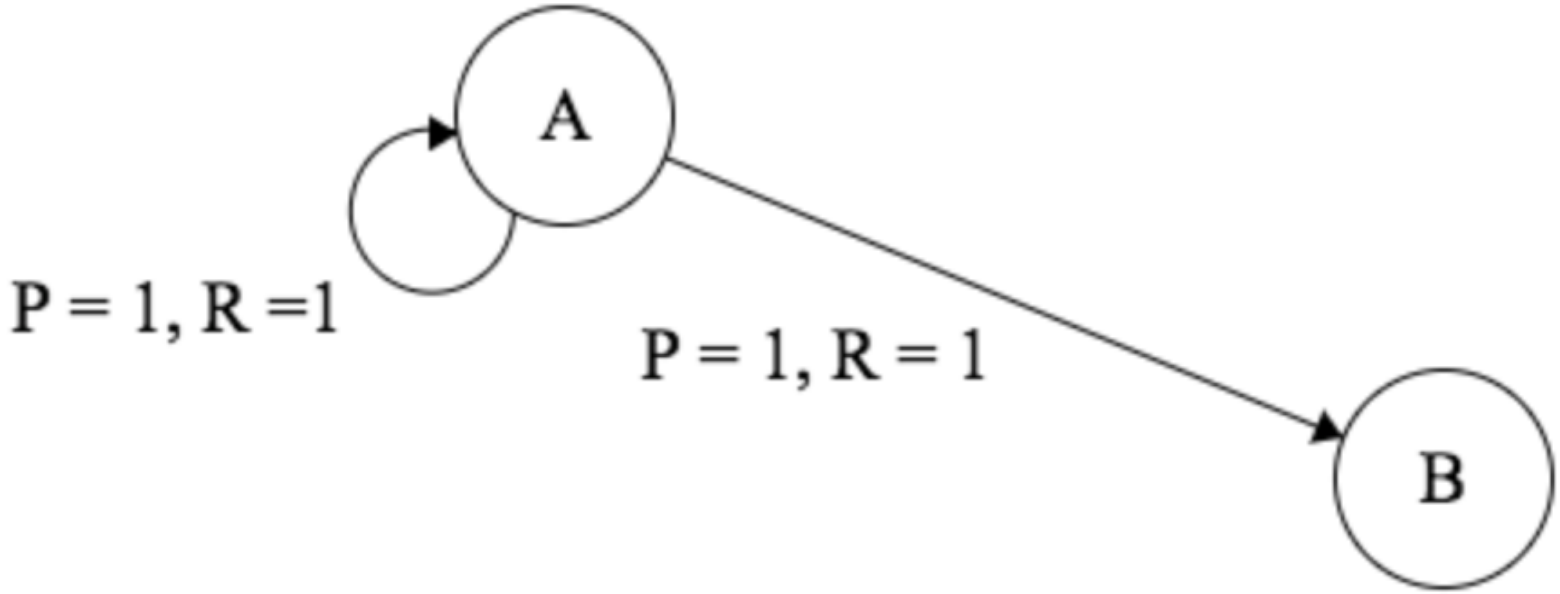
- **Markov:** A type of process where the the next states and probability of reaching them depends **only** on the current state.

- This is sometimes referred to as **memoryless-ness**, since our future options don't depend on the sequence of states we've visited, only our **current position.**

MDPs


0.5
+1
Slow
+1
0.5
Fast
1.0
-10
Warm
Slow
1.0
+1
Cool
Fast
0.5
+2
0.5
+2
Overheated

Infinite Reward?



Discounting

- Fixes the infinite reward problem, makes us prefer earlier rewards.
- At action t (where the first action is t = 0), multiply the reward to collect by γ^t
- Previous example: Would collect $1 + 1\gamma + 1\gamma^2 + ...$
- Assume $\gamma < 1$
- Then, this is finite by geometric sum: $1/(1-\gamma)$

Discounting

- General case:

$$U([r_0, \ldots r_\infty]) = \sum_{t=0}^{\infty} \gamma^t r_t \leq R\text{max}/(1-\gamma)$$

- No infinite rewards if $\gamma < 1$