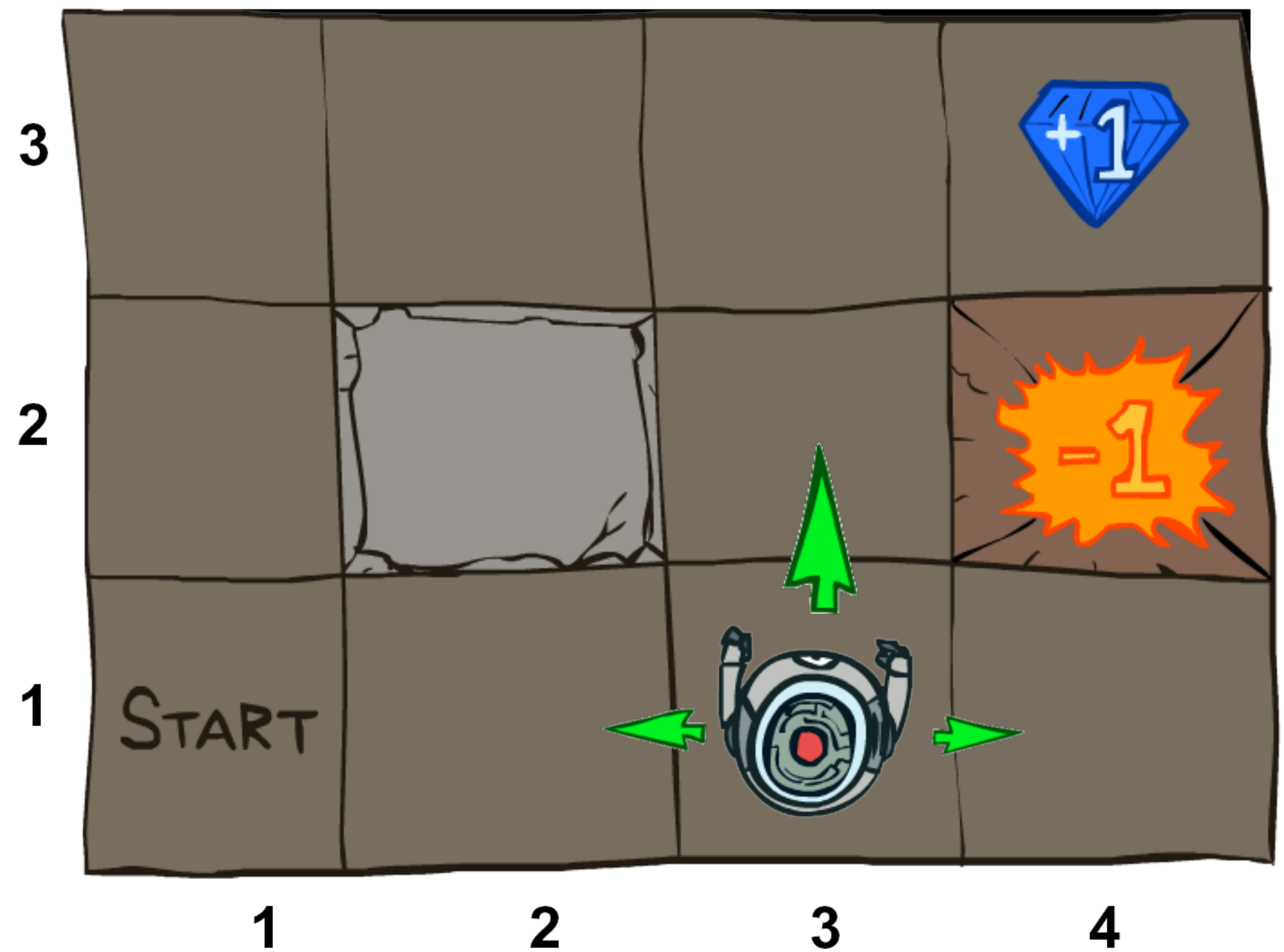
Complex sequential decisions

Tomas Svoboda, BE5B33KUI
2017-03-27, 2017-04-03

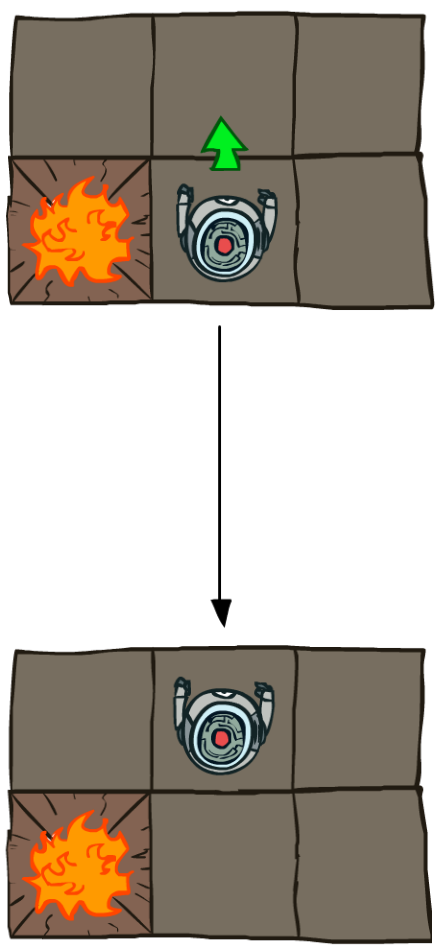
Slide material partly from CS 188: Artificial Intelligence at UCB
by Dan Klein, and Pieter Abbeel, used with permision

Almost like search, …

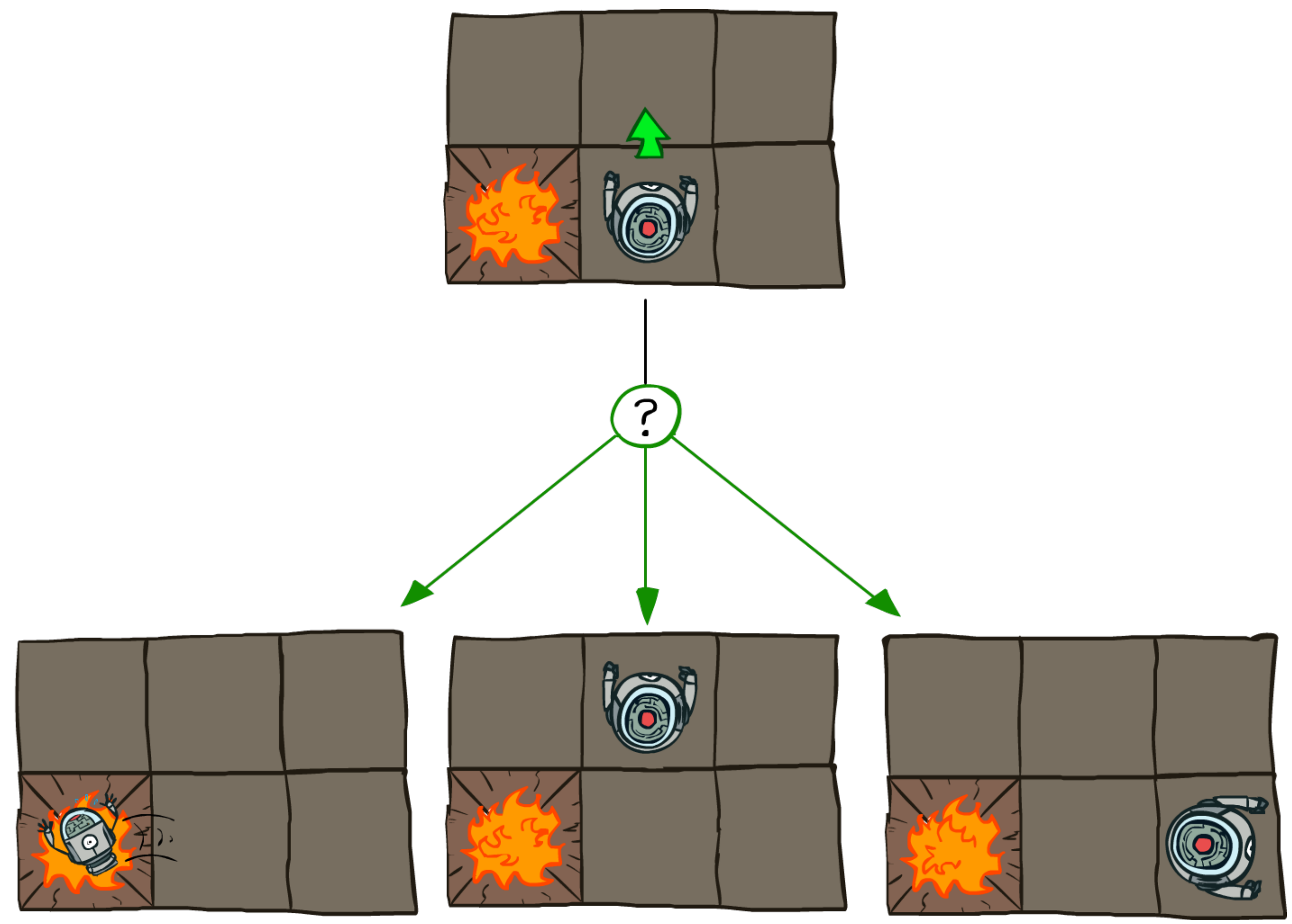


Stochastic actions

Deterministic Grid World

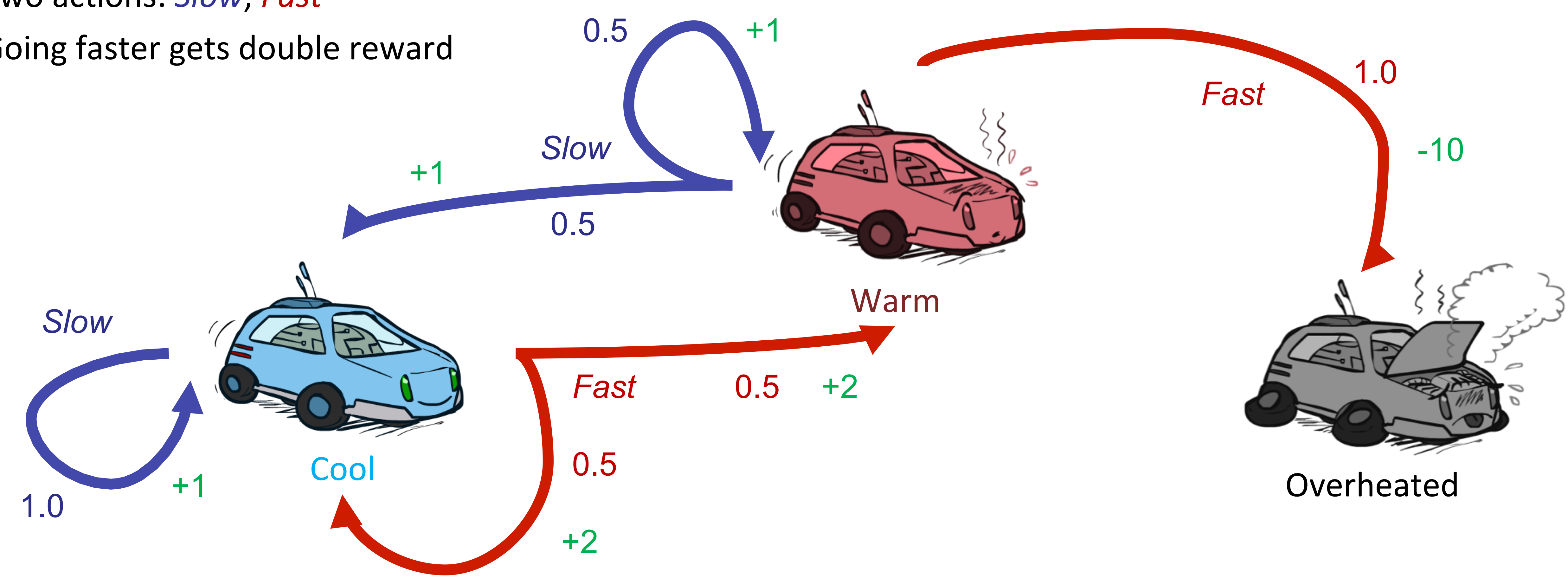


Stochastic Grid World

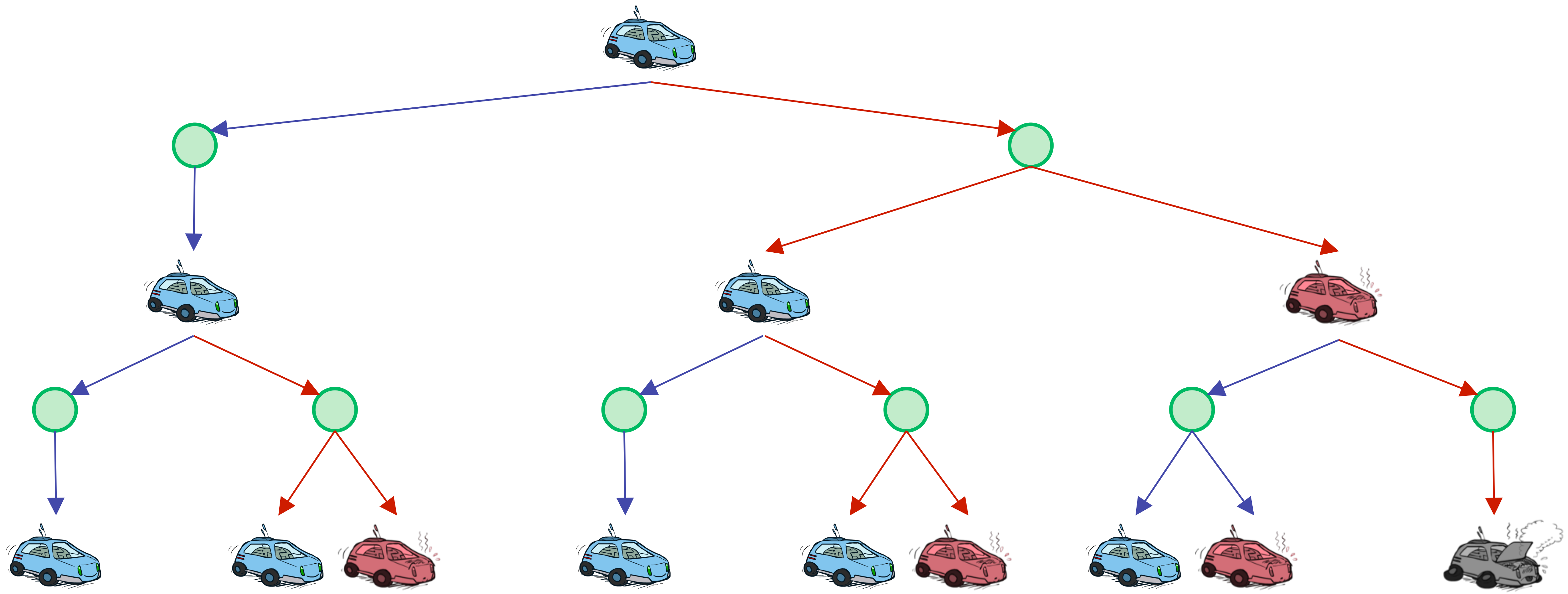


car racing

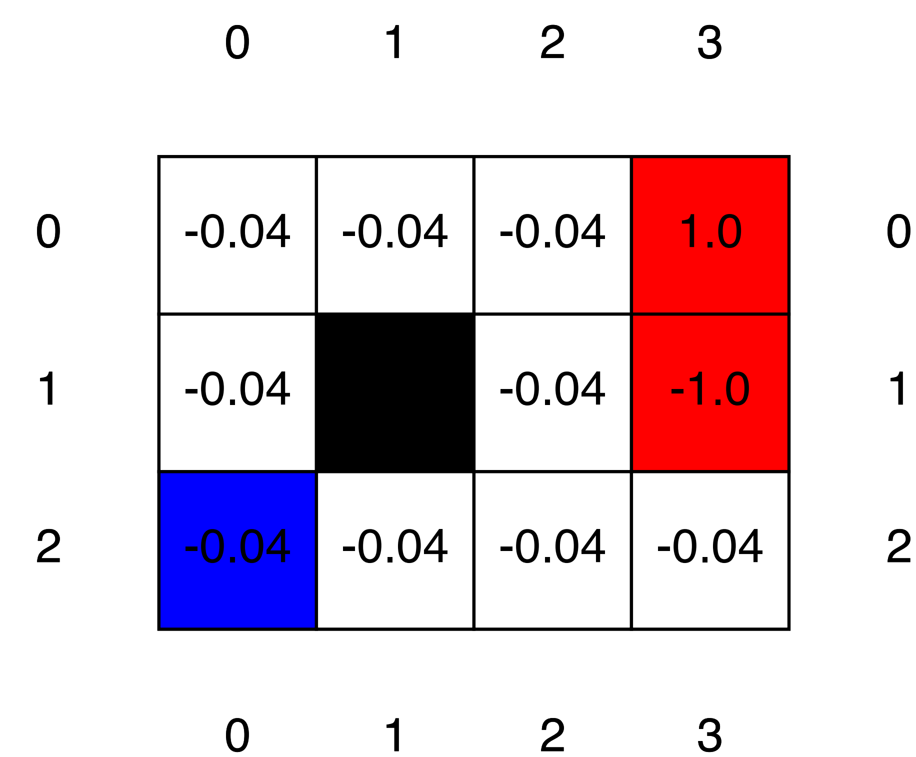
- A robot car wants to travel far, quickly
- Three states: Cool, Warm, Overheated
- Two actions: *Slow*, *Fast*
- Going faster gets double reward



Racing search tree



Grid world MDP



States $s \in S$, actions $a \in A$

Model $T(s, a, s') \equiv P(s'|s, a) =$ probability that a in s leads to s'

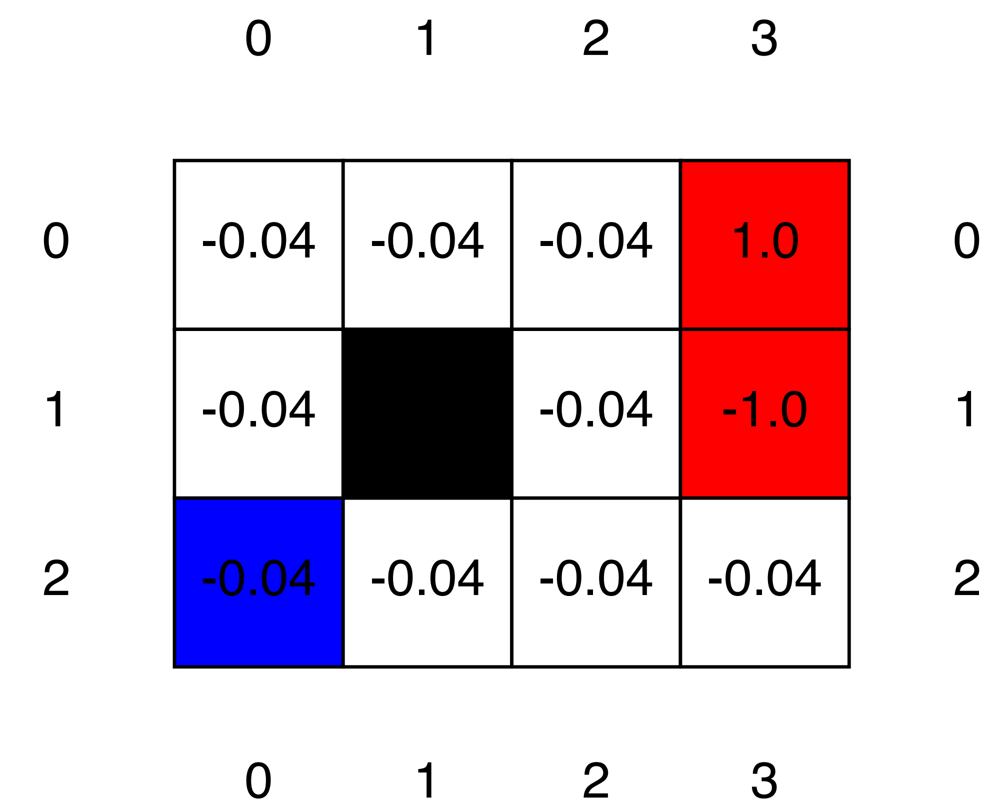
Reward function $R(s)$ (or $R(s, a)$, $R(s, a, s')$)

$$= \begin{cases} -0.04 & \text{(small penalty) for nonterminal states} \\ \pm 1 & \text{for terminal states} \end{cases}$$

Utility of a sequence

State reward $R(s)$

State sequence $[s_0, s_1, s_2, \ldots]$



Utility (h - history)

$$U_h([s_0, s_1, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \cdots$$

Discounted utility (h - history)

$$U_h([s_0, s_1, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots$$

Discounted rewards

Discounted utility (h - history)
$U_h([s_0, s_1, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots$

Agent stationary preference

$$[s_0, s_1, s_2, \ldots] \succ [s_0, s'_1, s'_2, \ldots]$$

$$[s_1, s_2, \ldots] \succ [s'_1, s'_2, \ldots]$$

How to find a policy

maximize *Expected* utility

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t)\right]$$

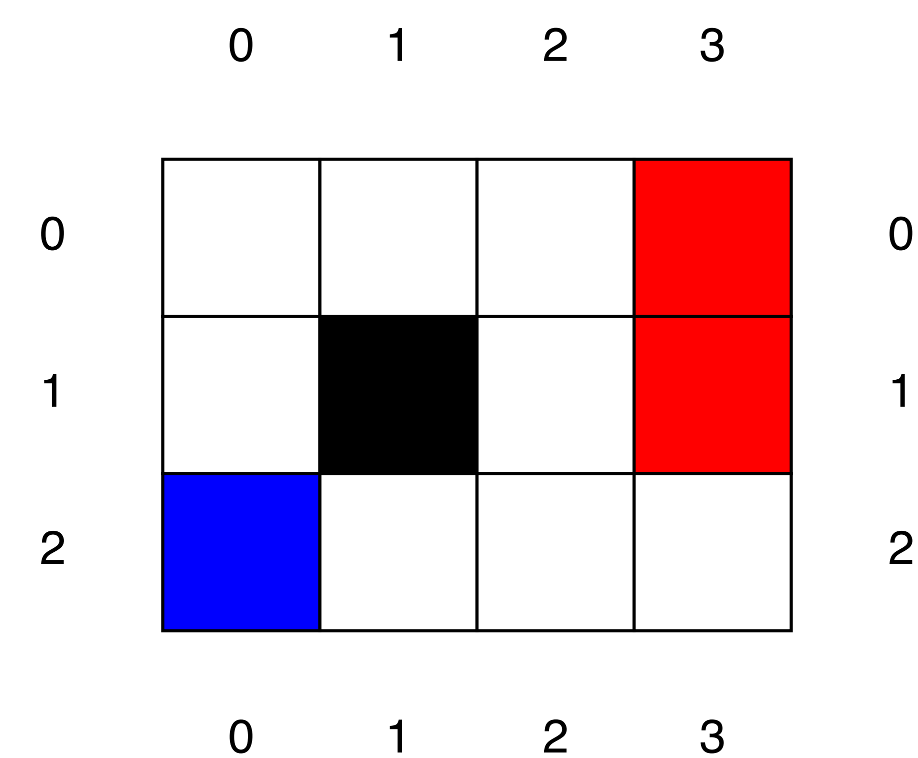
Optimal policy

maximize expected utility of the subsequent state

$$\pi^*(s) = \underset{a \in A(s)}{\text{argmax}} \sum_{s'} P(s'|s,a)U(s')$$

Bellman equation for utilities

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a) U(s')$$



Value iteration algorithm

$$U_{i+1}(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a) U_i(s')$$

	0	1	2	3	
0	-0.04	-0.04	-0.04	1.0	0
1	-0.04		-0.04	-1.0	1
2	-0.04	-0.04	-0.04	-0.04	2
	0	1	2	3	

	0	1	2	3	
0	0.81	0.87	0.92	1.0	0
1	0.76		0.66	-1.0	1
2	0.71	0.66	0.61	0.39	2
	0	1	2	3	