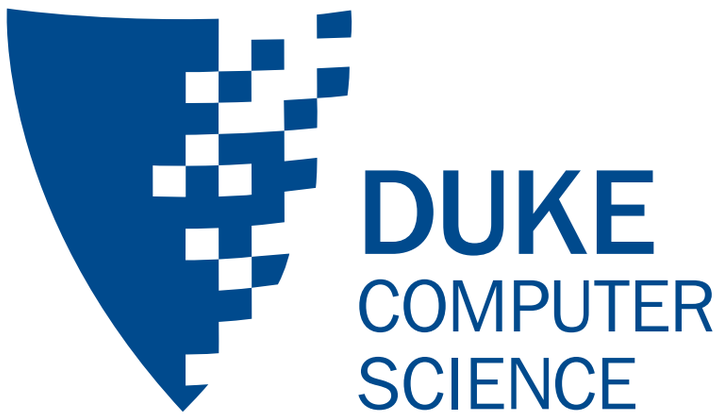
Decision Making for Robots and Autonomous Systems

Fall 2015



George Konidaris
gdk@cs.duke.edu

POMDPs

More formally, a POMDP is:

S, a set of states

A, a set of actions

T, transition function

R, reward function

γ , discount factor

Ω, set of observations

O, observation function $O(\omega_t|s_t)$

The Belief MDP

A belief MDP consists of a tuple (B, A, τ, r, γ):

B is the set of belief states.

A is the action set

τ is the belief state transition function

r is the belief reward function

γ is the discount factor

A and γ are taken directly from the originating POMDP, and the definition of *B* follows from it.

Define τ and *r*, which using stuff from the original POMDP.

One Approach

One approach to solving the Belief MDP:

- Treat it as an MDP
- Use function approximation
- Generate a value function
- Do policy iteration

Does the belief MDP result in any special structure in the resulting VF? *Yes!*

Finite-Horizon POMDPs

DUKE COMPUTER SCIENCE

We restrict ourselves to the finite horizon case.
Let us consider H=1.

Here, we want to take the single action that maximizes immediate reward:

$$\max_a \mathbb{E}_b \left[R(s, a) \right]$$

Therefore the value function looks like:

$$Q(b, a) = \sum_s b(s) R(s, a)$$

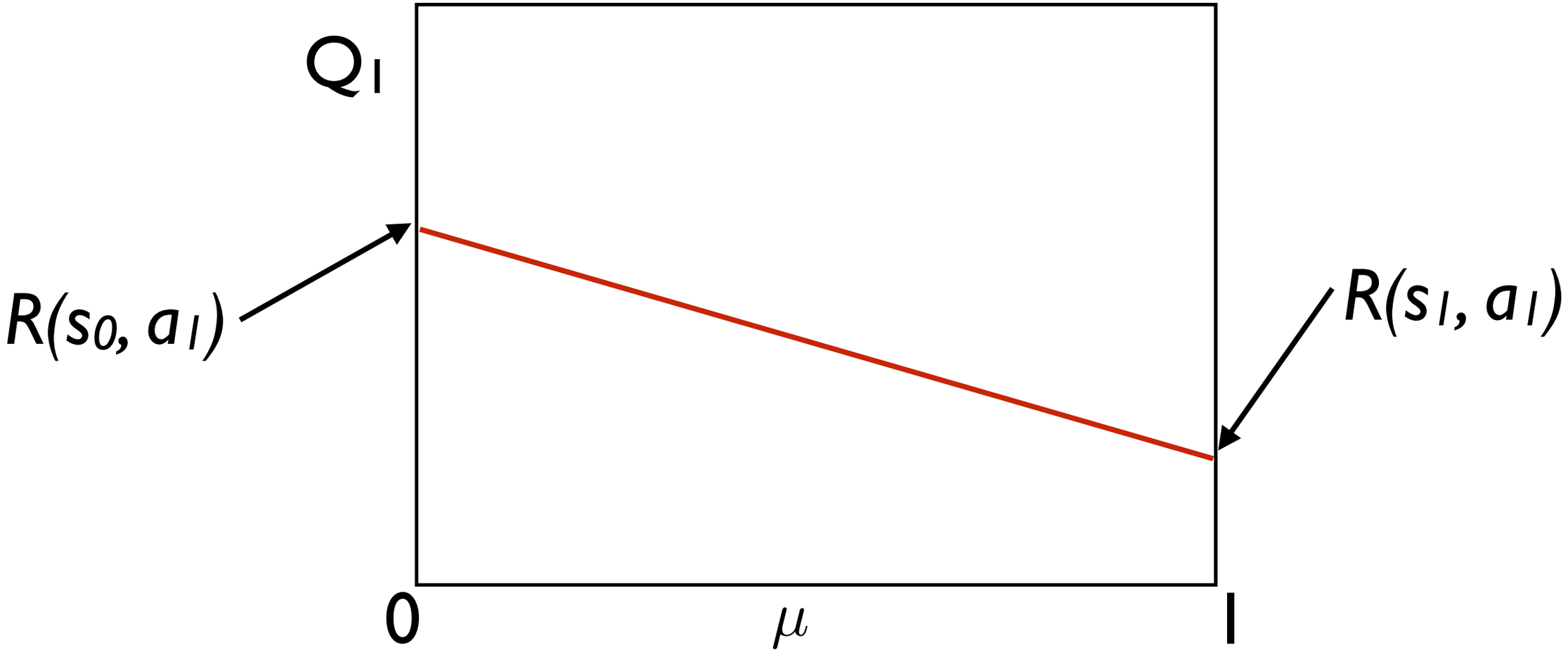
presentation heavily based on http://cs.brown.edu/research/ai/pomdp/tutorial/

Visualizing This

Assume we have only two states s_1 and s_2.

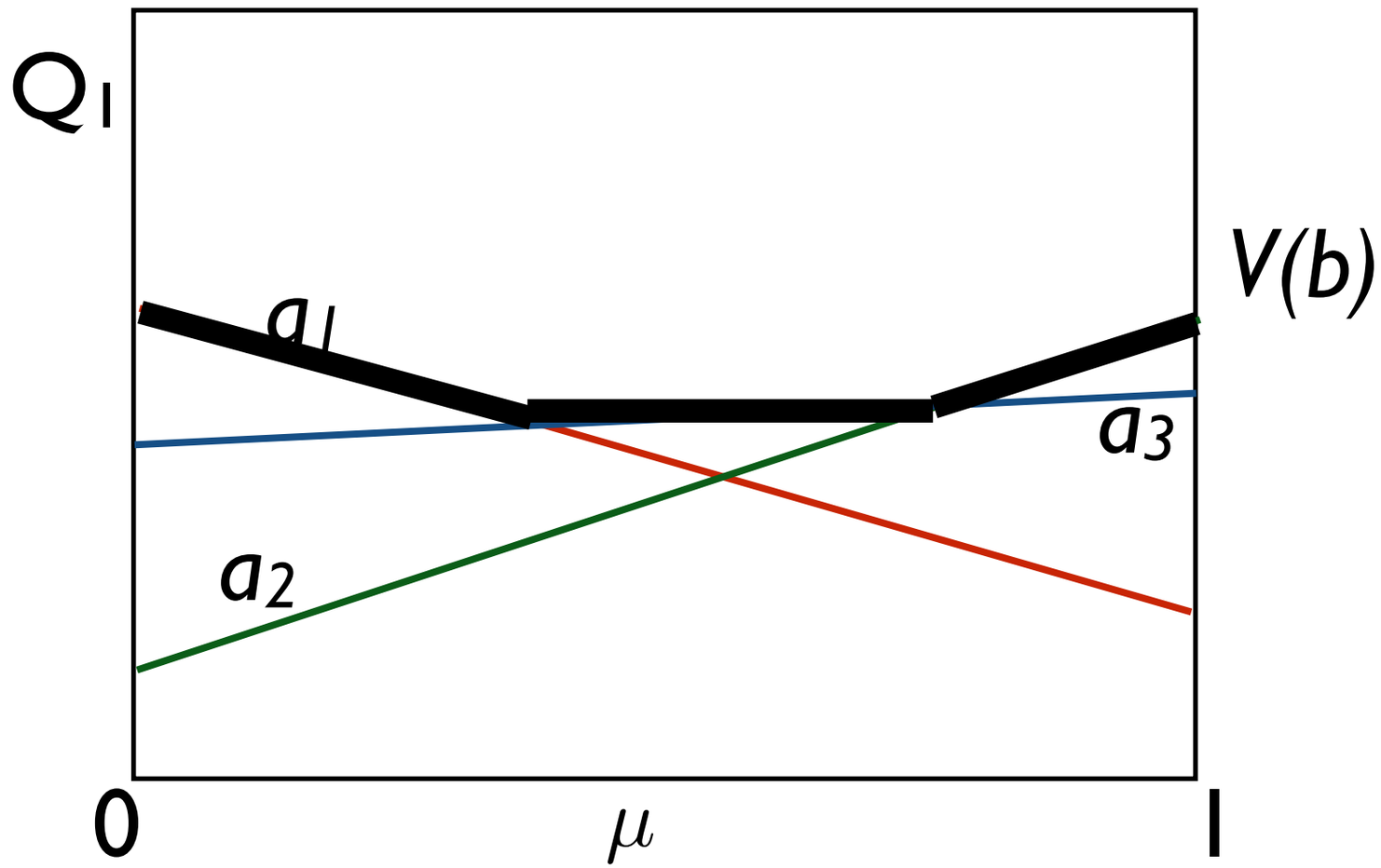
Now $b(s)$ is described by μ, the probability that $s = s_1$.

$Q(b, a_1)$ looks like:



Piecewise Linear

This is true for each action:



So

The value function for H=1 is piecewise linear, and convex.

- There are as many lines as actions.
- There are as many dimensions as possible states, -1
- Lines become hyperplanes in higher dimensions

What about H=2?

Well:

$$Q_2(b,a) = \int_{b'} \tau(b'|b,a)\,[r(b,a,b') + V_1(b')]$$

belief transition

belief reward

convex, piecewise linear

Remember that τ depends on possible observations, each leading to **one** *b'*: finite possible *b'* means integral is a sum.

Convex sum of convex, piecewise linear is convex, piecewise linear. **Holds for all *H*: V is convex, piecewise linear.**

Practical Algorithms

POMDPs are hard:

- Must sample *b* to do backup.
- b is high-dimensional.
- How many lines do we need?

Recent algorithm: SARSOP [Kurniawati et al., RSS 2008]

Represent *V* at set of sample points.

Generate points based on reachable beliefs.

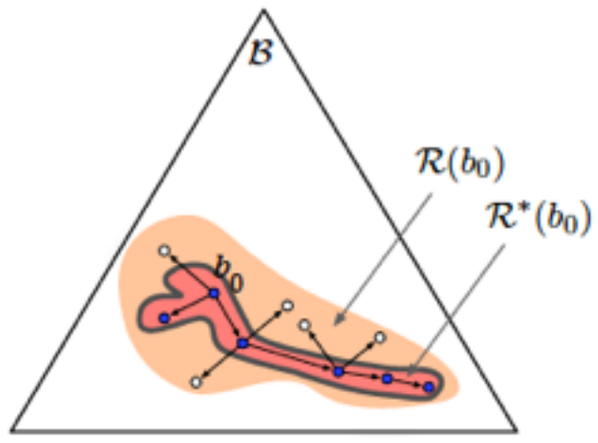
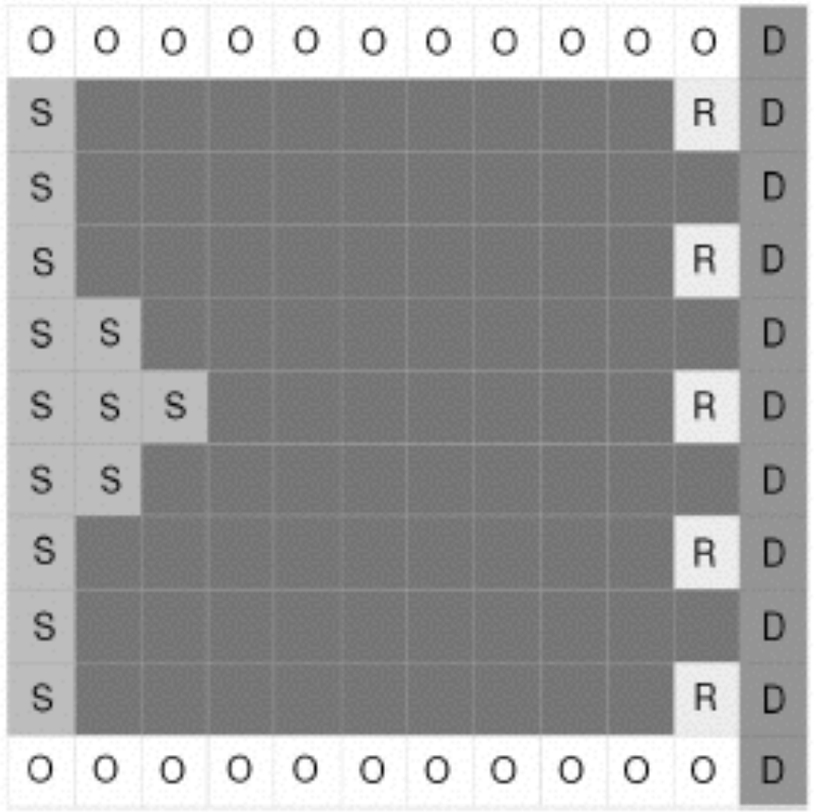


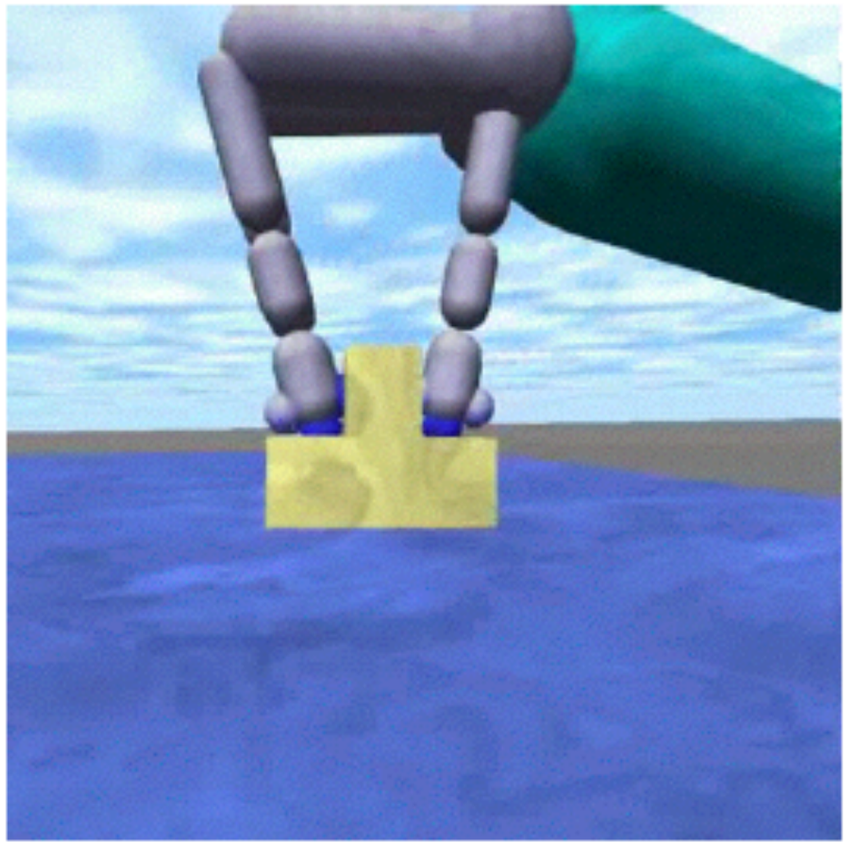
Fig. 1. Belief space $\mathcal{B}$, reachable space $\mathcal{R}(b_0)$, and optimally reachable space $\mathcal{R}^*(b_0)$. Note that $\mathcal{R}^*(b_0) \subseteq \mathcal{R}(b_0) \subseteq \mathcal{B}$.

SARSOP



(*a*) Underwater Navigation, an instance of coastal navigation, shown on a reduced map with a 11×12 grid. "S" marks the possible initial positions for the robot. The robot is equally likely to start in any of these positions. "D" marks the destinations. "R" marks the rocks. "O" marks places that the robot can fully localize itself.

SARSOP



(*b*) Grasping. A fingered robot arm grasps a stepped block. Courtesy of L.P. Kaelbling and T. Lozano-Pérez.

SARSOP

TABLE I

PERFORMANCE COMPARISON.

	Reward	Time (s)
Underwater Navigation,		
$\|\mathcal{S}\|$=2,653,$\|\mathcal{A}\|$=6,$\|\mathcal{O}\|$=103		
SARSOP	722.59 ± 1.30	72
HSVI2	721.45 ± 0.75	720
Grasping		
$\|\mathcal{S}\|$=1,253,$\|\mathcal{A}\|$=6,$\|\mathcal{O}\|$=96		
SARSOP	320.00 ± 0.16	8
HSVI2	319.88 ± 0.14	60
Integrated Exploration		
$\|\mathcal{S}\|$=15,517,$\|\mathcal{A}\|$=8,$\|\mathcal{O}\|$=1,015		
SARSOP	$(1.58 \pm 0.03) \times 10^6$	5,400
HSVI2	$(1.41 \pm 0.02) \times 10^6$	5,400
	$(1.43 \pm 0.02) \times 10^6$	7,200
Rock Sample (7,8)		
$\|\mathcal{S}\|$=12,545,$\|\mathcal{A}\|$=13,$\|\mathcal{O}\|$=2		
SARSOP	21.27 ± 0.13	400
HSVI2	21.27 ± 0.09	250
Tag		
$\|\mathcal{S}\|$=870,$\|\mathcal{A}\|$=5,$\|\mathcal{O}\|$=30		
SARSOP	-6.13 ± 0.12	6
HSVI2	-7.43 ± 0.11	6
	-6.40 ± 0.10	7,200
Homecare		
$\|\mathcal{S}\|$=5,408,$\|\mathcal{A}\|$=9,$\|\mathcal{O}\|$=928		
SARSOP	16.86 ± 0.45	960
HSVI2	16.88 ± 0.37	2,880

source code available!

Models (Reprise)

DUKE COMPUTER SCIENCE

Even though robots are *the one true way*, most of this course was about **models**.

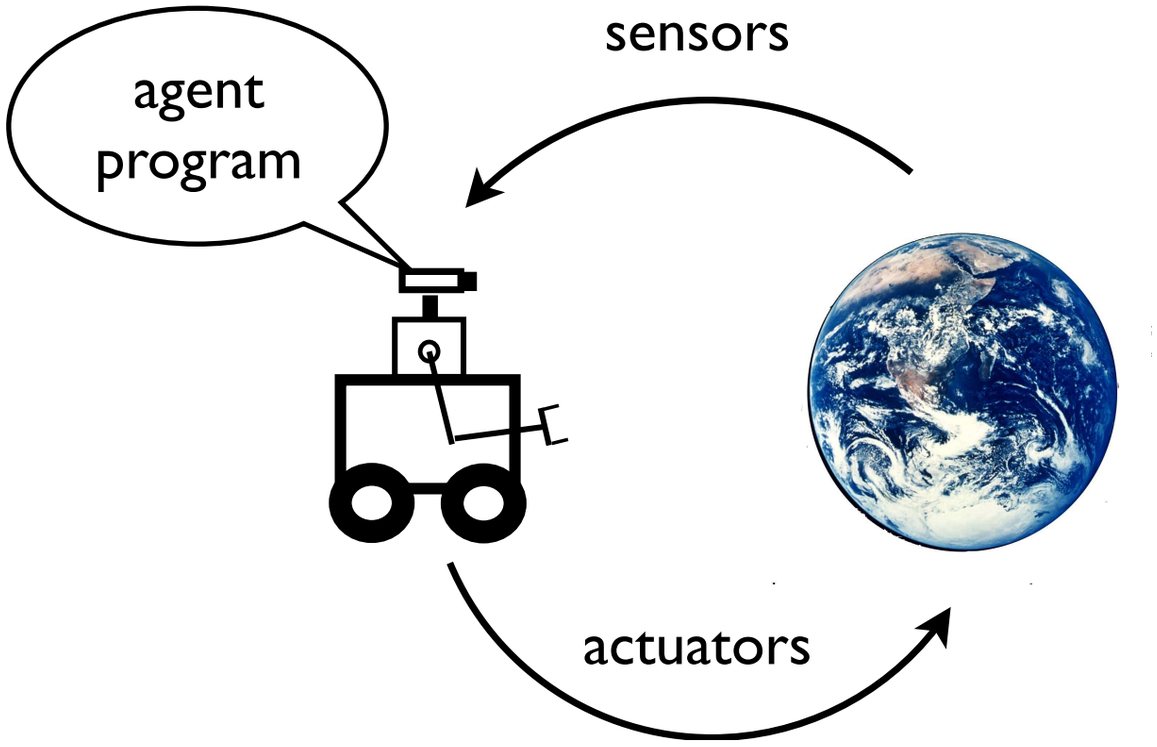
Models

A model is a formal specification of a ***class*** of problems.

- Not a particular problem.
- Captures (abstractly) the essential components of class.
- *Generalizes across* robots, environments, utility functions.

A model is ***not*** an algorithm.

- *Harder to get model right than algorithm right.*
- Algorithm often follows from model.



model

robot

Models

Properties of models:

- They are always wrong.
- They are sometimes useful.

- They are never "real".
  - AI communities often make this mistake. Models are not real. Only robots are real.

- They make assumptions.
  - These assumptions are often wrong.
  - Sometimes we make them anyway.

Doing Research

When faced with a hard (robot) research problem:

FIRST formalize the problem in the most widely applicable way.

- Capture the essential properties.
- Get the model right.
  - In full knowledge that it is wrong/approximate.

- NEVER be idiosyncratic.
  - This specific problem may never occur again.

THEN gain insight into the structure that the model exposes.

THEN develop/apply a general-purpose algorithm.