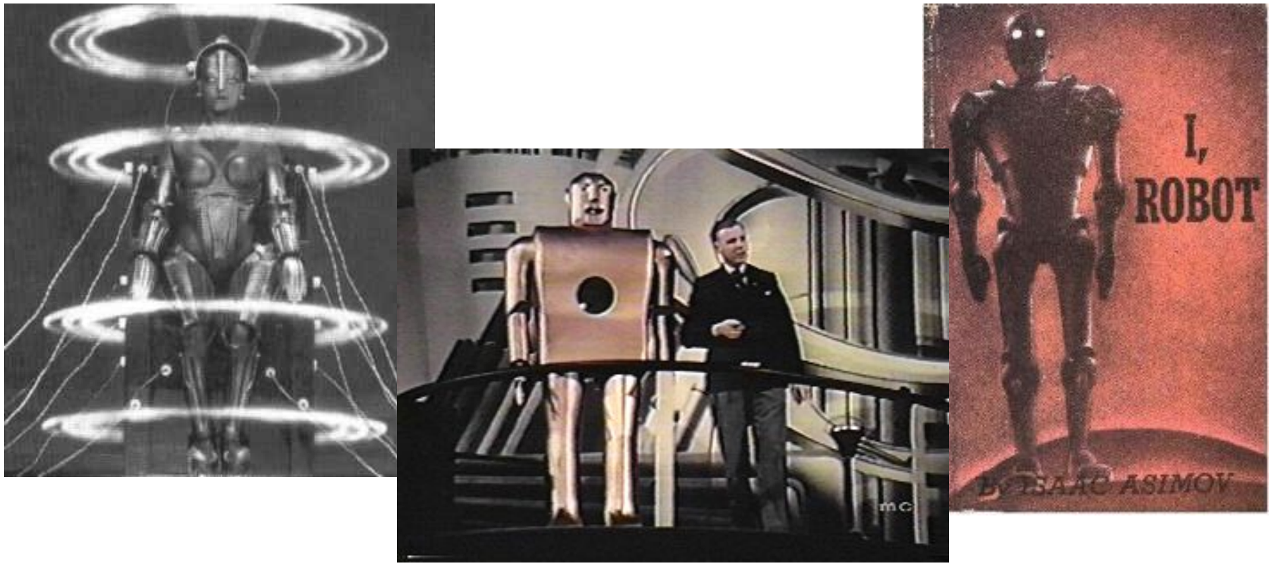
Robot Intelligence

Leslie Pack Kaelbling
MIT CSAIL

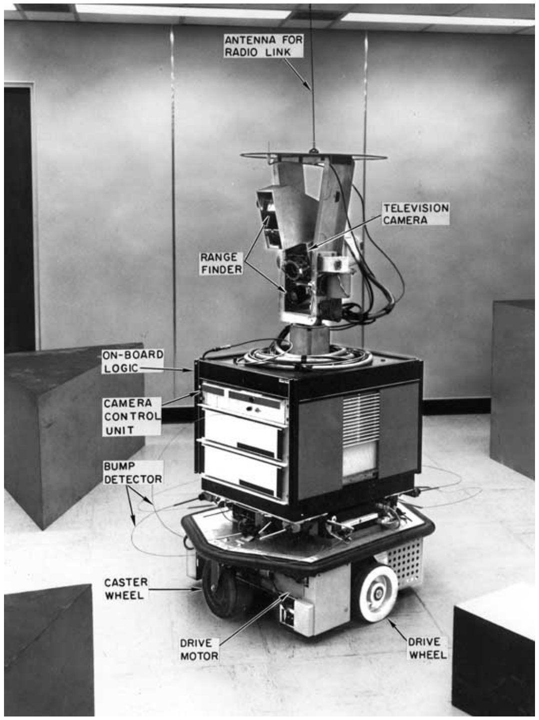
A dream of robots



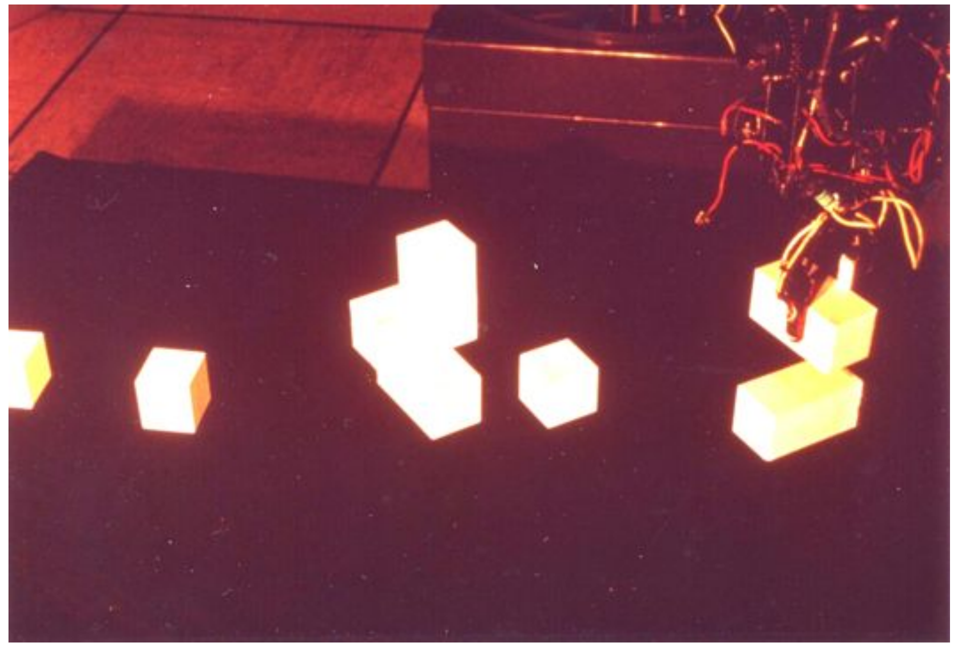
Commercial reality

Research frontier

Robots and AI were once very close



SRI Shakey



MIT Copy Demo

Robotics drove advances in artificial intelligence: planning, learning, reasoning, vision, natural language….

Should we give it another try?

The best of times

Good robot hardware

- range sensors, cameras, actuators, …

Fast computers

Good fundamental algorithms for

- robot motion planning, visual object recognition, …

Technical advances in

- probabilistic inference, machine learning, knowledge representation

The worst of times

Super-human robot fallacy:

- Focus on optimality limits our vision

Fragmented research community

- Subfields with individual standards, vocabulary, benchmarks
- Pieces won't fit together

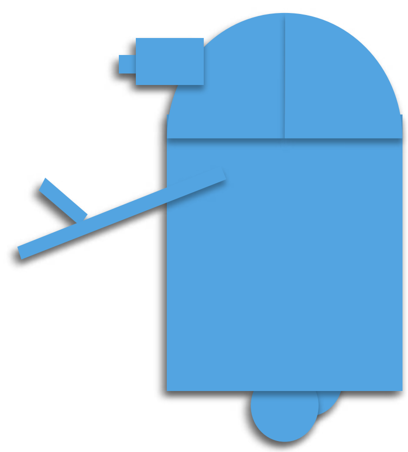
Many other attractive and important applications

- Web applications, data mining, finance, …

The age of wisdom

How to build the **'central' computational mechanisms** for

- **closed-loop control** of a system with
- **sensors and actuators** that has
- **long-term goal-directed** interactions with
- a **complex**
- **imperfectly predictable** external environment



Three technical levers

Compact description of functions and sets in large spaces

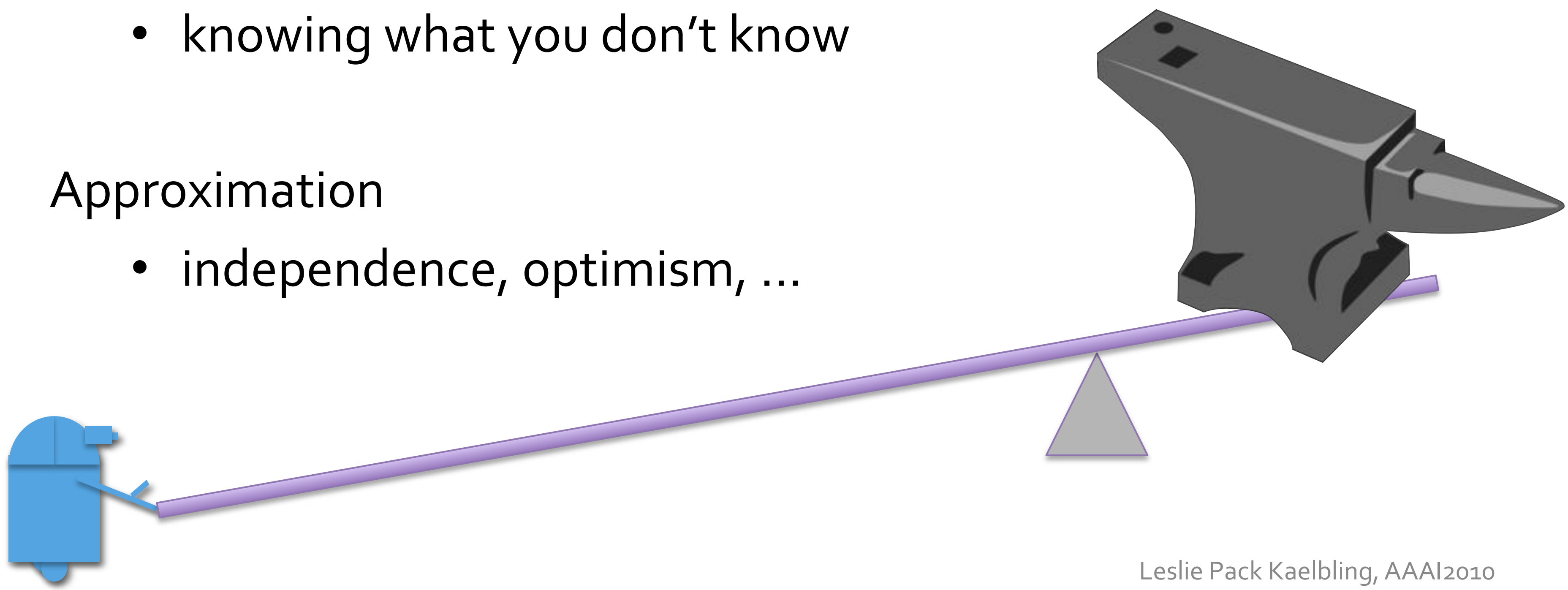
- continuity, geometry
- factoring, logical languages

Explicit representation of uncertainty

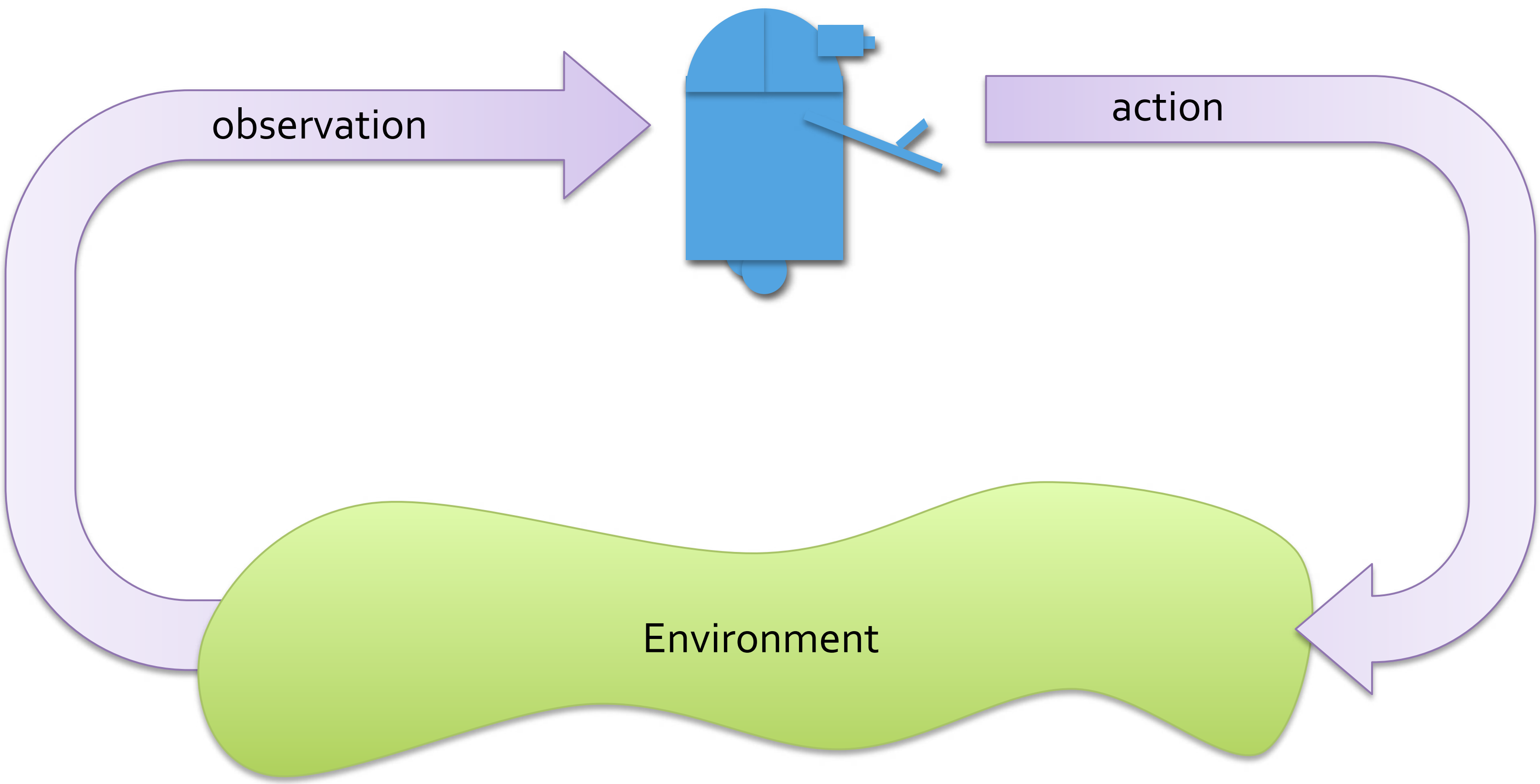
- knowing what you don't know

Approximation

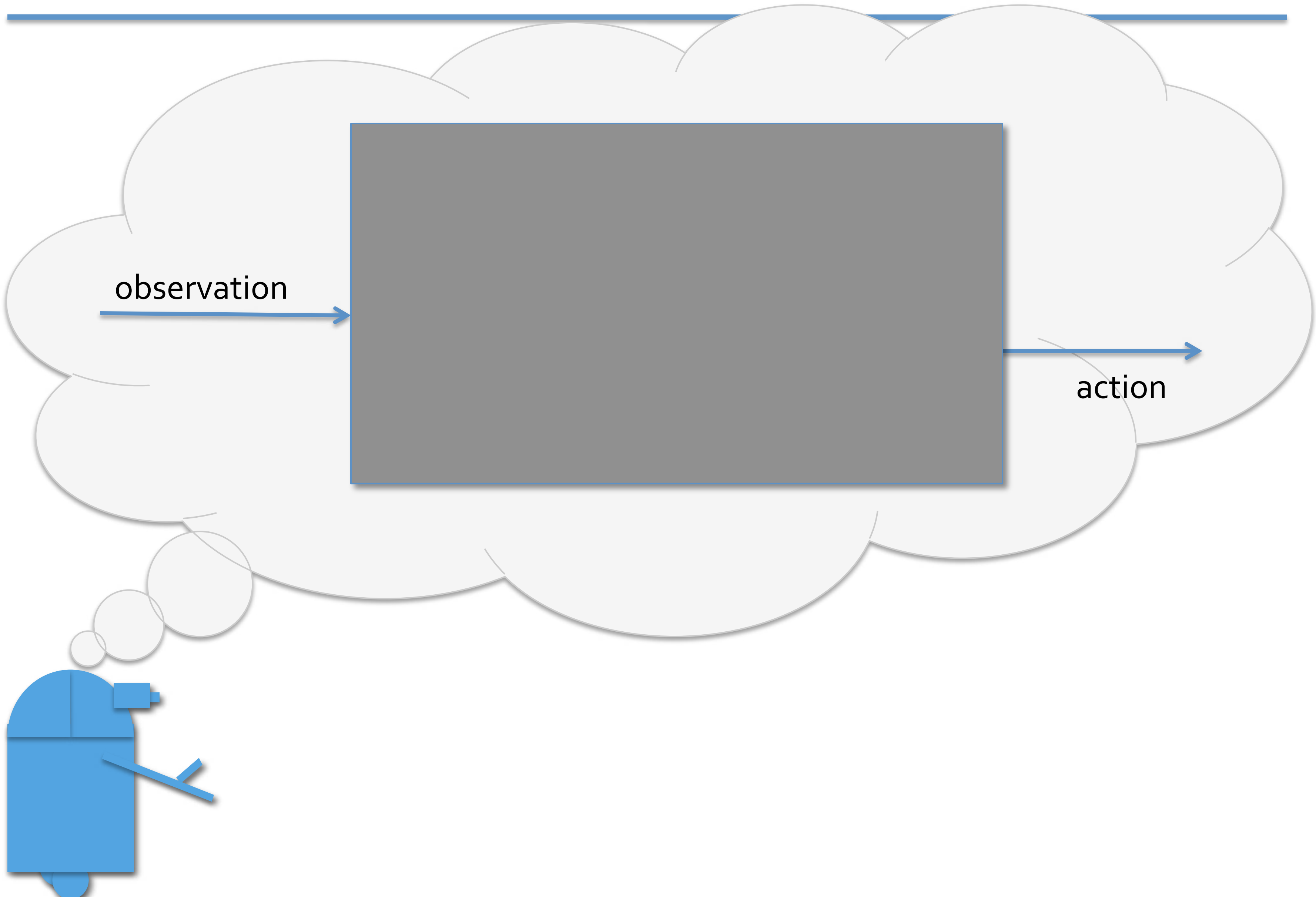
- independence, optimism, …



Interaction with an external environment



What to learn? What to build in?



Prior + Experience = Learned competence

How 'big' is the prior? Where does it come from?

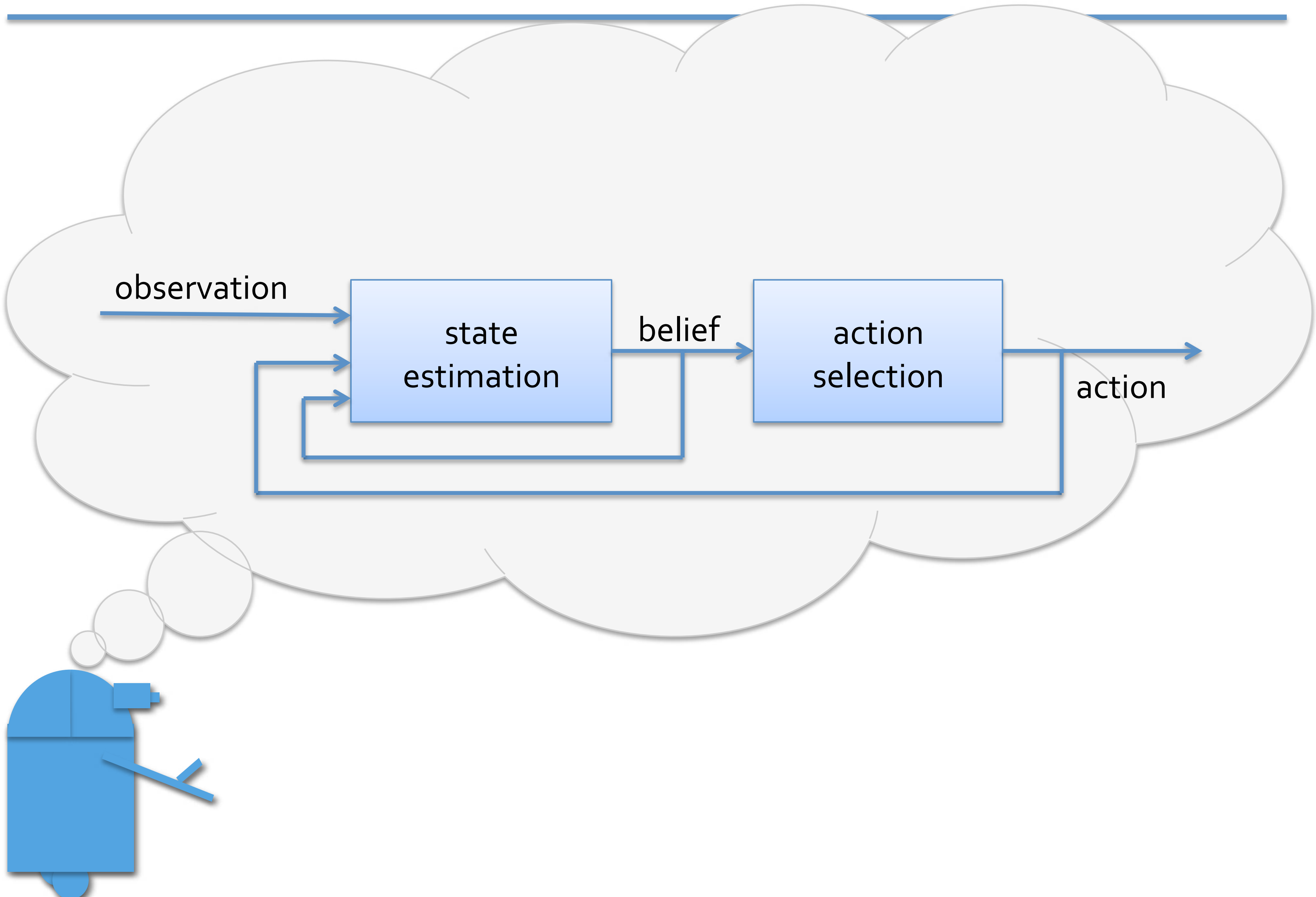
Engineers must do for robots what evolution did for us

- Build in architectural constraints and fundamental truths (e.g. physical laws)

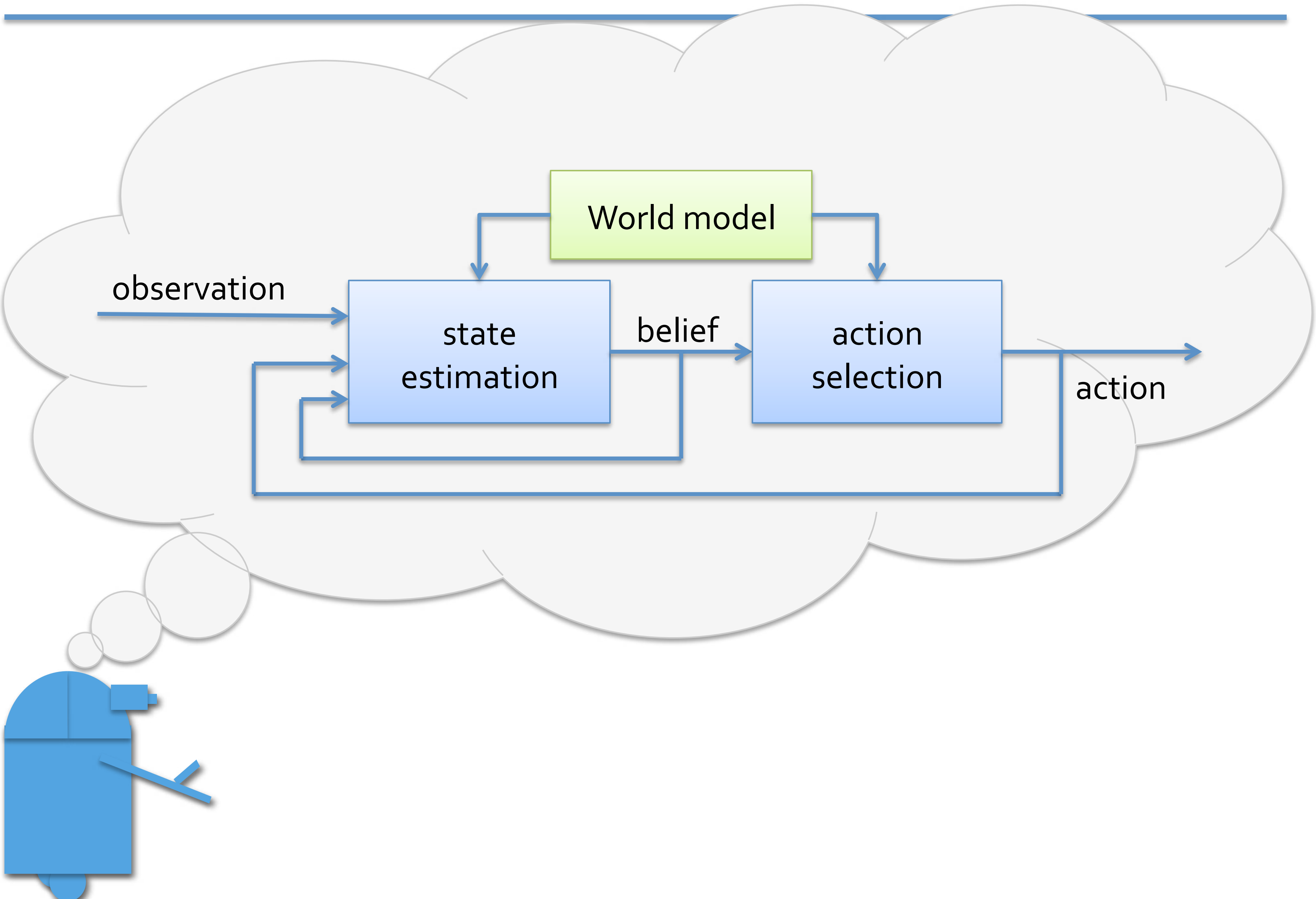
Agents must learn niche-specific competences
(and things the engineers can't articulate)

- sensory-motor loops
- world model at several levels of abstraction
- strategies for managing internal computation

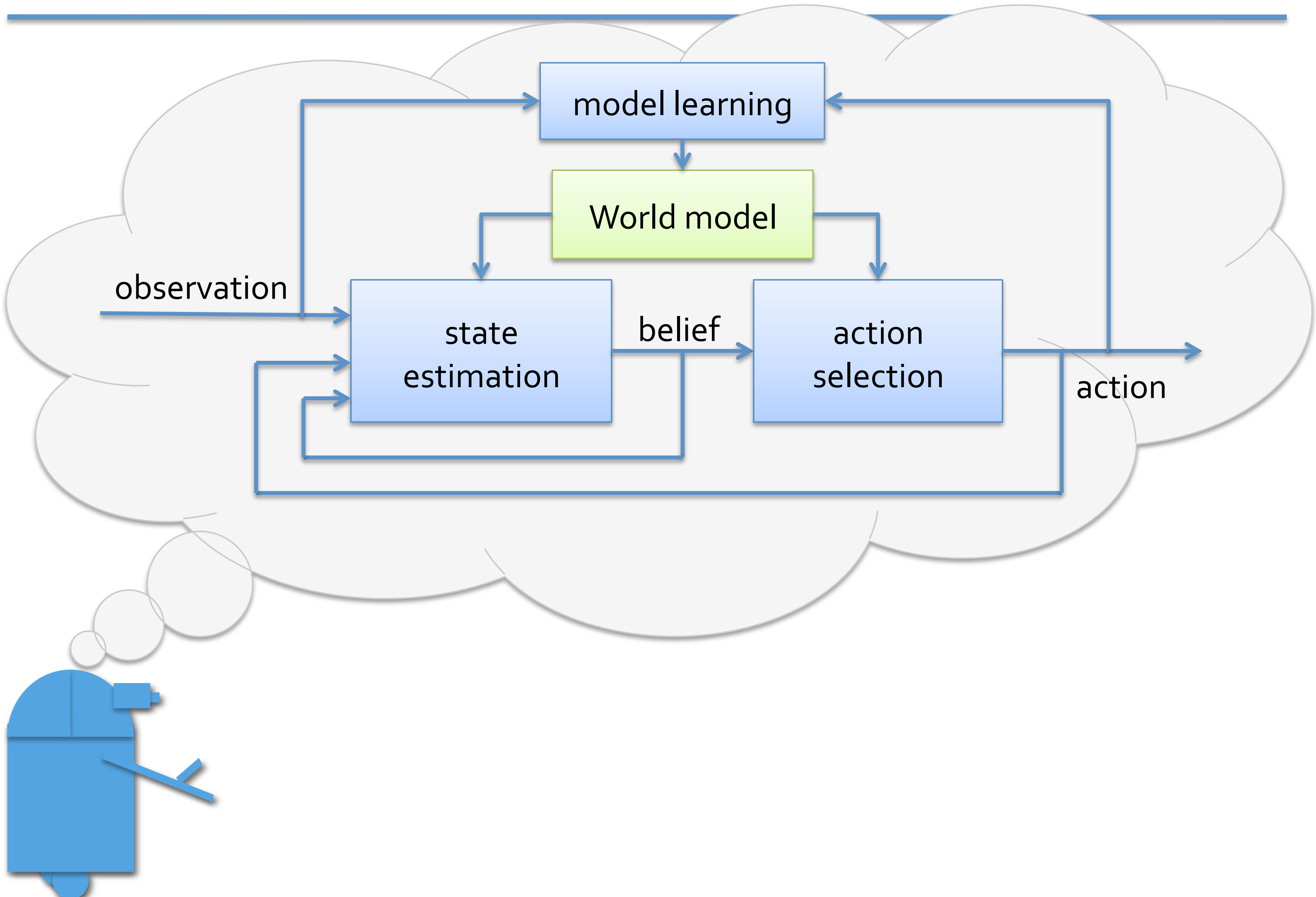
Internal architecture



Internal architecture



Internal architecture



This talk: getting leverage

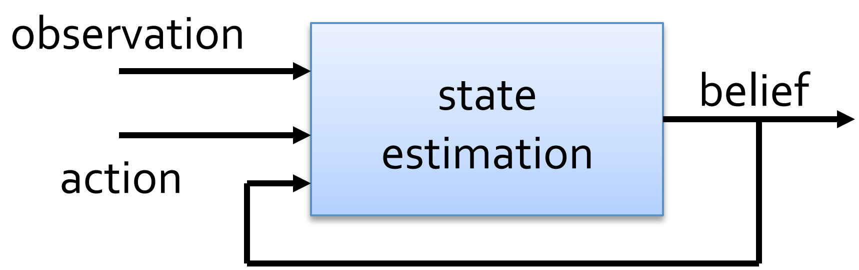
Sample points in the technical space

- **state estimation** method
  - combining logic, probability, and approximation
- **3 action selection** examples
  - combining logic or geometry, probability, and approximation
- **model learning** method
  - combining logic, probability, and approximation

Important areas neglected:

- perception, actuation, language and human interaction, multi-agent systems, …

The epoch of belief and of incredulity



State estimation: Explicitly represent state of knowledge about external environment using probability

Joint work with Luke Zettlemoyer and Hanna Pasula

State estimation

Problem: given history of past observations and actions, what do we know about the current state of the world?

Lazy: store history of observations, do inference when necessary

Eager: maintain an explicit representation of the current distribution over the state of the world ("filtering")

Filtering: Bayesian belief state update

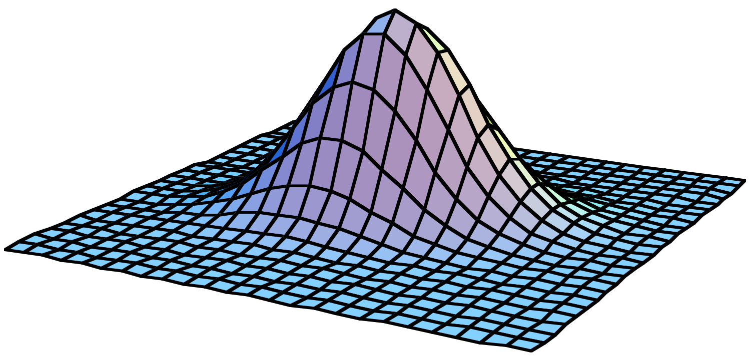
Update after executing an action and receiving an observation

$$\Pr(s_{t+1} \mid a_t, o_{t+1}) \propto \Pr(o_{t+1} \mid s_{t+1}) \sum_{s_t} \Pr(s_{t+1} \mid s_t, a_t) \Pr(s_t)$$

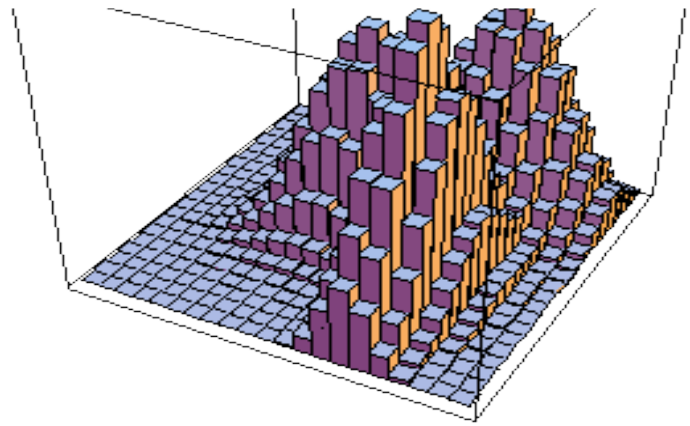
posterior belief | observation model | transition model | prior belief

Representing the belief state

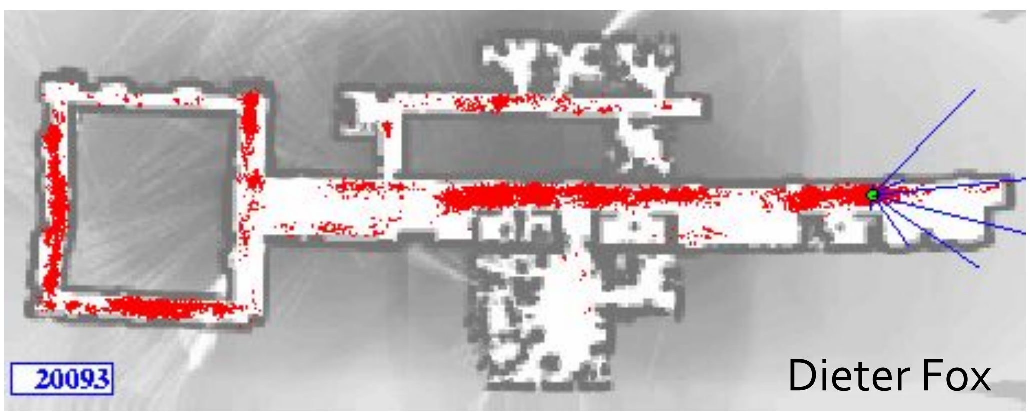
Gaussian



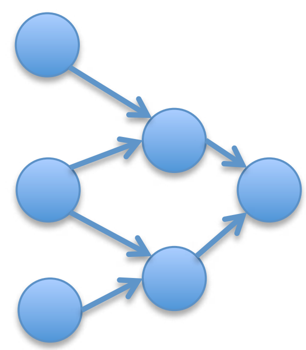
Histogram



Set of particles



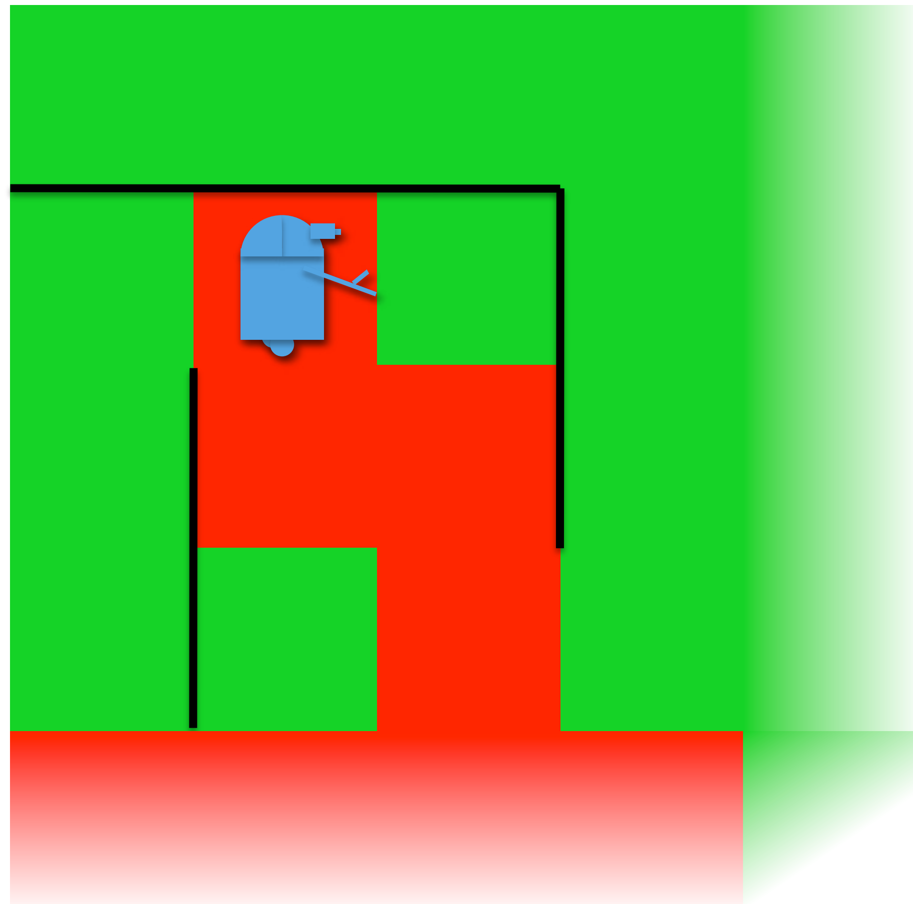
Bayesian network



Leslie Pack Kaelbling, AAAI2010

A big (toy) world

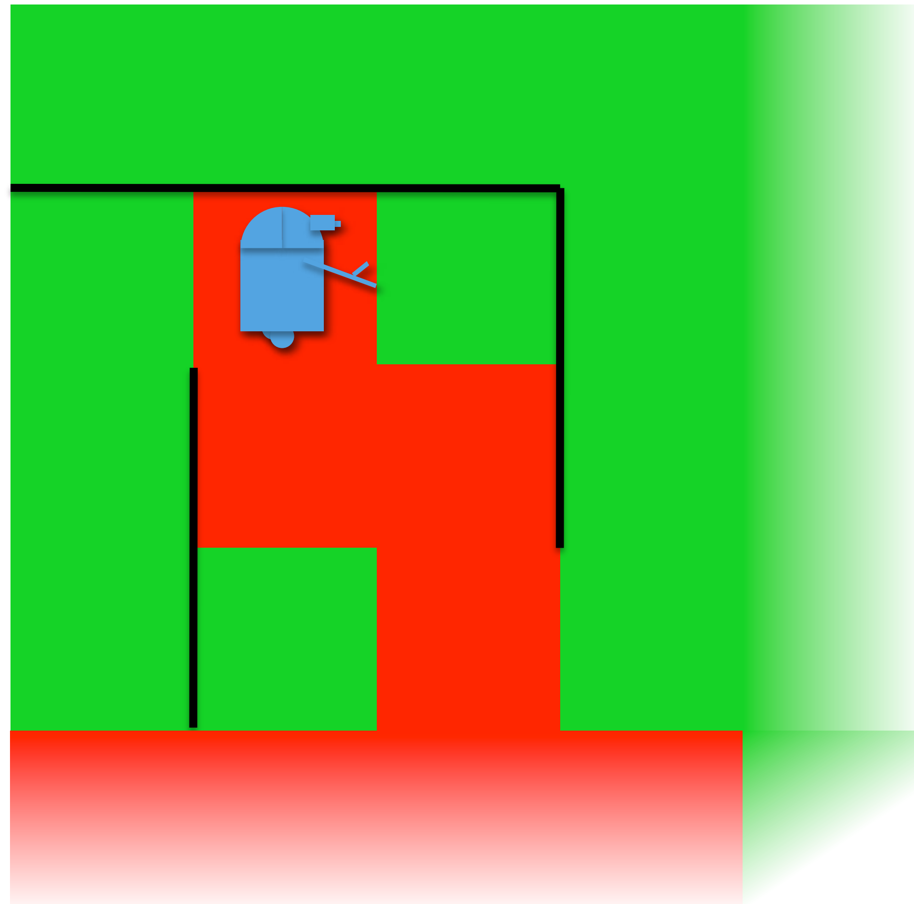
- dimensions are unknown (possibly infinite)
- walls between some locations
- locations have appearance
- R moves (with error) through the world
- R observes (with error) the color at his location



A day in the life

R is booted up,

- sees a red square
- tries to move right
- sees a green square



What does R know about the world?

Combine logic and probability to get compact representations of beliefs in complex domains

First-Order particle filtering

Hypothesis: set of states that are indistinguishable based on the history of observations and actions

$$\Pr(s \mid o_{0:t}, a_{1:t}) \propto \Pr(o_{0:t} \mid s, a_{1:t}) \Pr(s)$$

posterior belief

observation and transition probability

prior belief

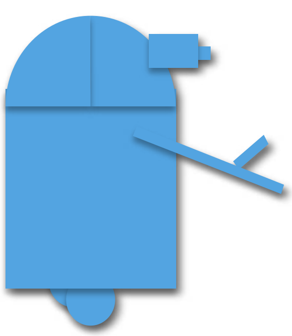
Use logic sentences to describe hypotheses

Only represent likely hypotheses

R wakes up

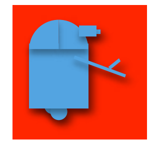
- One hypothesis

True

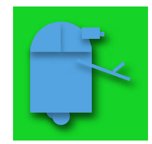


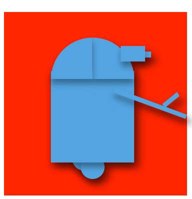
R sees a red square

$\exists x.at(R, x) \wedge red(x)$ — 0.8



$\exists x.at(R, x) \wedge green(x)$ — 0.2

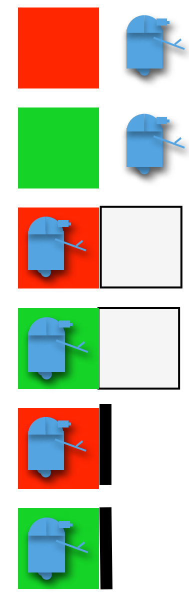


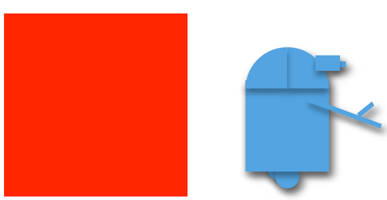


P(see red | at red square) = 0.8

R tries to move right

$\exists x, y. at(R,y) \wedge red(x) \wedge rightOf(y,x)$	.504
$\exists x, y. at(R,y) \wedge green(x) \wedge rightOf(y,x)$	.126
$\exists x, y. at(R,x) \wedge red(x) \wedge rightOf(y,x)$	.056
$\exists x, y. at(R,x) \wedge green(x) \wedge rightOf(y,x)$	.014
$\exists x. at(R,x) \wedge red(x) \wedge \neg \exists y. rightOf(y,x)$	0.24
$\exists x. at(R,x) \wedge green(x) \wedge \neg \exists y. rightOf(y,x)$	0.06



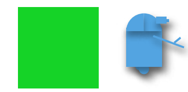


Prob wall to right: 0.3
Prob fail to move (if no wall): 0.1
Prob fail to move (if wall): 1.0

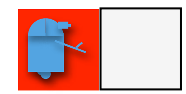
R tries to move right: sample

$\exists x, y. at(R, y) \wedge red(x) \wedge rightOf(y, x)$

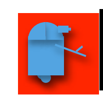
$\exists x, y. at(R, y) \wedge green(x) \wedge rightOf(y, x)$

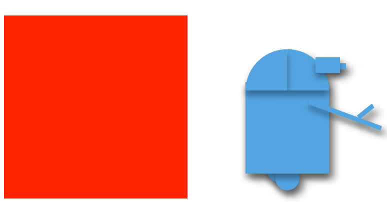


$\exists x, y. at(R, x) \wedge red(x) \wedge rightOf(y, x)$



$\exists x. at(R, x) \wedge red(x) \wedge \neg \exists y. rightOf(y, x)$

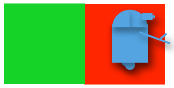


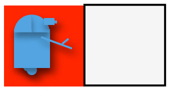


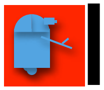
Prob wall to right: 0.3
Prob fail to move (if no wall): 0.1
Prob fail to move (if wall): 1.0

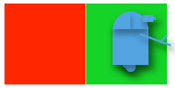
R sees a green square

$\exists x. at(R, y) \wedge red(x) \wedge rightOf(y, x) \wedge red(y)$ — .147 

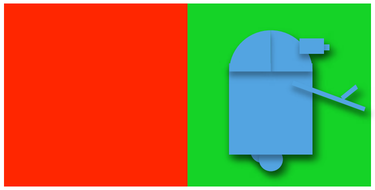
$\exists x. at(R, y) \wedge green(x) \wedge rightOf(y, x) \wedge red(y)$ — .036 

$\exists x, y. at(R, x) \wedge red(x) \wedge rightOf(y, x)$ — .016 

$\exists x. at(R, x) \wedge red(x) \wedge \neg \exists y. rightOf(y, x)$ — .069 

$\exists x. at(R, y) \wedge red(x) \wedge rightOf(y, x) \wedge green(y)$ — .585 

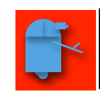
$\exists x. at(R, y) \wedge green(x) \wedge rightOf(y, x) \wedge green(y)$ — .147 



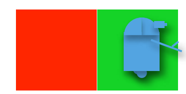
R sees a green square: sample

$$\exists x. at(R, y) \wedge red(x) \wedge rightOf(y, x) \wedge red(y)$$

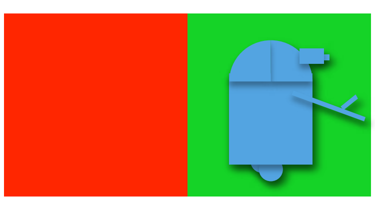
$$\exists x. at(R, x) \wedge red(x) \wedge \neg \exists y. rightOf(y, x)$$



$$\exists x. at(R, y) \wedge red(x) \wedge rightOf(y, x) \wedge green(y)$$



$$\exists x. at(R, y) \wedge green(x) \wedge rightOf(y, x) \wedge green(y)$$



Technical Story

Rao-Blackwellization:

$$E_{Pr(x_1,x_2)} f(x_1,x_2) = E_{Pr(x_2)} E_{Pr(x_1|x_2)} f(x_1,x_2)$$
$$\approx \frac{1}{n} \sum_{\text{samples from } Pr(x_2)} E_{Pr(x_1|x_2)} f(x_1,x_2)$$

For us:

$$x_2 : \text{logical partition}$$
$$x_1 : \text{state within the partition}$$
$$f(x_1,x_2) : \text{Am I in room 6?}$$

created dynamically depending on observations

depends only on prior

Many other possible f

Demand-driven complexity

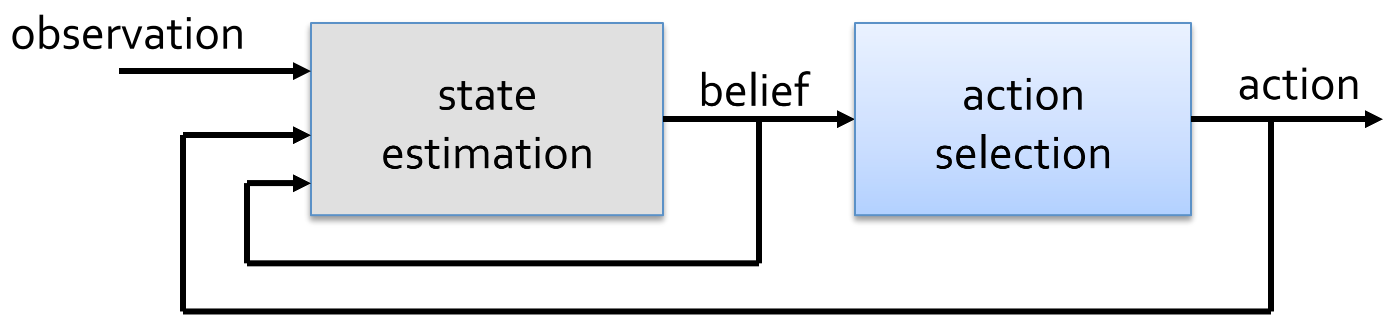
Logical particle filter:

- complexity of **logical form** driven by observations
- concentrates on **most probable** part of the space

Be lazier!

- focus on small set of objects and properties relevant to current goal
- dynamically change focus
- use observation history to initialize new filters

Action selection



Plan in belief space:

- every action gains information and changes the world
- changes are reflected in new belief via estimation
- goal is to believe that the environment is in a desired state

The spring of hope and the winter of despair

In domains that lack terrible outcomes:

- plan assuming actions will result in most likely transition and observation
- replan if expectation is violated at runtime

Great success of FF-Replan at ICAPS probabilistic planning competition

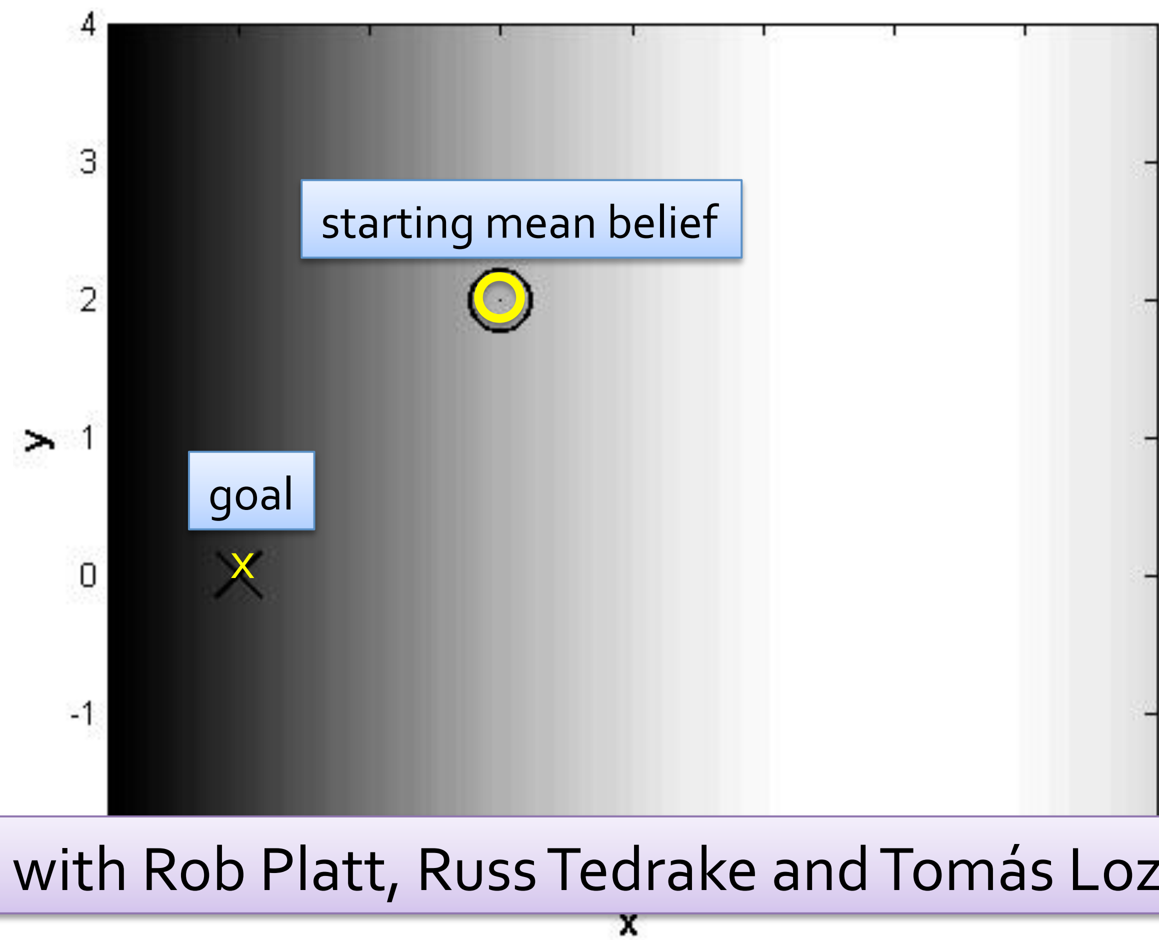
Same principle as feedback control using an idealized model

Optimistic (re)planning in belief space

- **control with state-dependent observation noise**: continuous state, action, observation spaces
- **robot grasping with tactile sensing**: continuous state, action, observation spaces
- **household robot with local observation**: mixed continuous and relational spaces

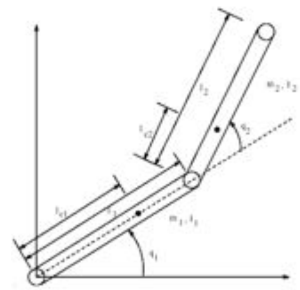
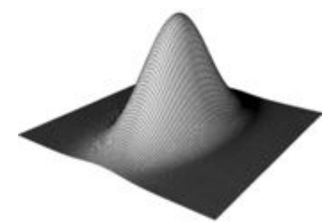
The season of light, the season of darkness

- robot in x, y space
- good position sensing in light regions; poor in dark



Joint work with Rob Platt, Russ Tedrake and Tomás Lozano-Pérez

Control in belief space: underactuated

		
	Acrobot	Gaussian belief:
State space:	$x = \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix}$	$b = \begin{pmatrix} m \\ \Sigma \end{pmatrix}$
Planning objective:	$x_g = \begin{pmatrix} \pi \\ 0 \end{pmatrix}$	$b_g = \begin{pmatrix} x_g \\ 0 \end{pmatrix}$
Underactuated dynamics:	$\ddot{\theta} = f(\theta, \dot{\theta}, u)$	???

Belief space dynamics

Dynamics specify next belief state, as a function of previous belief state and action

- state update: generalized Kalman filter

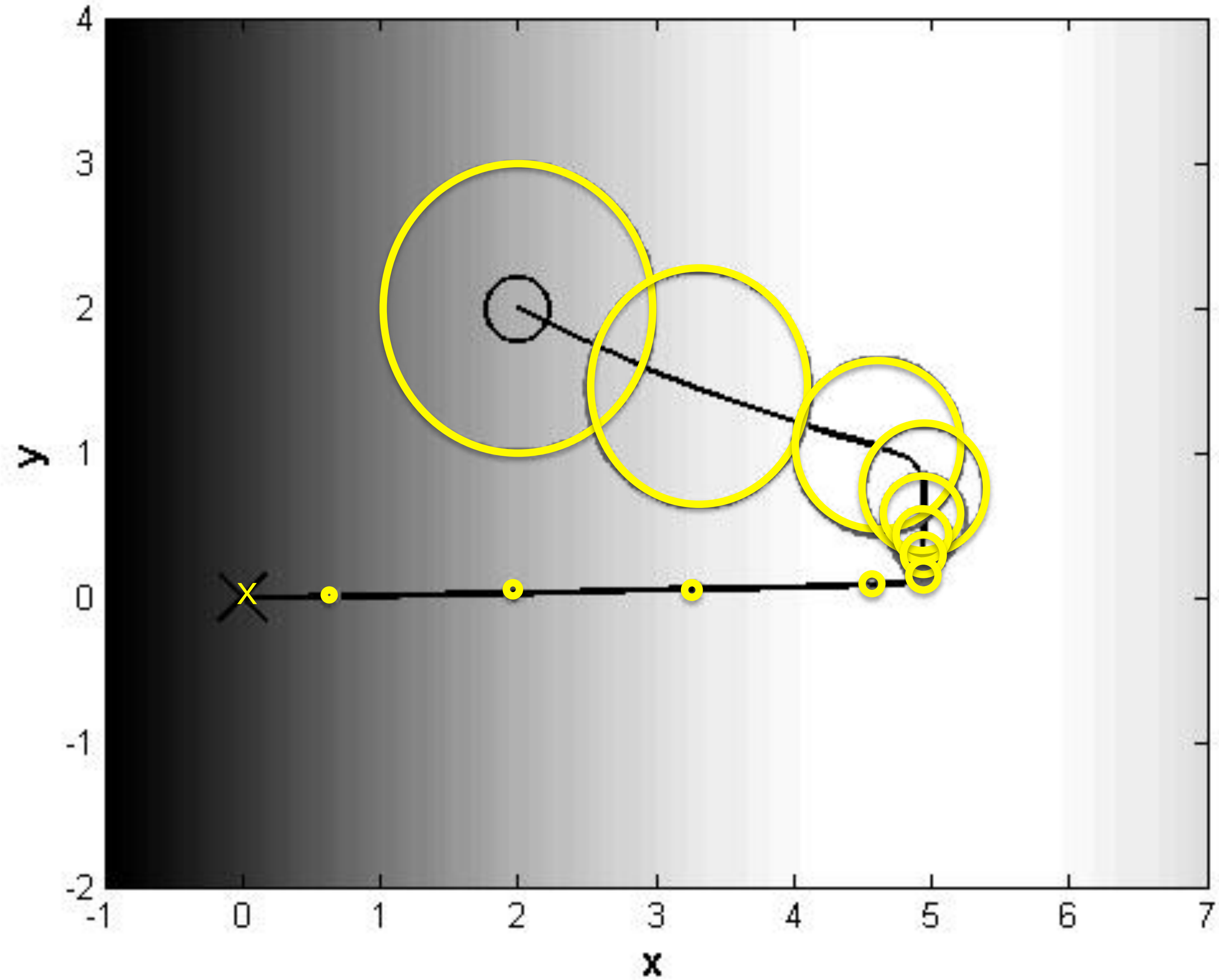
$$(\mu_{t+1}, \Sigma_{t+1}) = GKF(o_t, a_t, \mu_t, \Sigma_t)$$

- substitute expected observation in for actual one
add Gaussian noise

$$\begin{aligned}(\mu_{t+1}, \Sigma_{t+1}) &= F(a_t, \mu_t, \Sigma_t) + N \\ &= GKF(\bar{o}(\mu_t), a_t, \mu_t, \Sigma_t) + N\end{aligned}$$

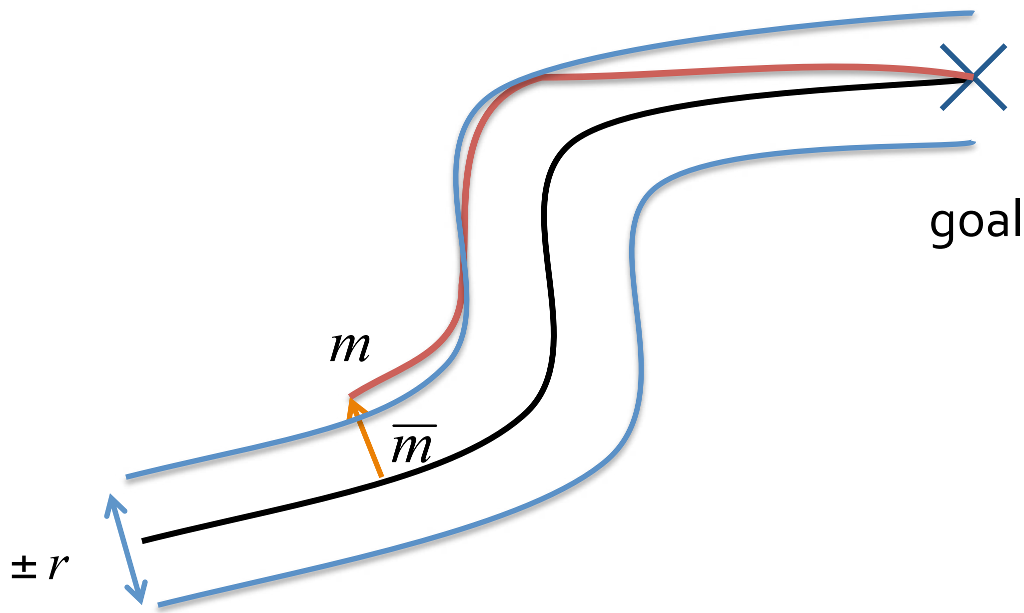
- continuous Gaussian non-linear dynamics:
apply tools from control theory

Light-dark plan

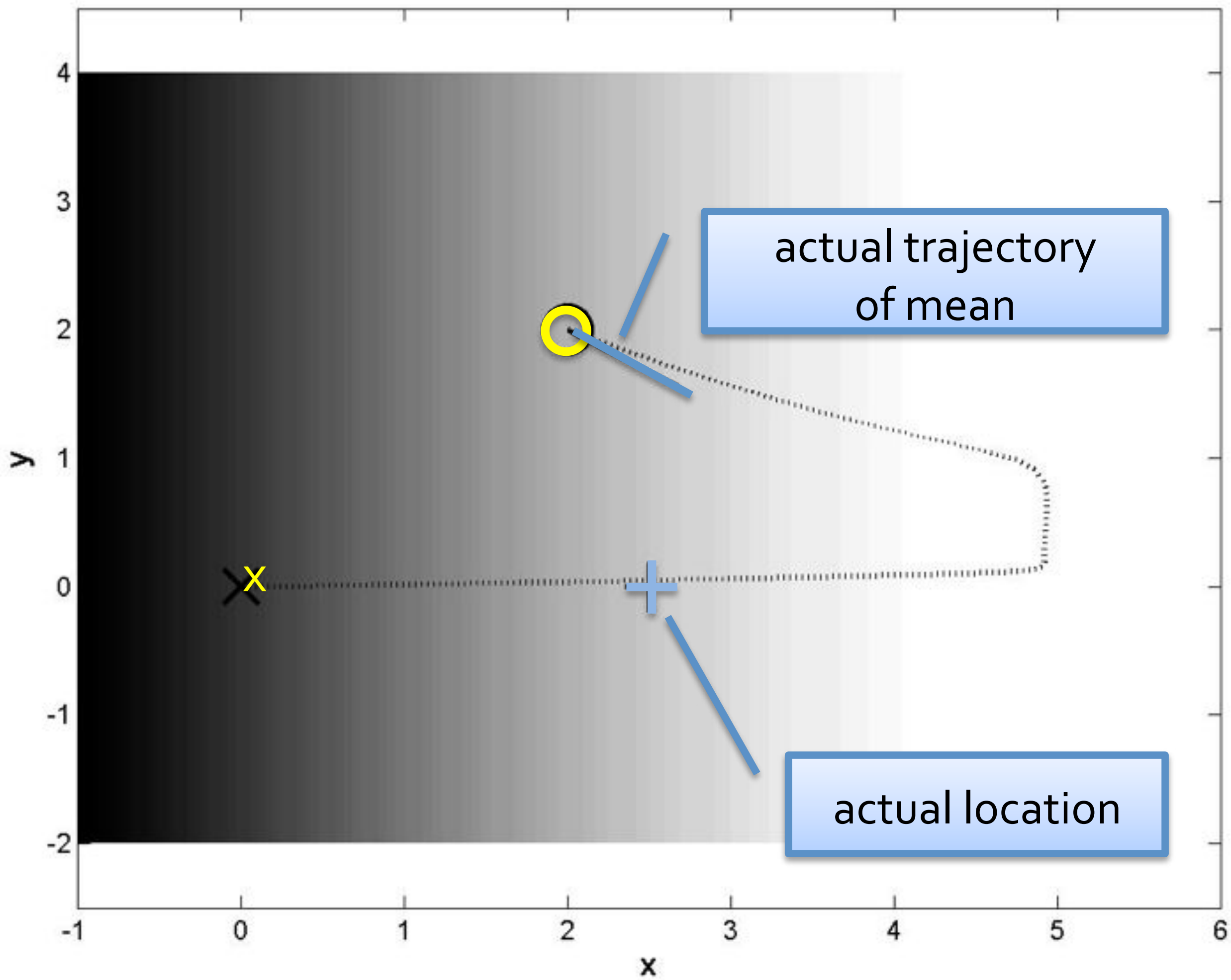


Replanning

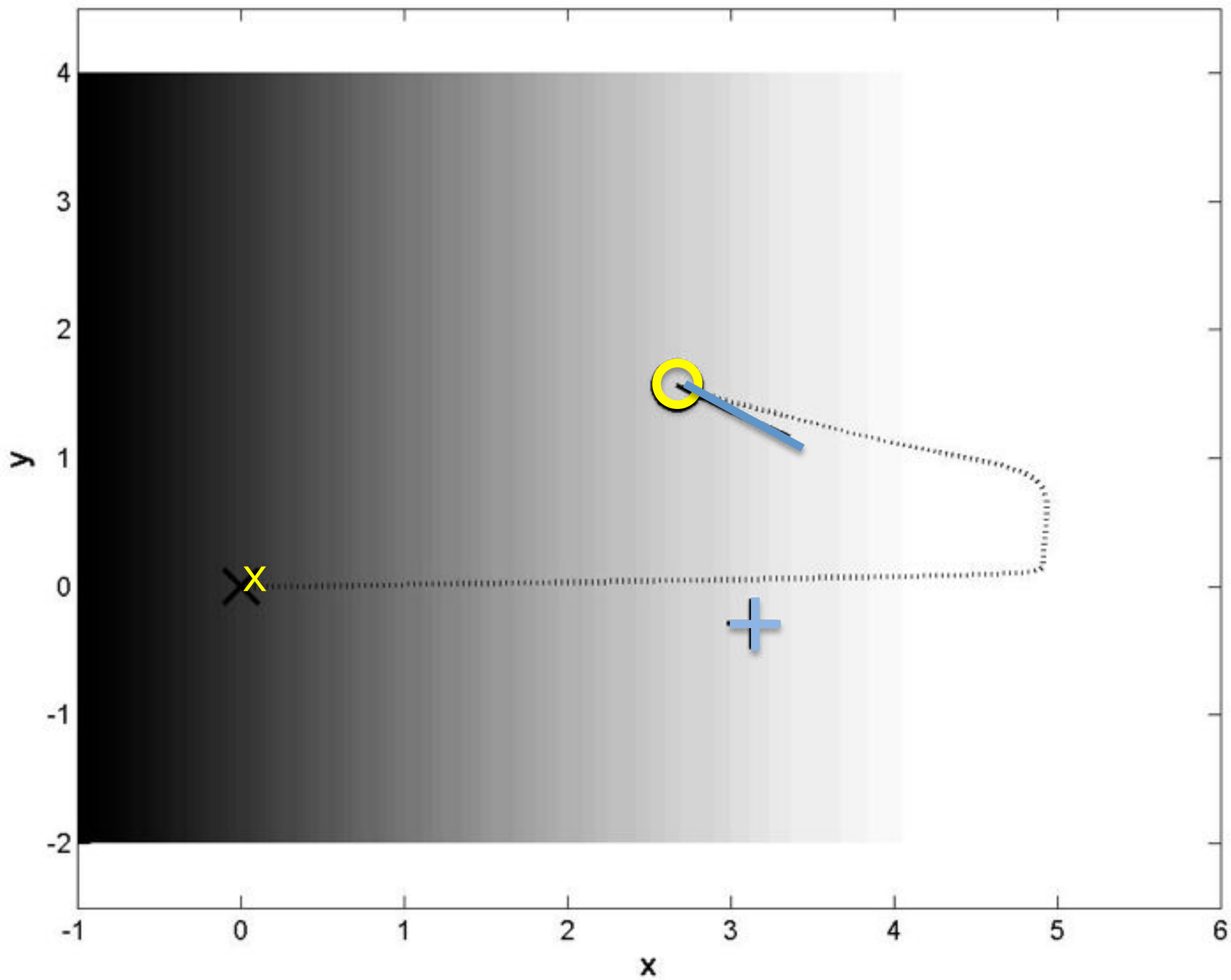
Replan when new belief state deviates too far from planned trajectory



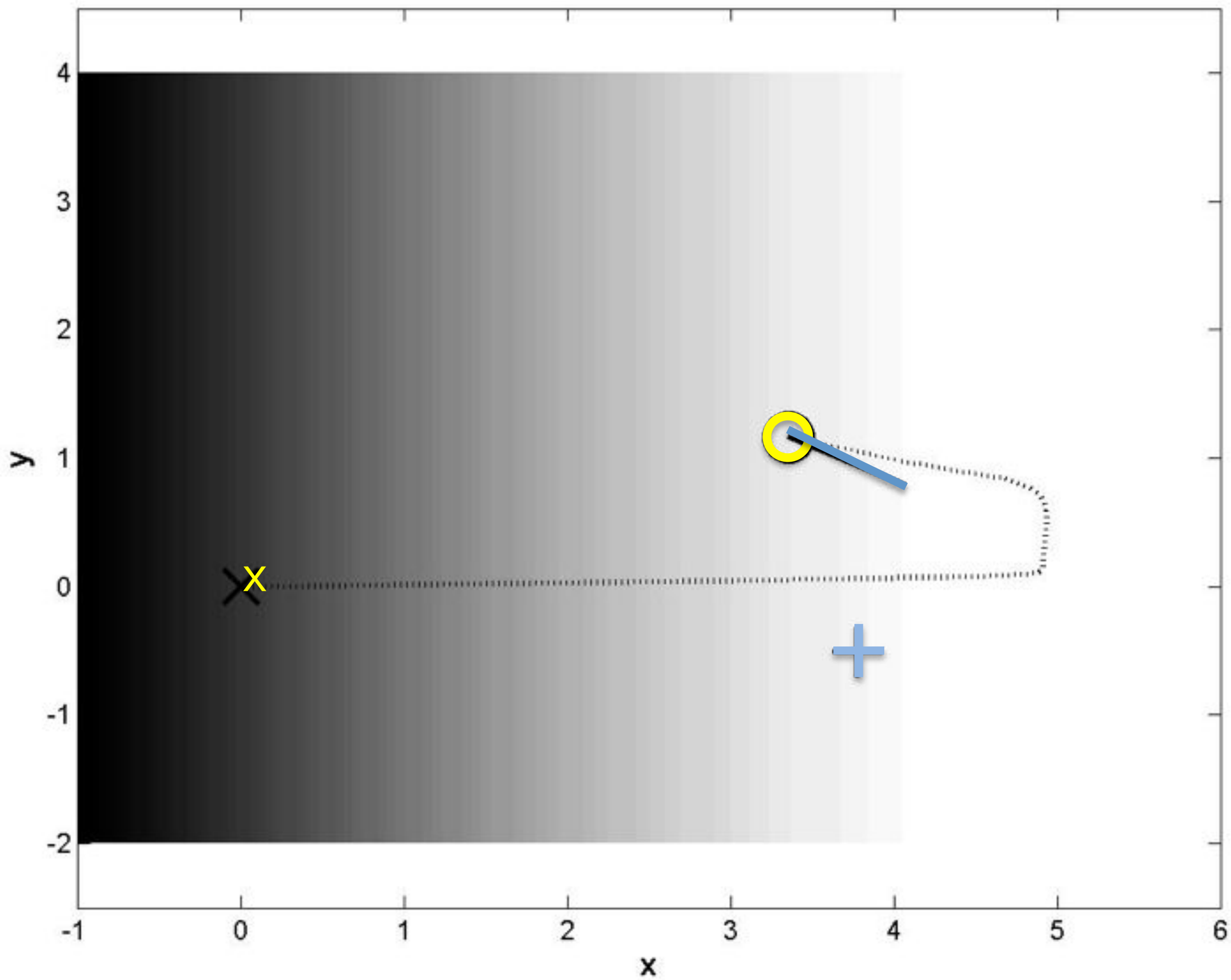
Replanning: light-dark problem



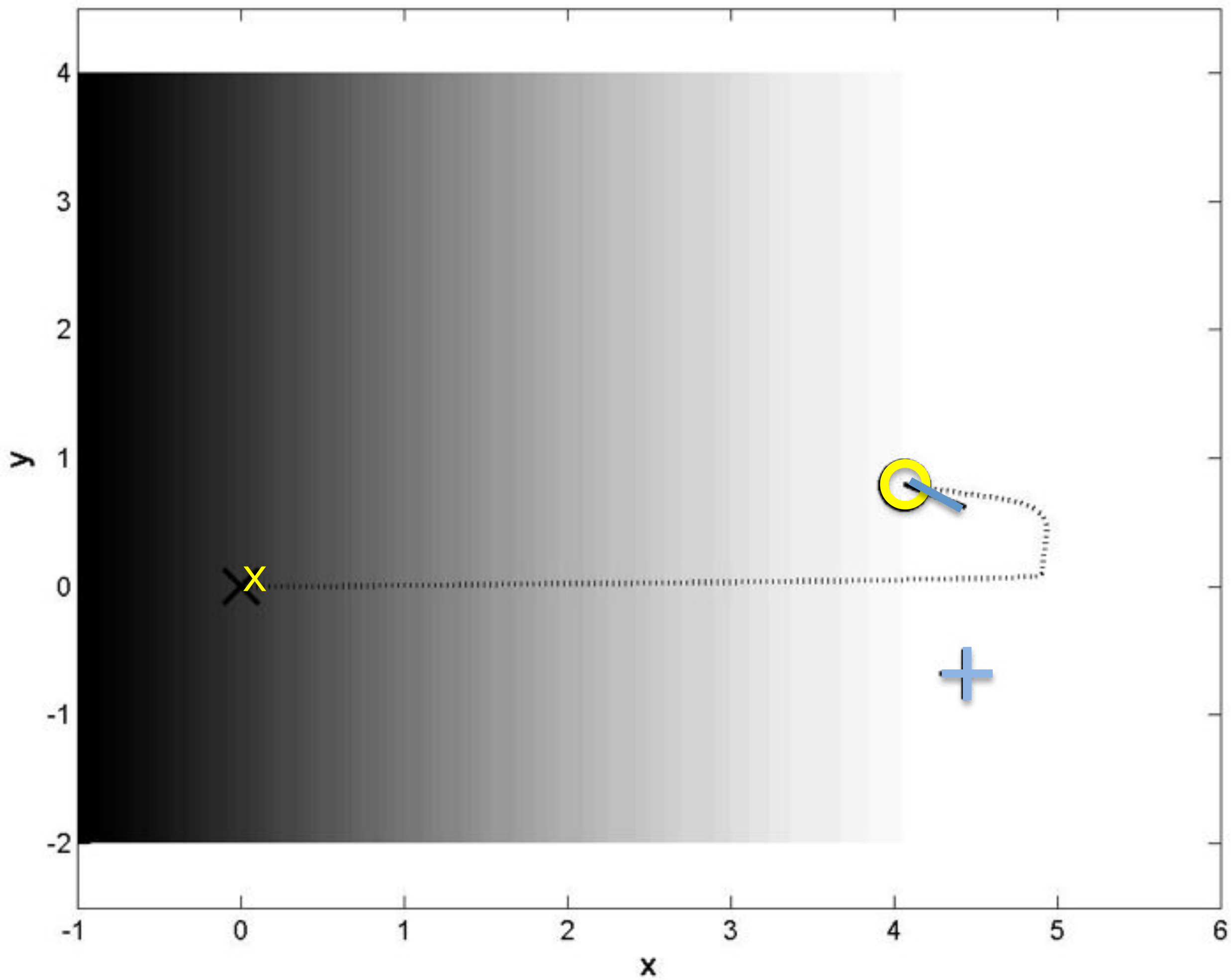
Replanning: light-dark problem



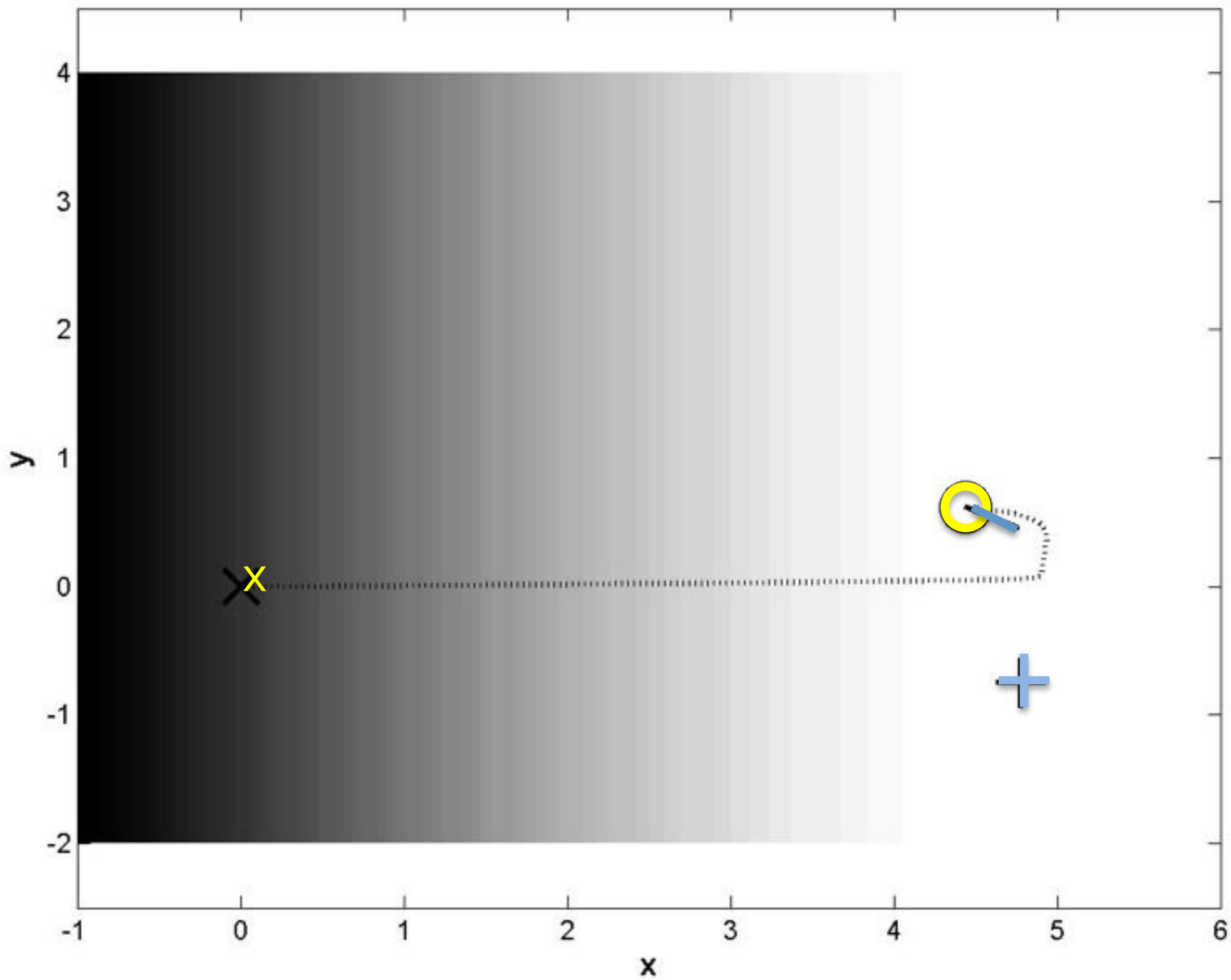
Replanning: light-dark problem



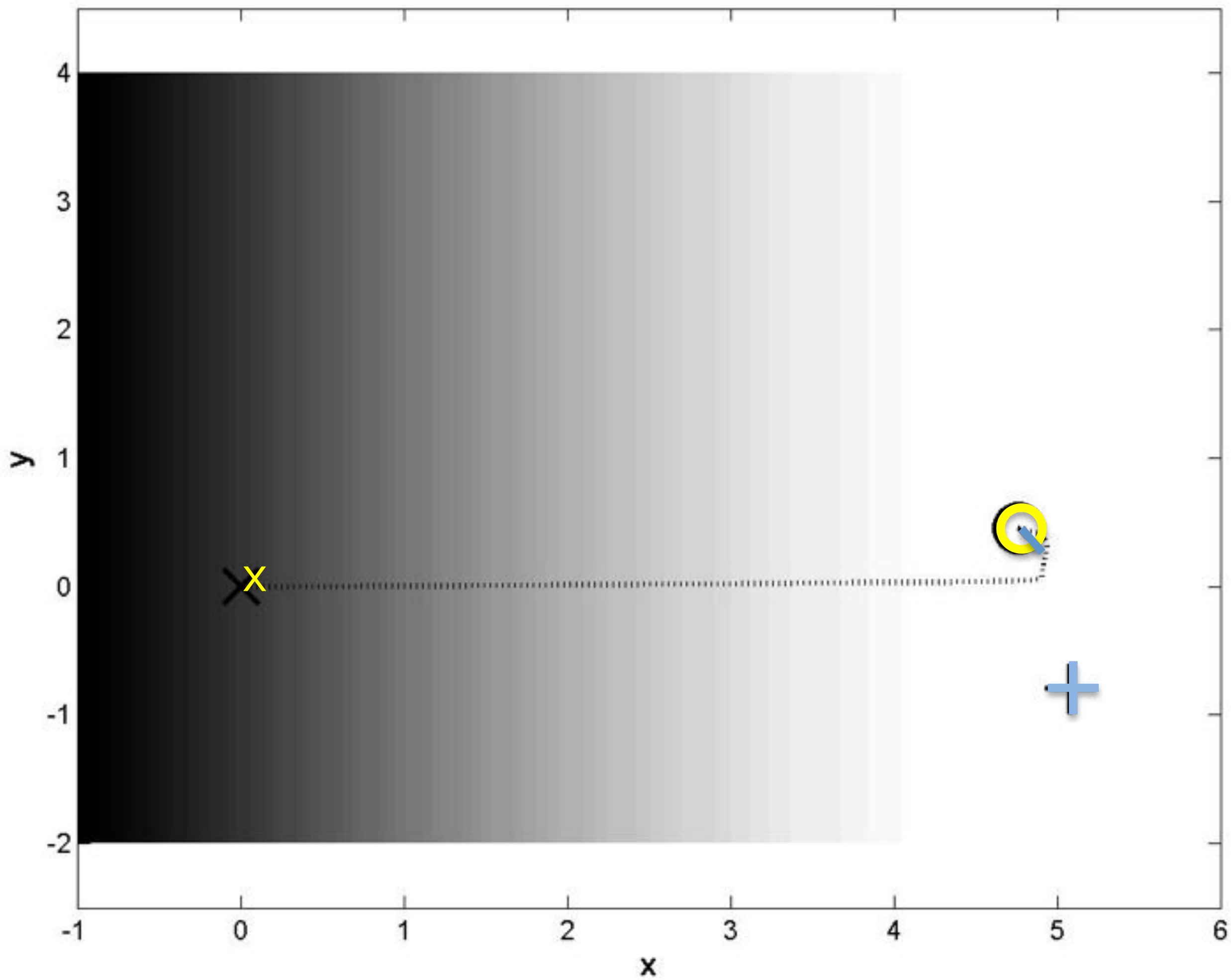
Replanning: light-dark problem



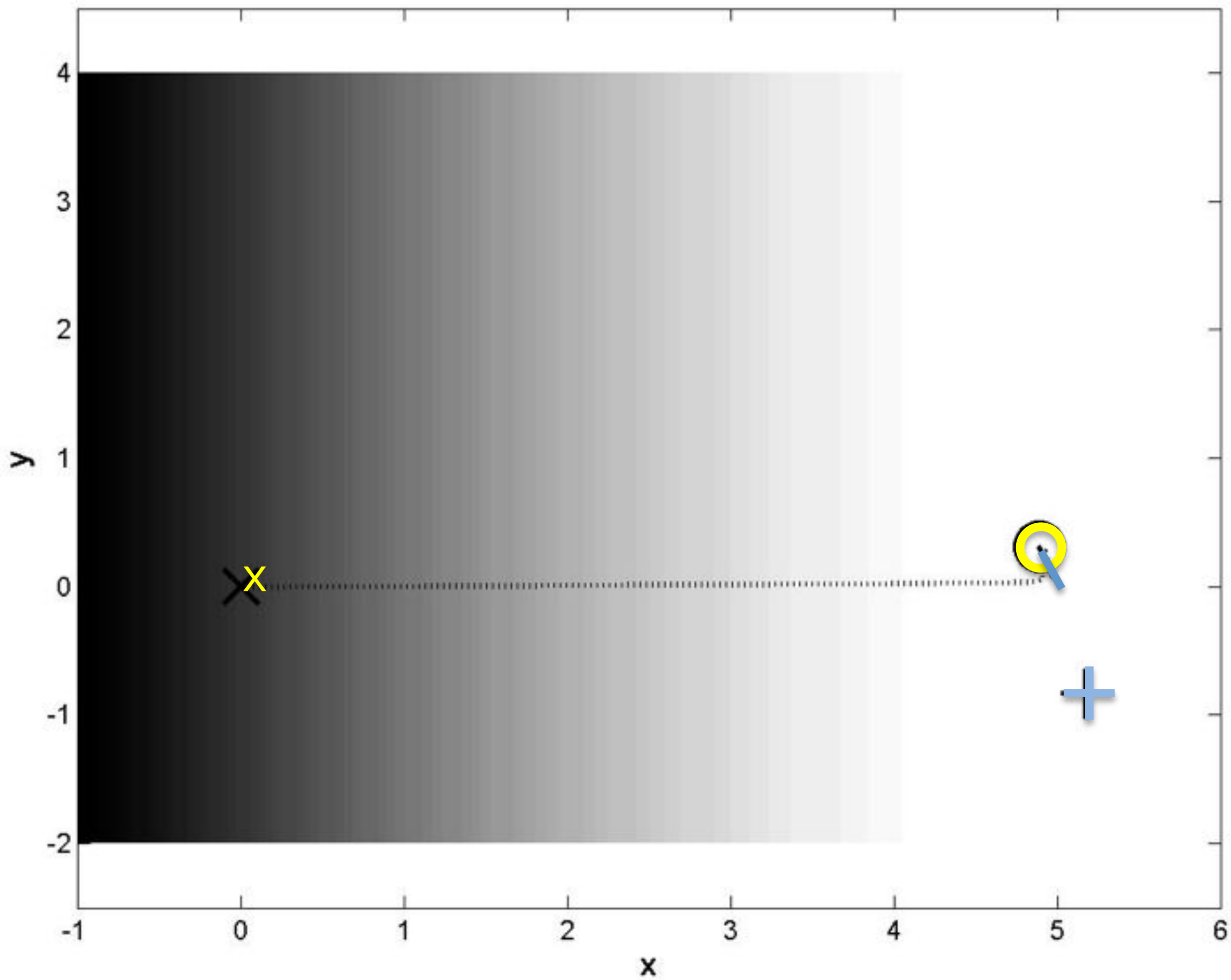
Replanning: light-dark problem



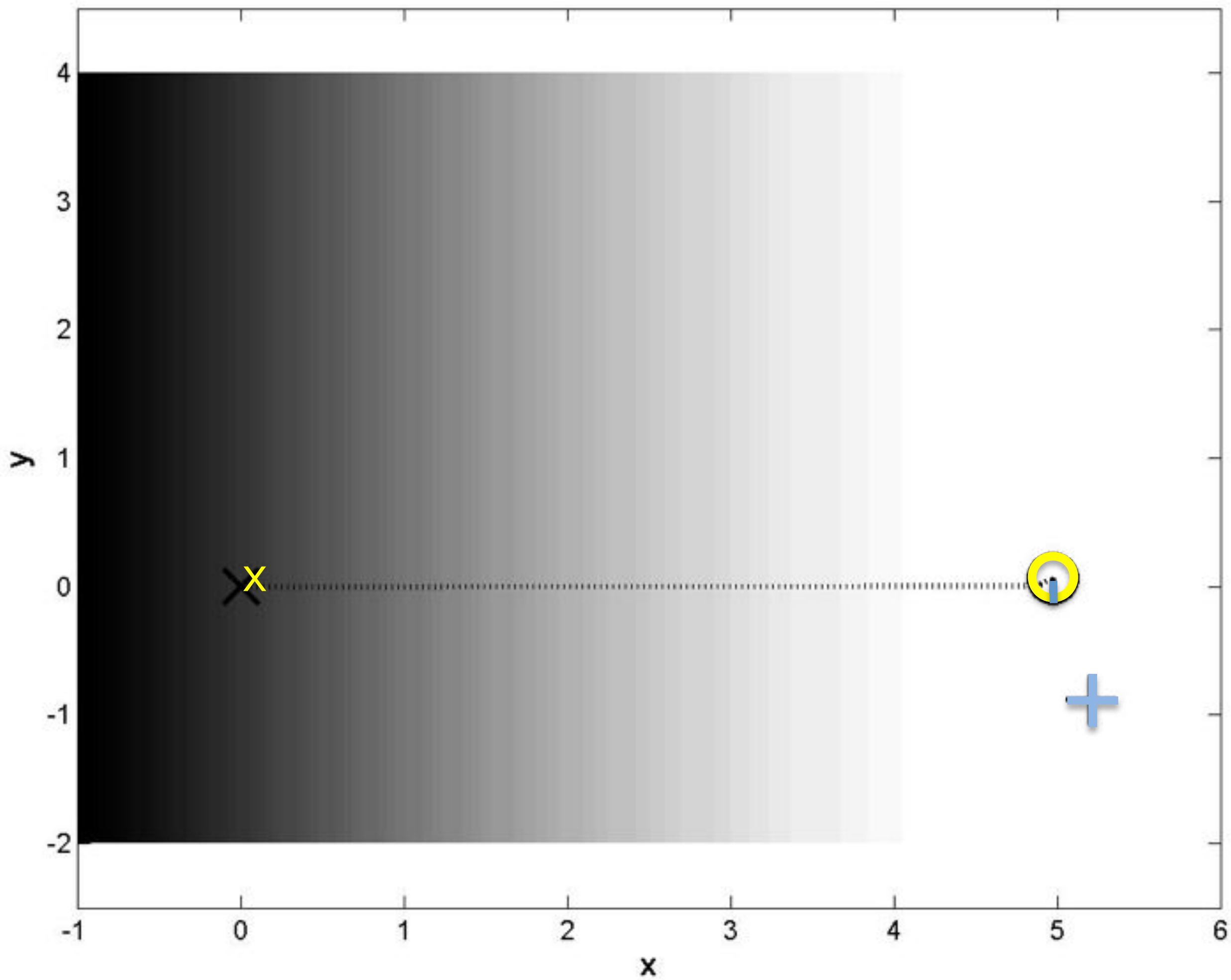
Replanning: light-dark problem



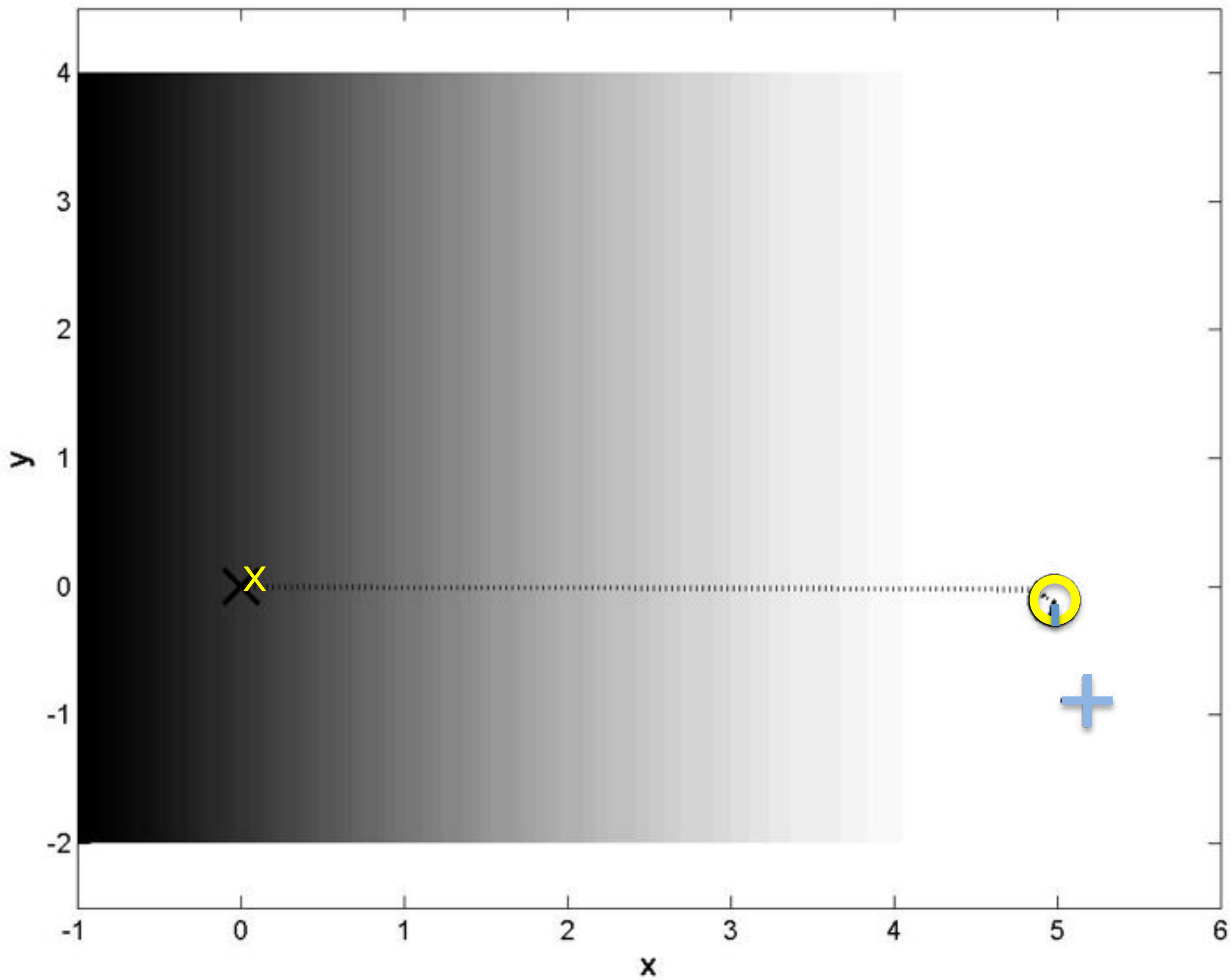
Replanning: light-dark problem



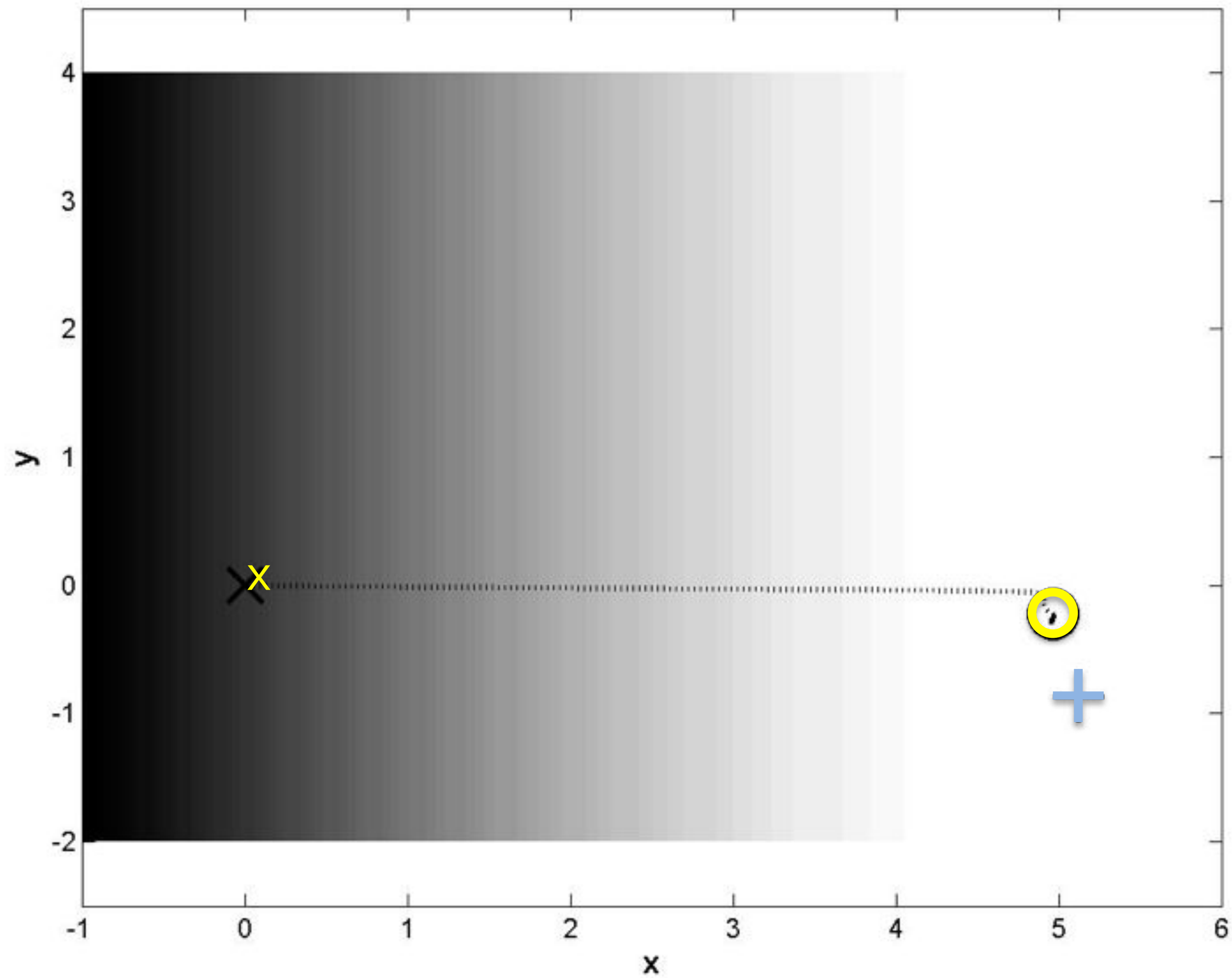
Replanning: light-dark problem



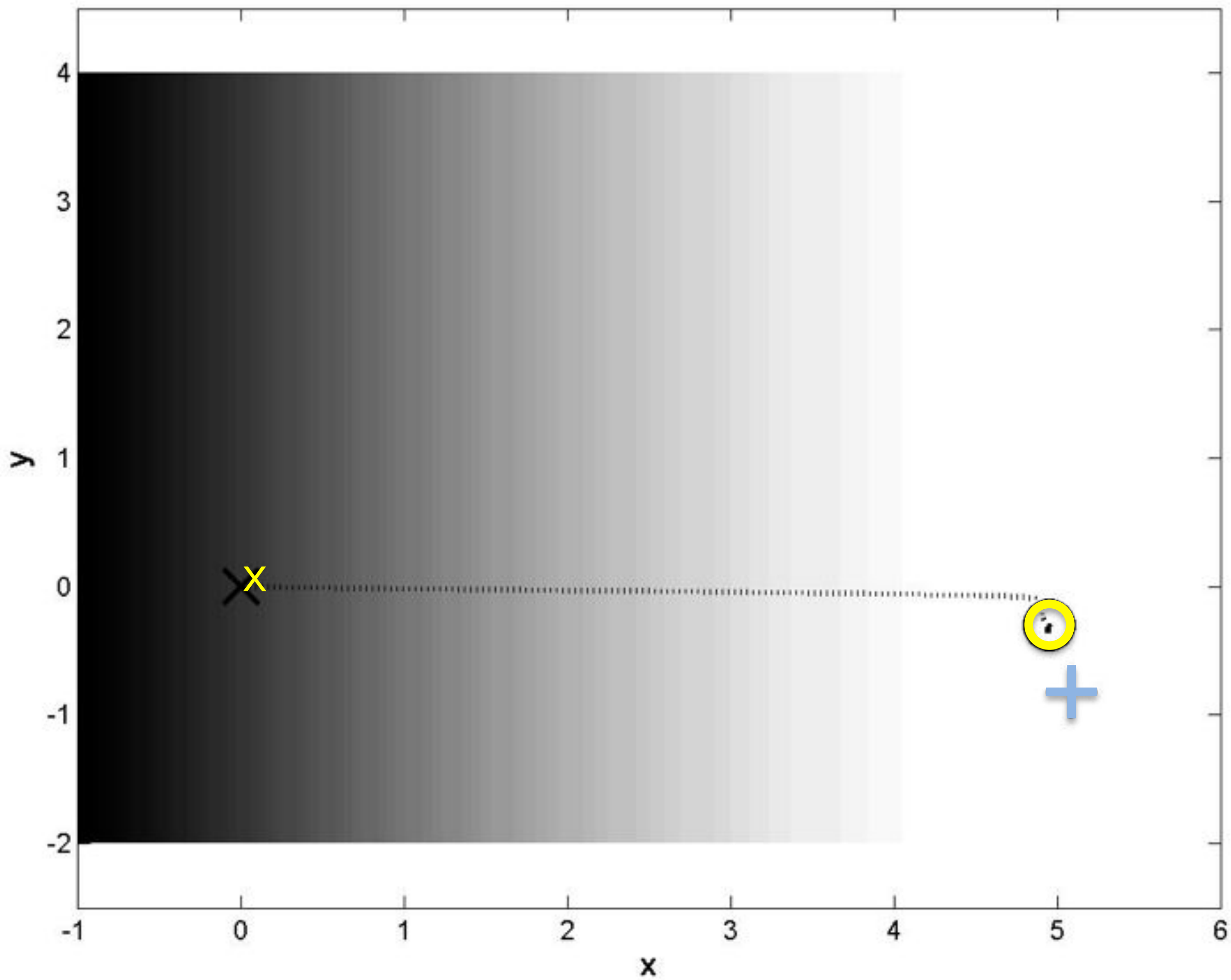
Replanning: light-dark problem



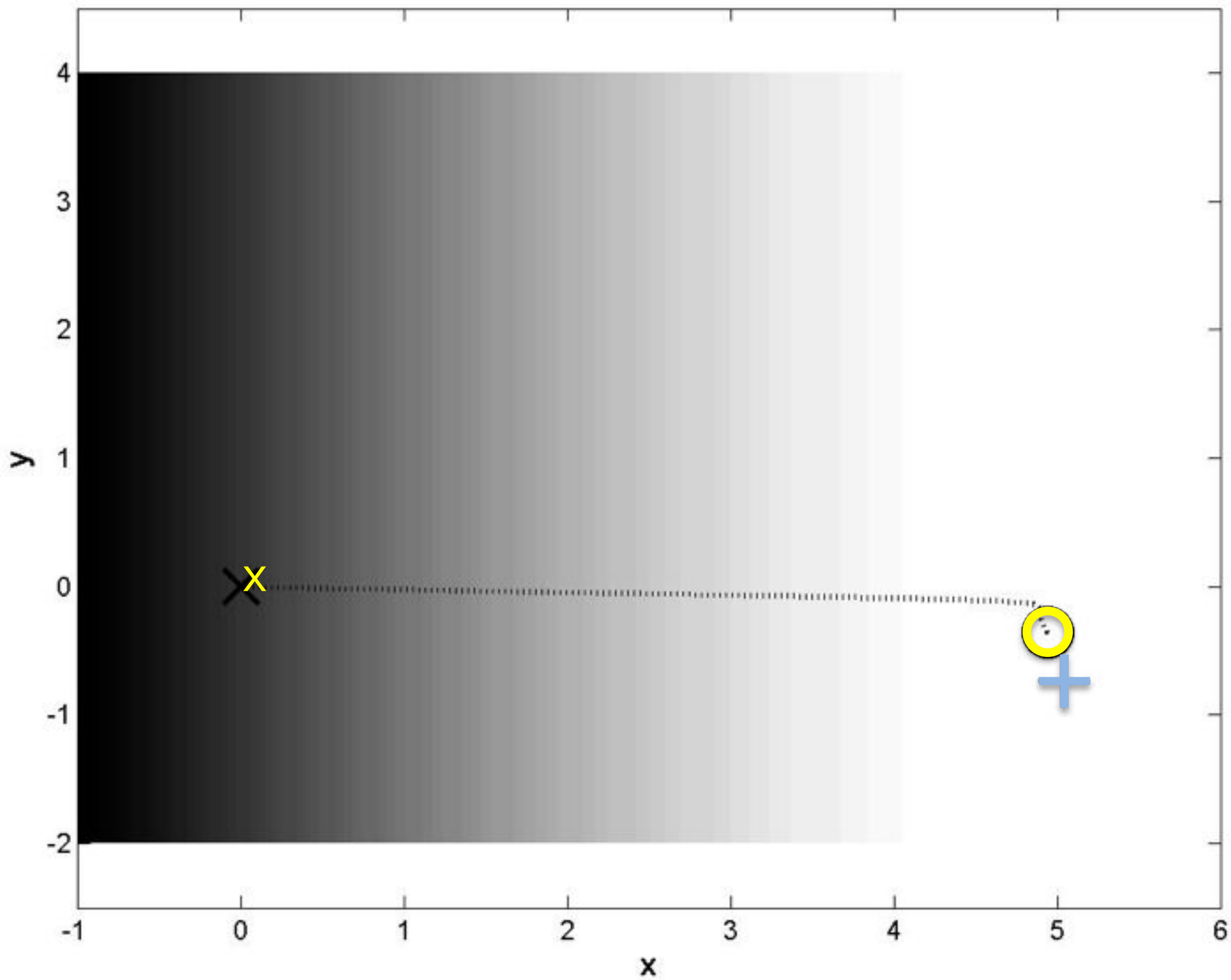
Replanning: light-dark problem



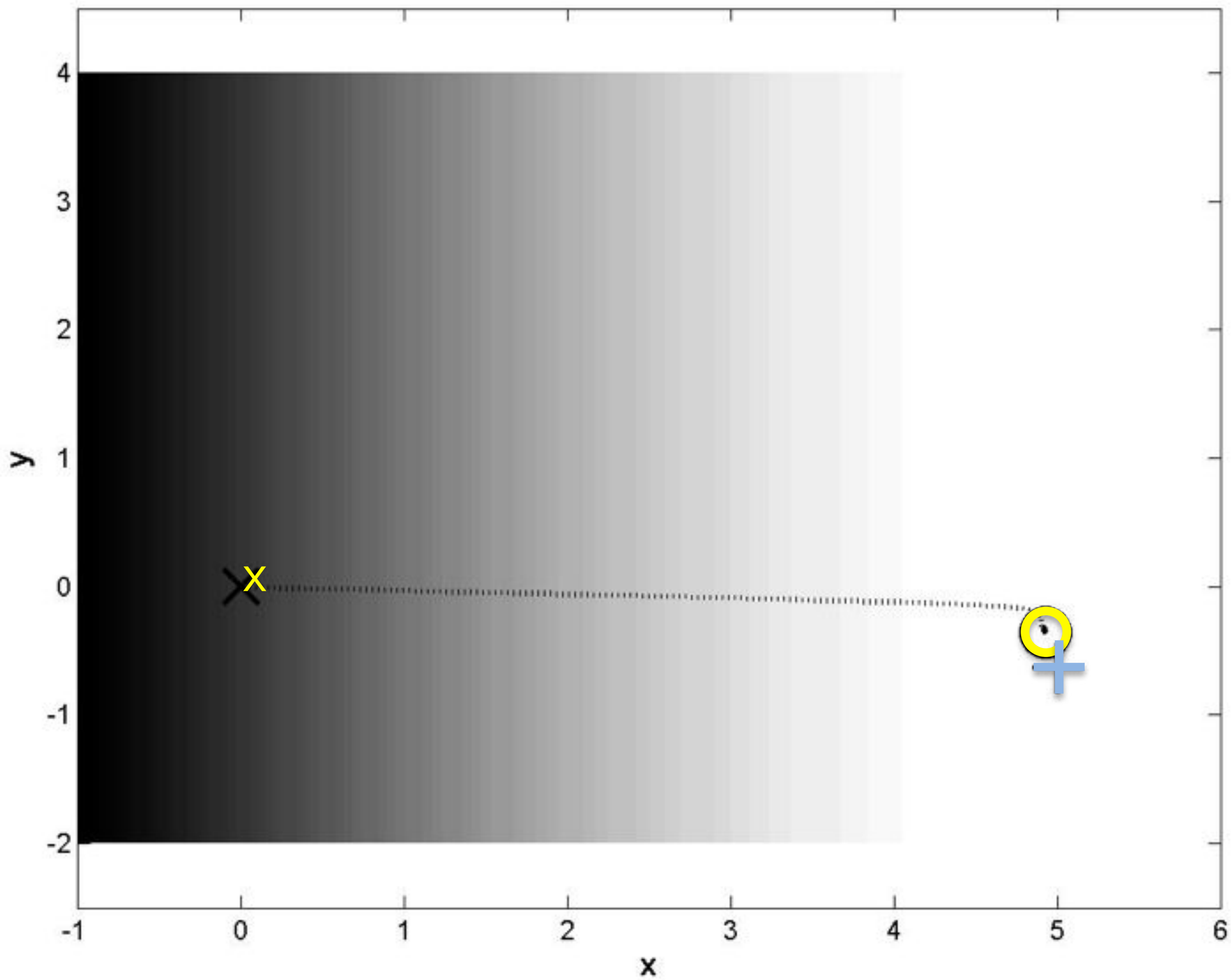
Replanning: light-dark problem



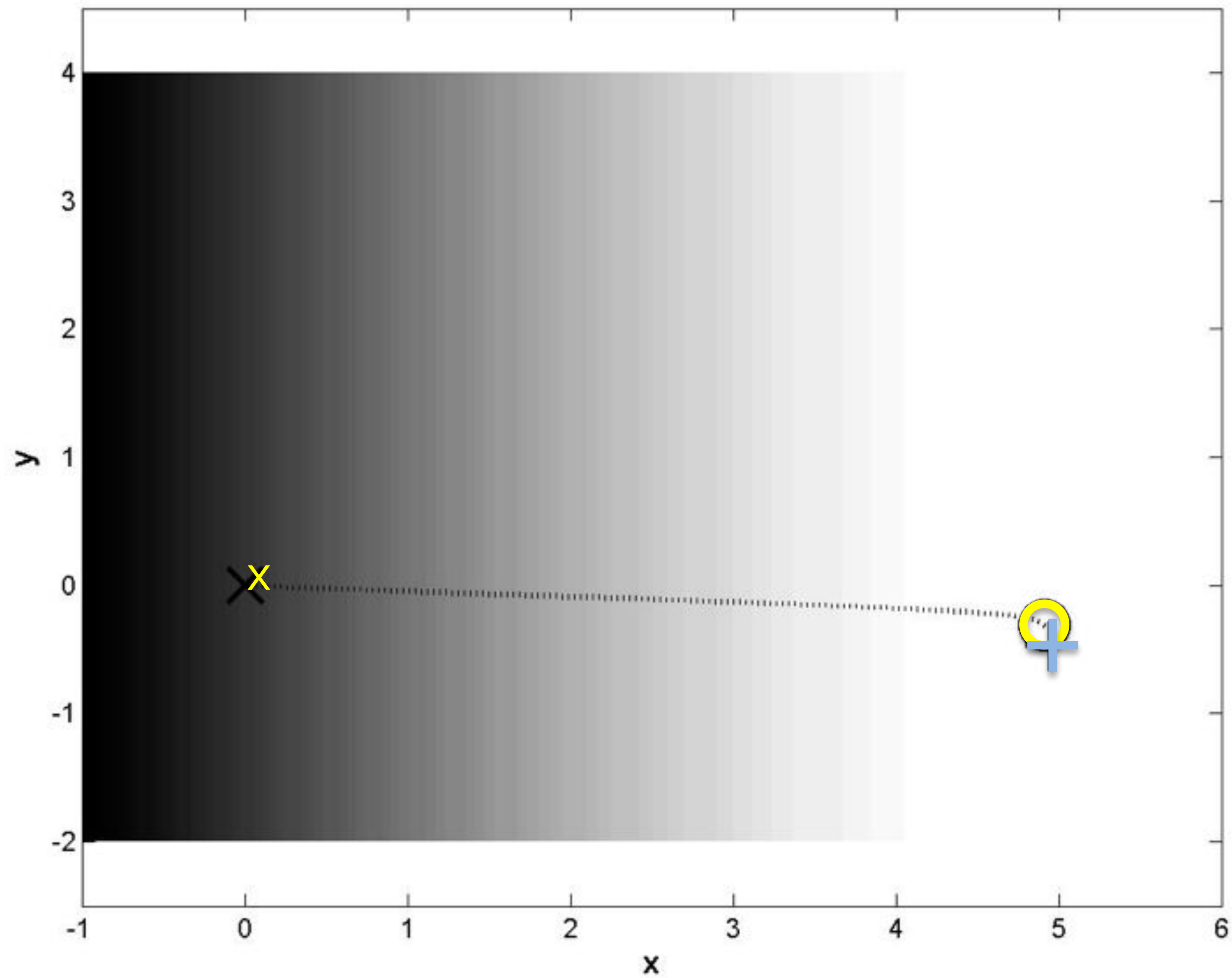
Replanning: light-dark problem



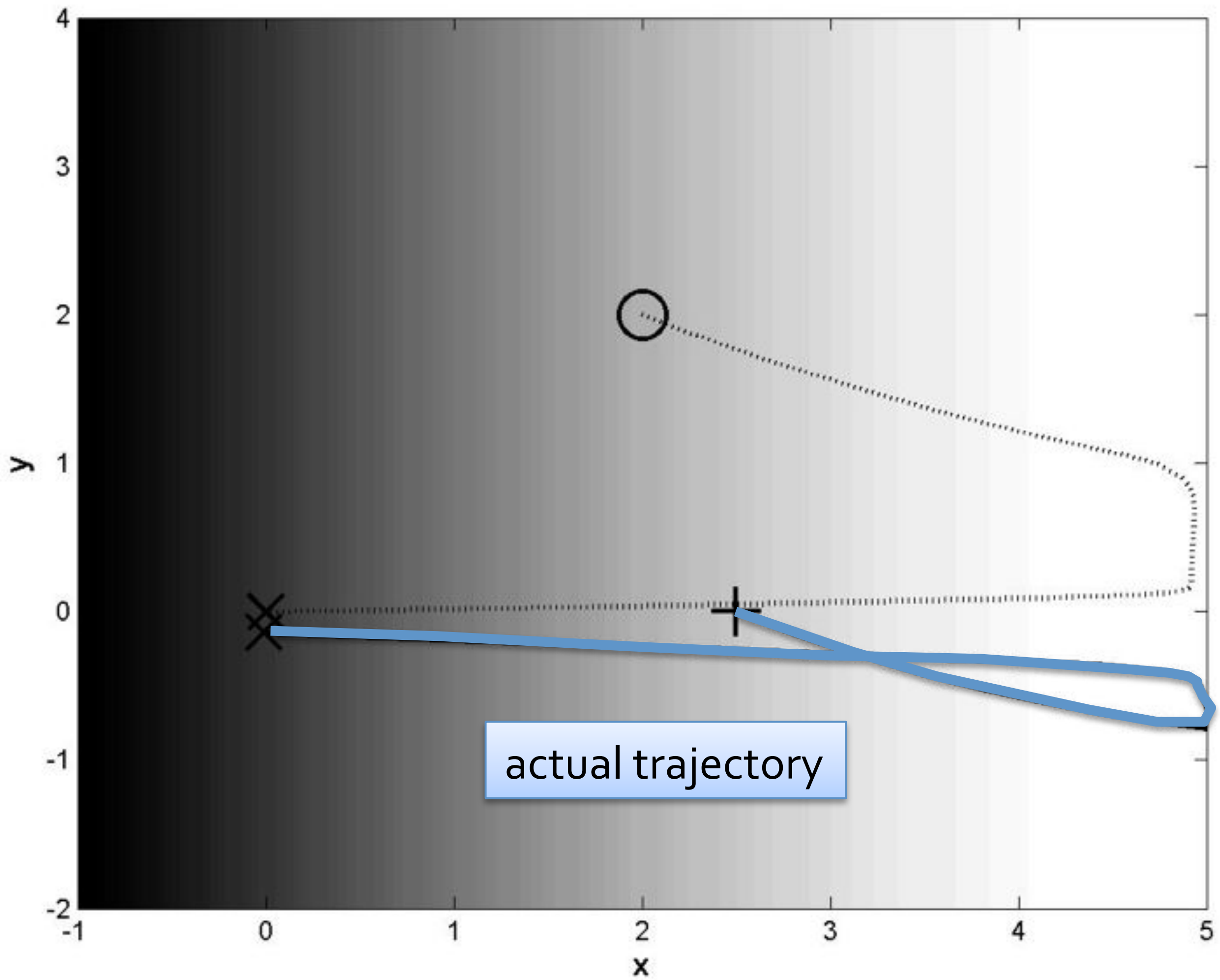
Replanning: light-dark problem



Replanning: light-dark problem



Replanning: light-dark problem



Optimistic (re)planning in belief space

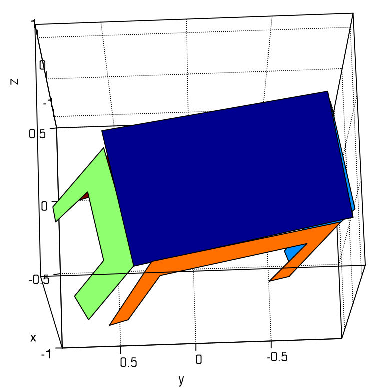
- **control with state-dependent observation noise**: continuous state, action, observation spaces

- **robot grasping with tactile sensing**: continuous state, action, observation spaces

- **household robot with local observation**: mixed continuous and relational spaces

Goal: pick up object of known shape with specific grasp

Visual localization and detection works moderately well…



Joint work with Kaijen Hsiao and Tomás Lozano-Pérez

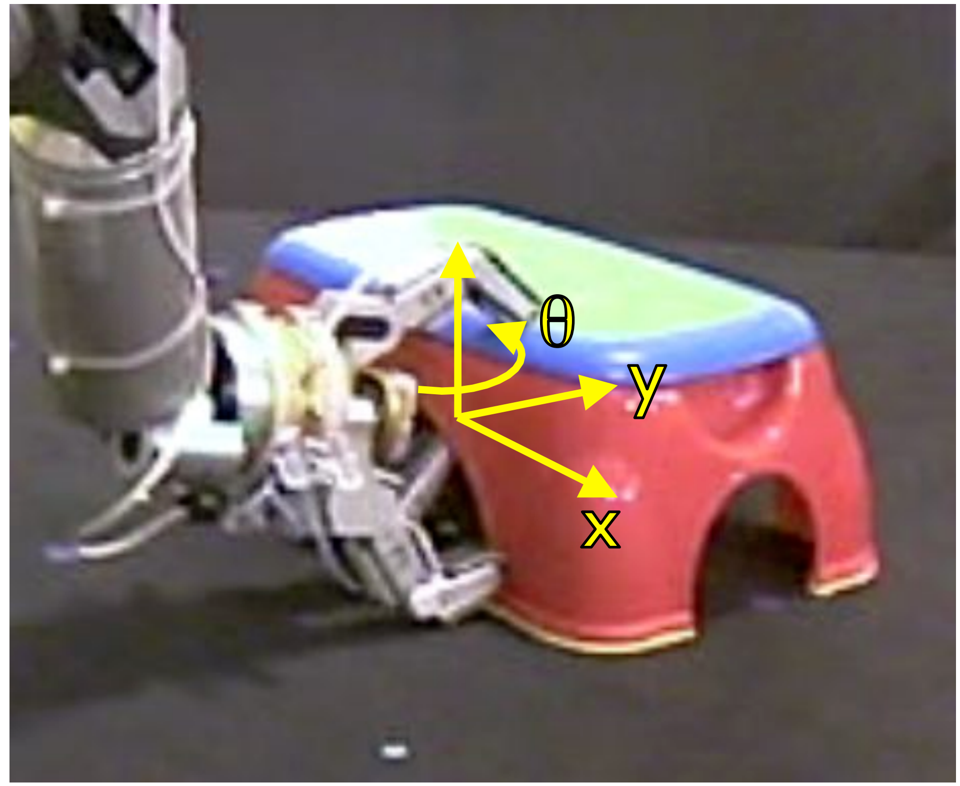
Hypothesis space

Robot pose:

- 11 DOF
- model as fully observable

Object pose:

- 3 DOF
- model as partially observable



State estimate: probability distribution over object pose

Macro actions

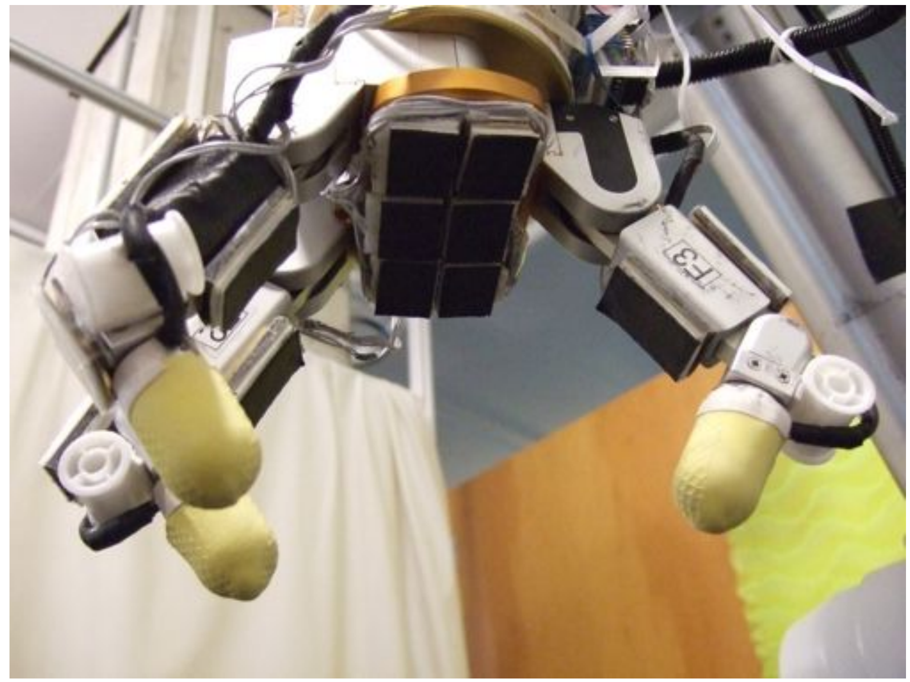
Execute a trajectory:

- stop moving arm if any contact is felt
- close each finger until it makes contact

Fixed set of parameterized trajectories, always executed with respect to most likely state

Observations

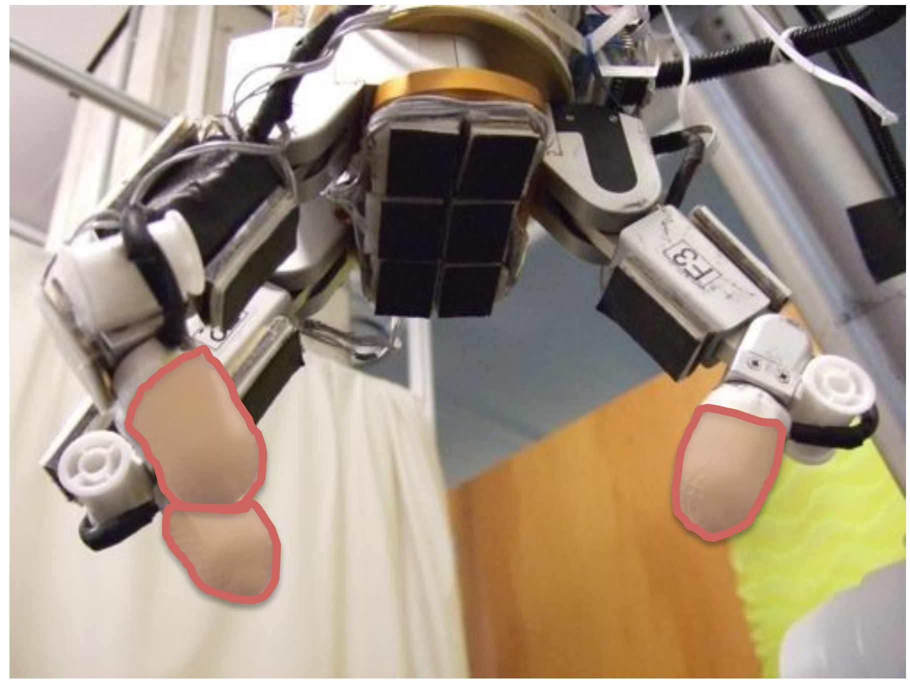
- Arm trajectory according to proprioception



Observations

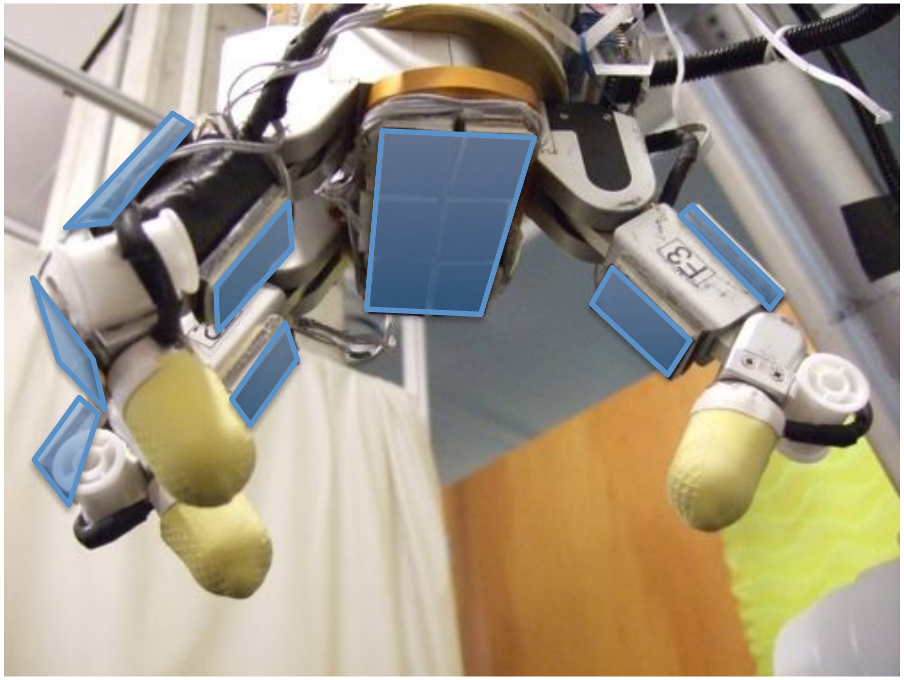
- Arm trajectory according to proprioception

- 6-axis force-torque sensors on fingertips



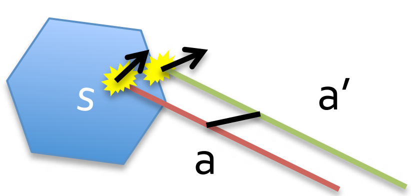
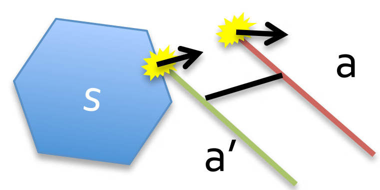
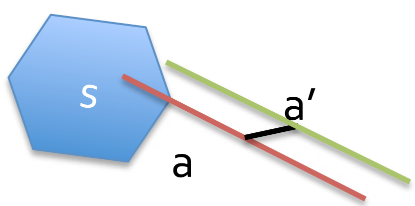
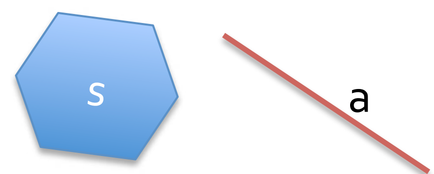
Observations

- Arm trajectory according to proprioception
- 6-axis force-torque sensors on fingertips
- Binary contact sensors



Observation model: $\Pr(o \mid s, a)$

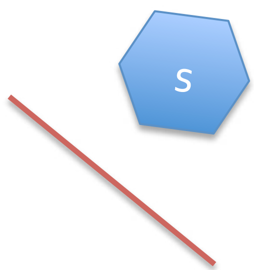
Nominal observation for s, a: o*

Actual o	Contact	no contact
Contact	Gaussian density on dist to closest a’ that would not have caused interpenetration X Gaussian density on dist between contact positions and normals 	Gaussian density on dist to closest a’ that would have caused contact X Gaussian density on dist between contact positions and normals 
no contact	Gaussian density on dist to closest a’ that would not have caused contact 	Max value of Gaussian density used for nominal contact case 

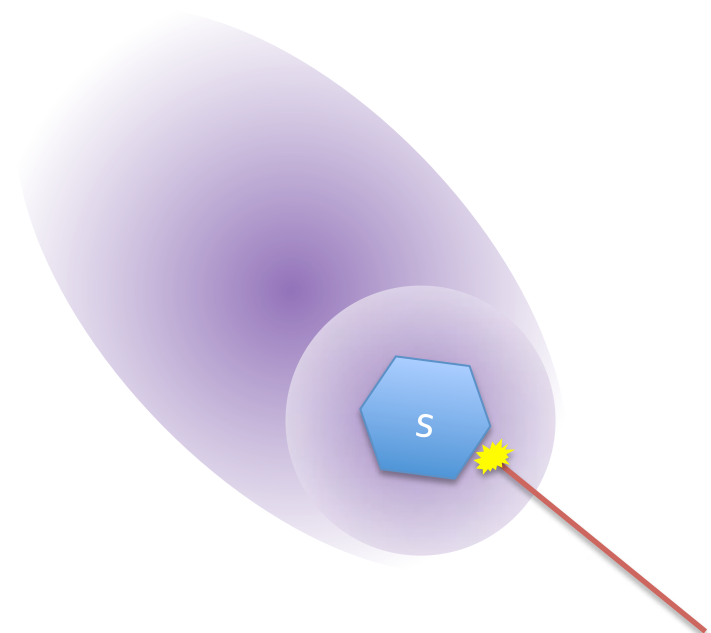
Leslie Pack Kaelbling, AAAI2010

Transition model: $\Pr(s_{t+1} \mid s_t, a_t)$

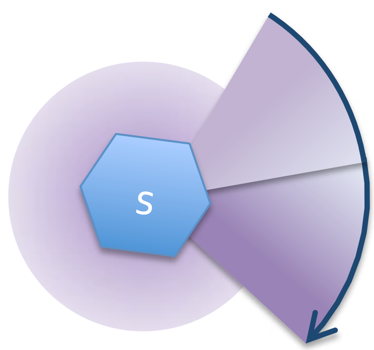
No contact: no change



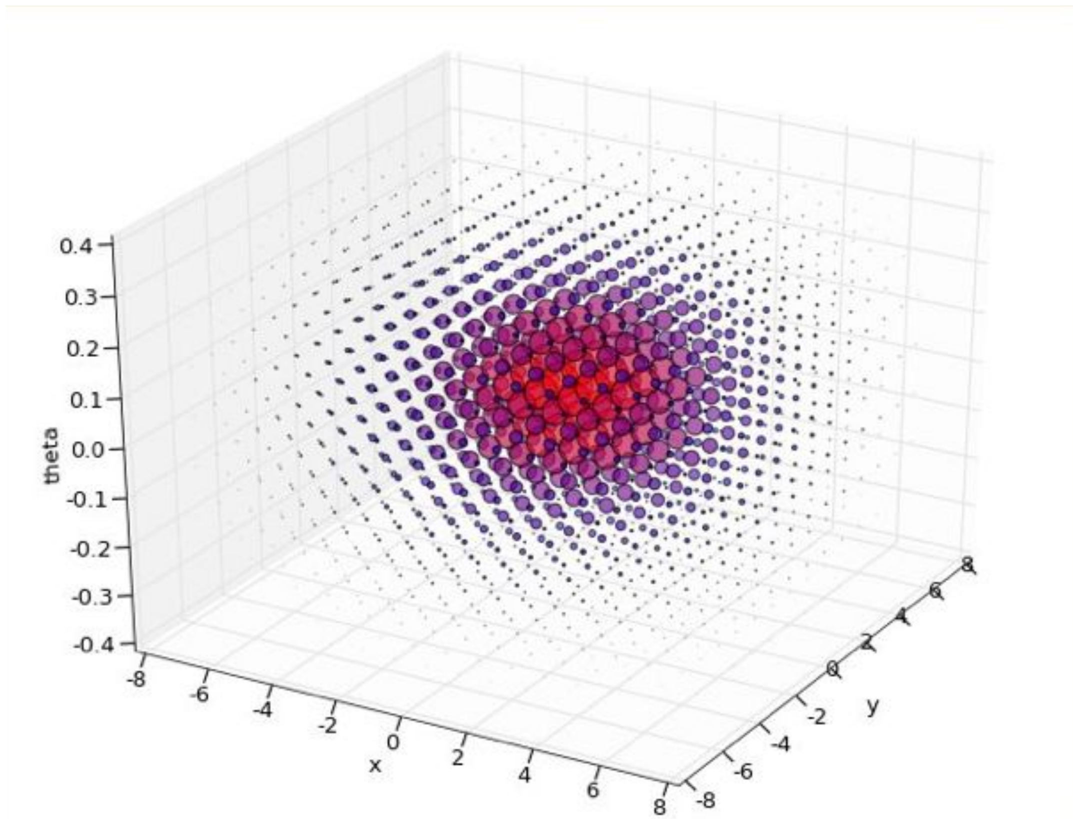
Contact: probability of being bumped depends on observation



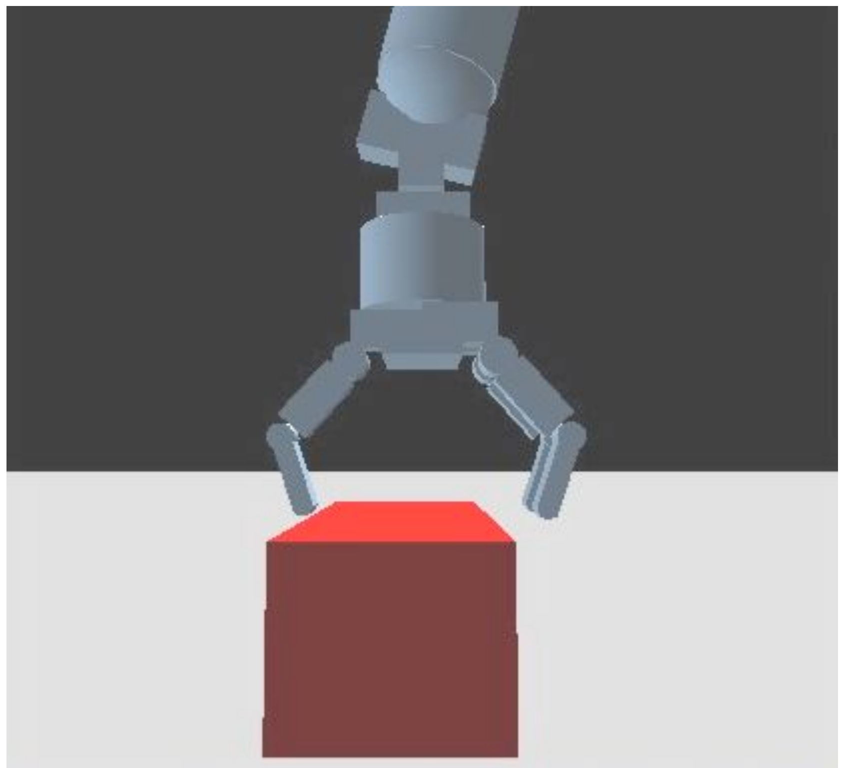
Reorientation: similar to contact with large rotational variances



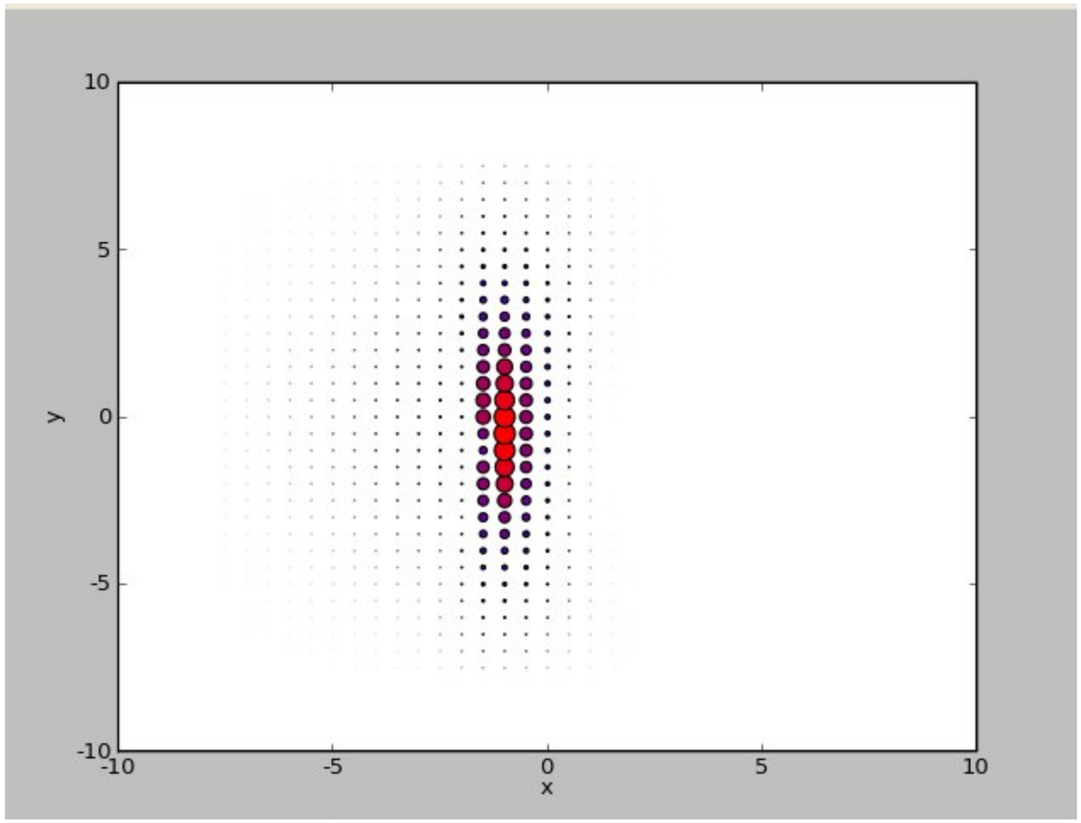
Initial belief state



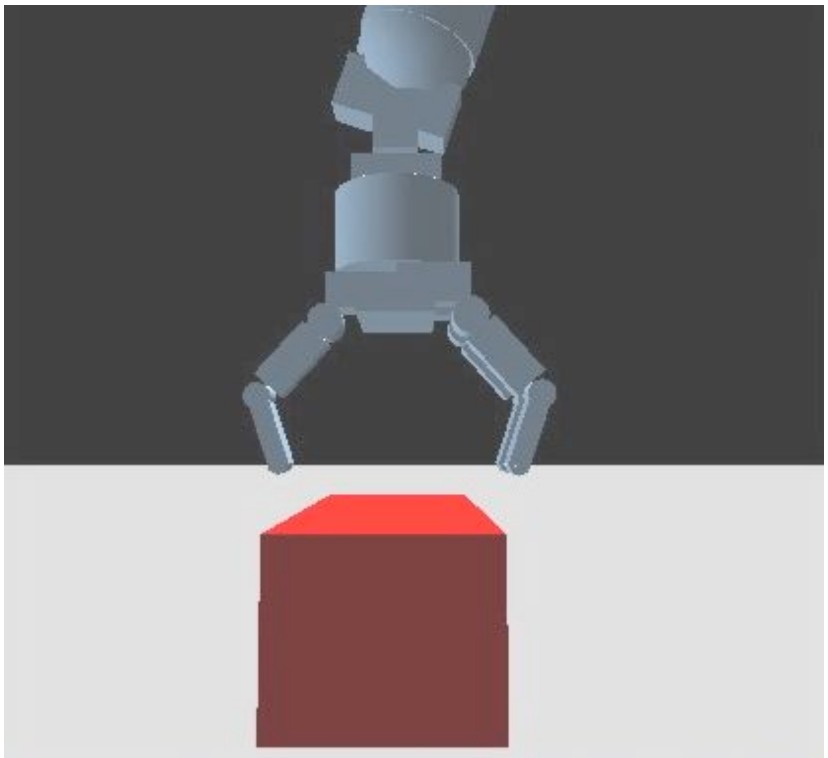
Tried to move down — finger hit corner

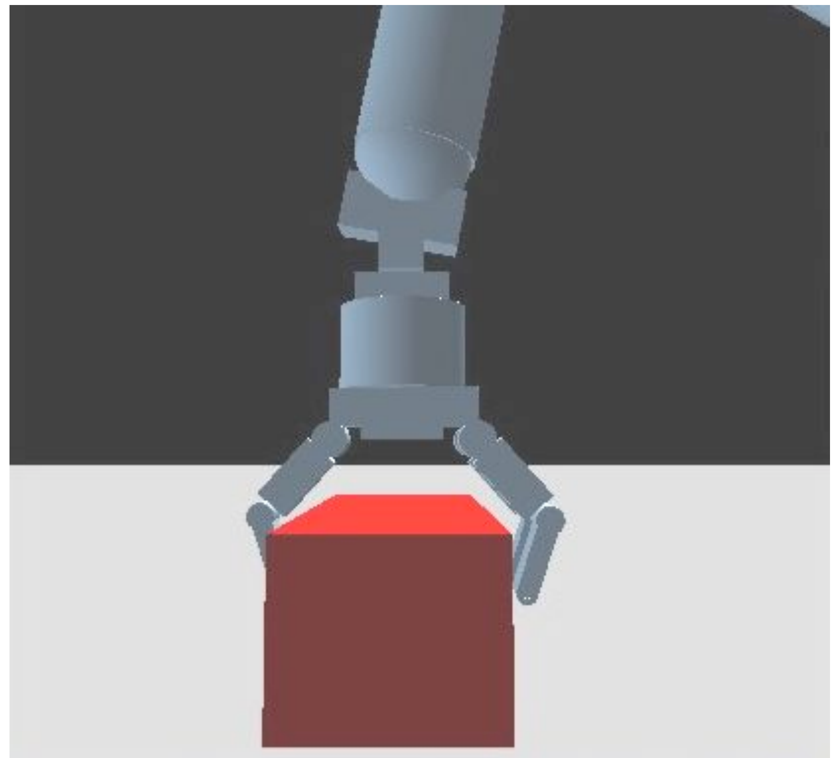


Updated belief



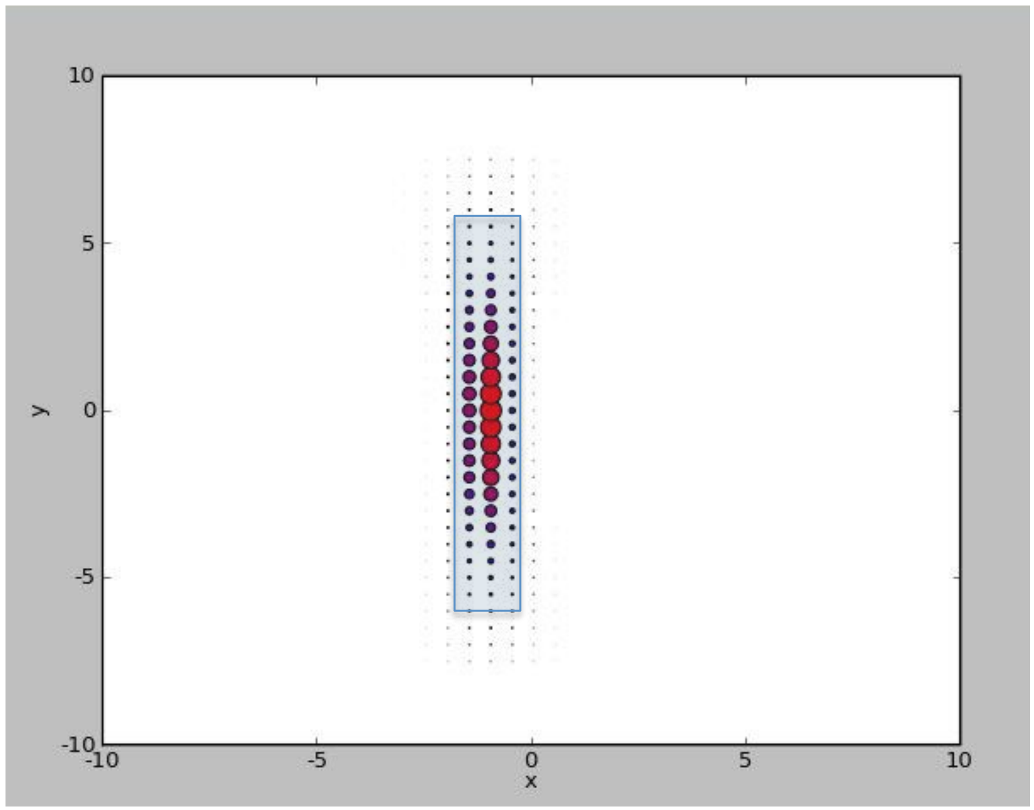
Another grasp attempt



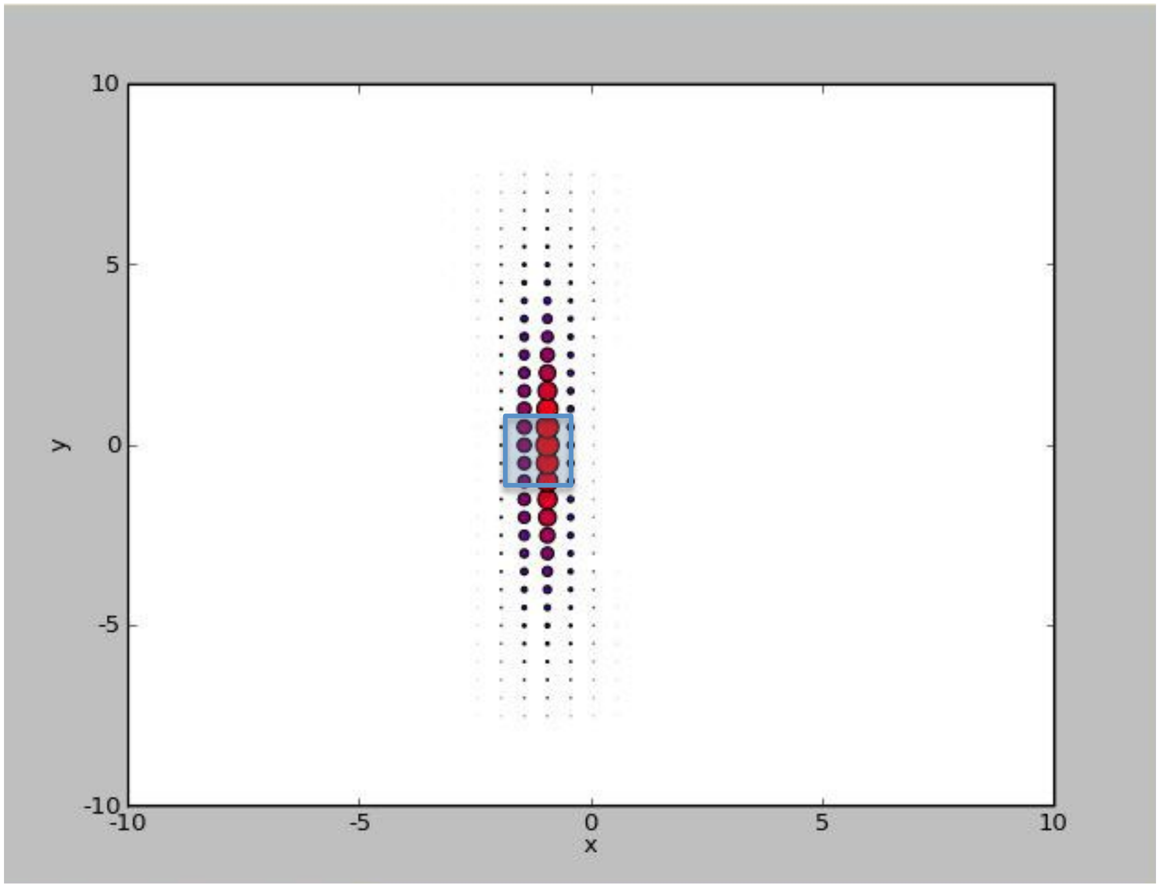


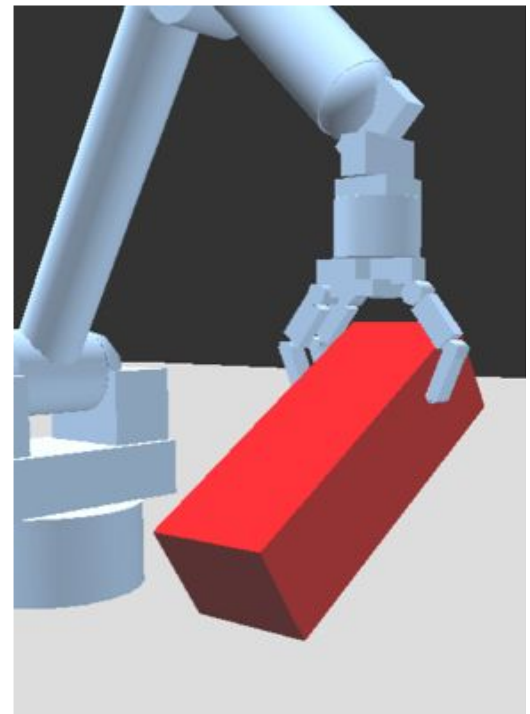
Goals in belief space

- Specify set of desirable ranges in X, Y, Θ
- Satisfied if probability that the pose is in that set is high



What if Y coordinate of grasp matters?





Action selection

How to select among the actions?

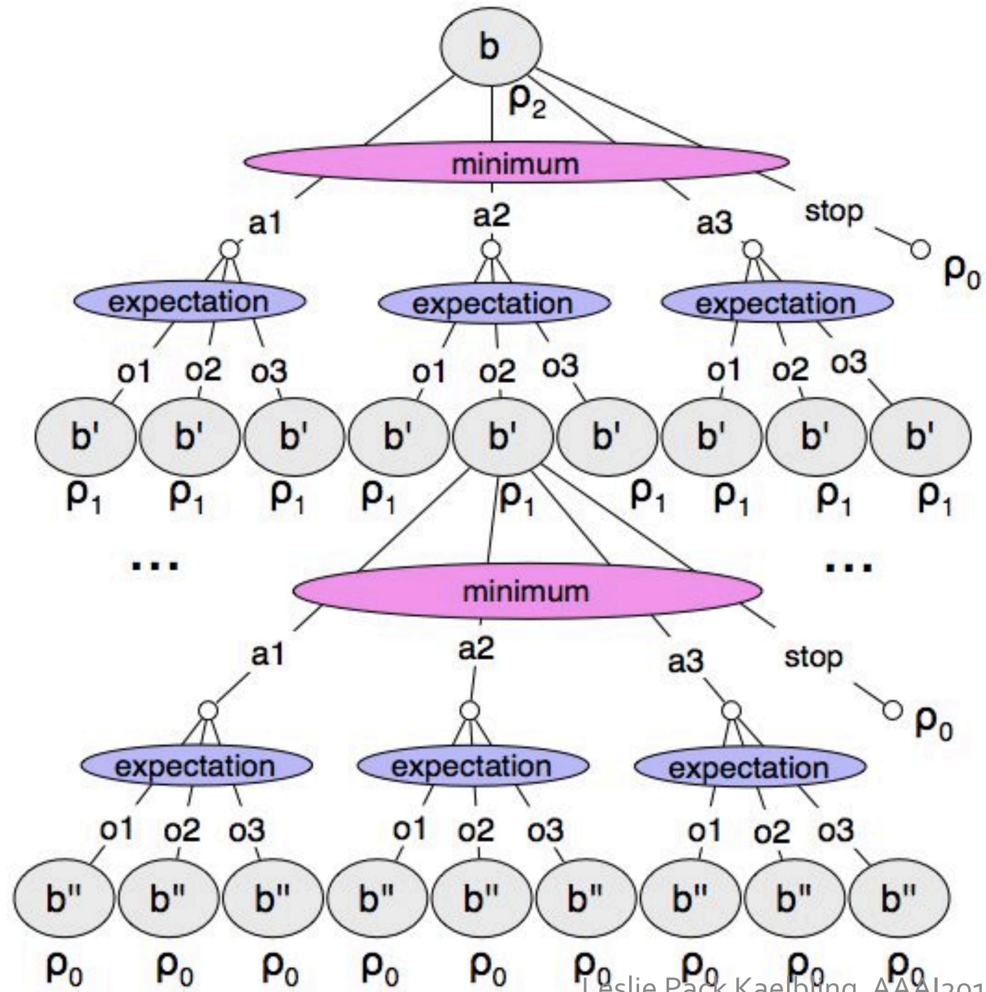
- Until probability of failure given belief is < eps
  - Select WRT by searching forward from belief
  - Execute WRT, and get observations o
  - Update belief

WRTs include:

- target grasp
- information-gain trajectories
- re-orientation

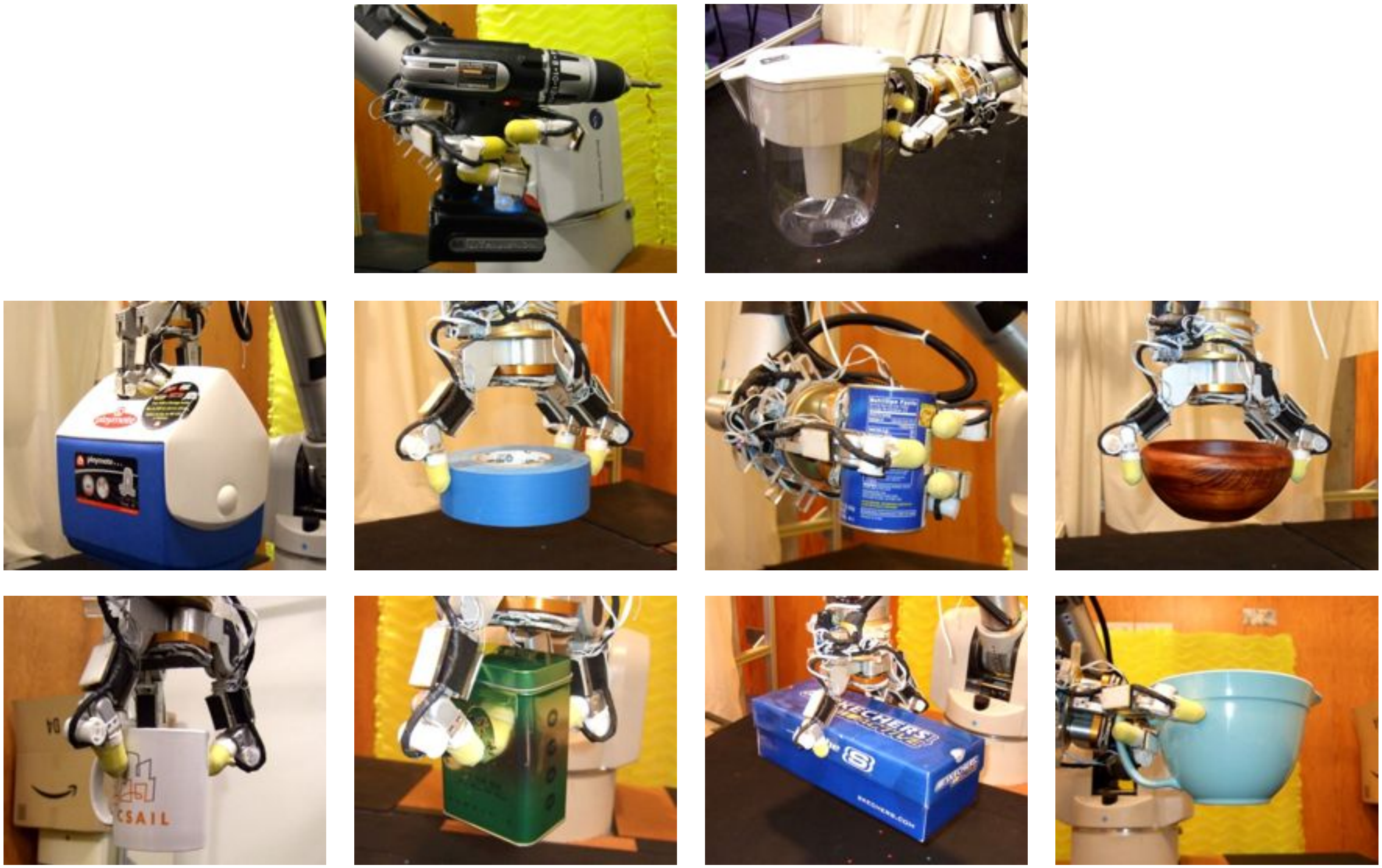
Forward search

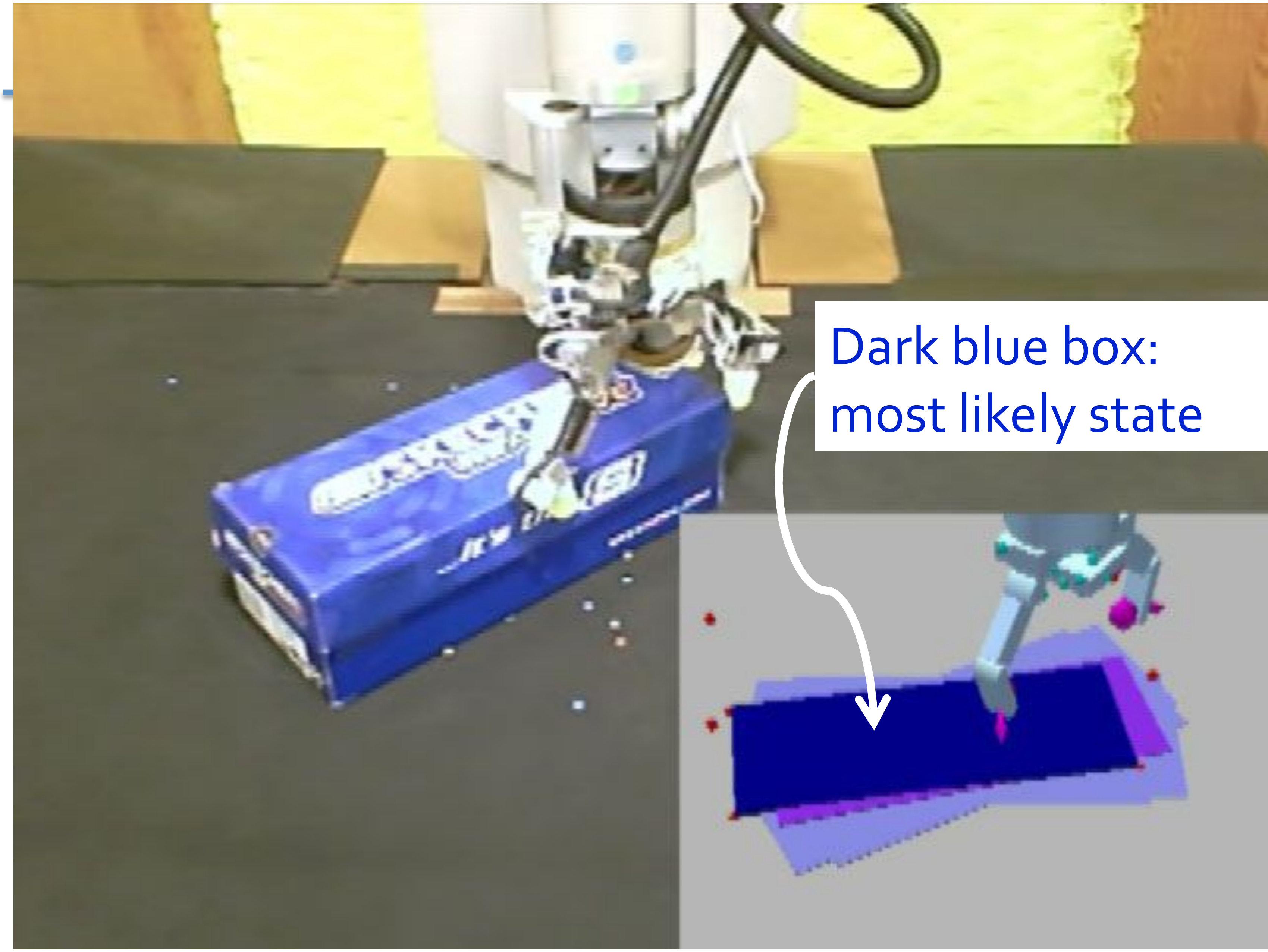
- Compute k-step risk using backward induction
- Prune and cluster to decrease observation branching
- Depth 2 sufficed in our problems
- Risk at leaves is likelihood of failure of target grasp

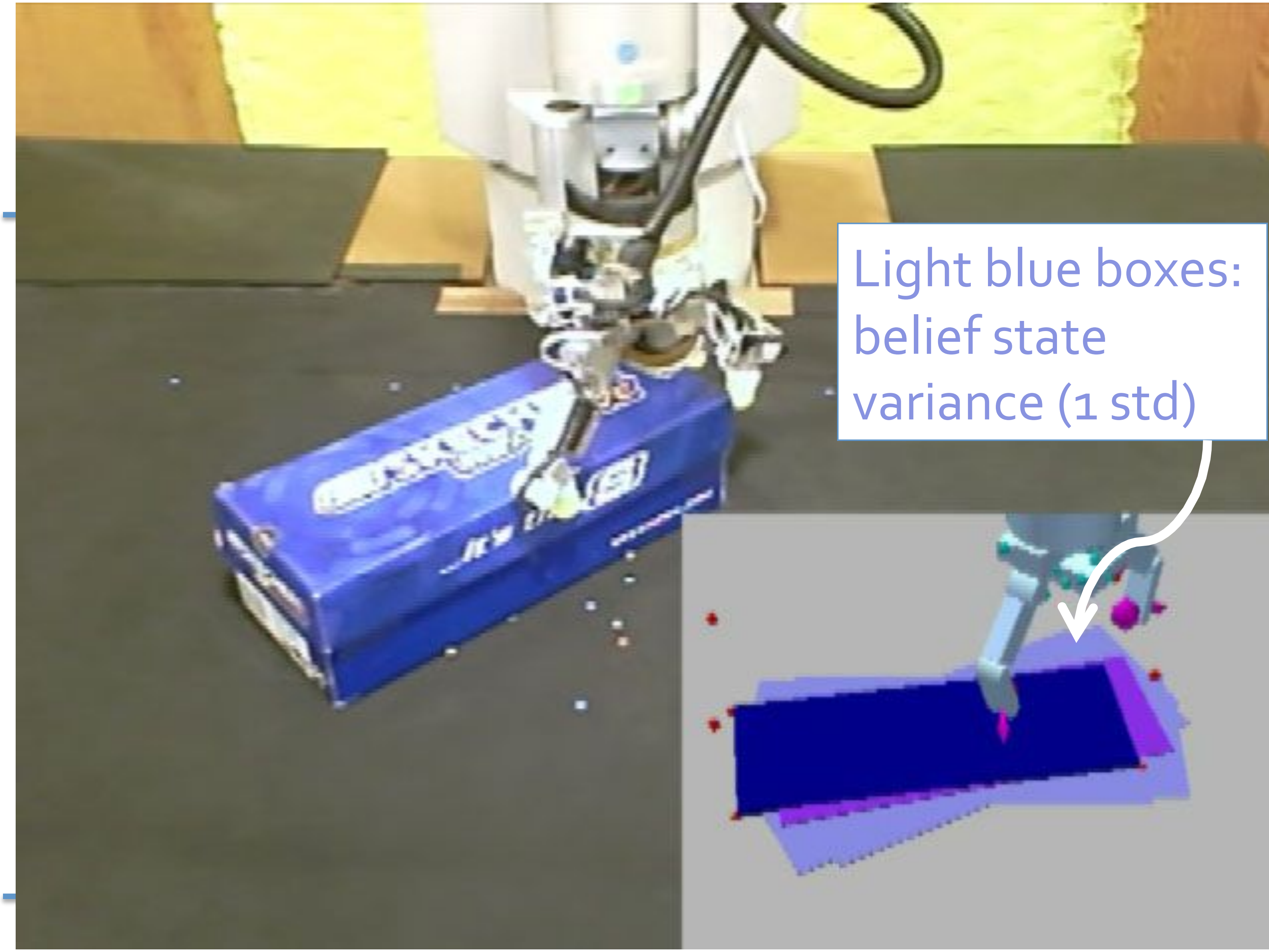


Leslie Pack Kaelbling, AAAI2010

Objects and desired grasps




Dark blue box:
most likely state


Light blue boxes: belief state variance (1 std)

Brita results: 10 / 10 successful grasps

Powerdrill: 10 / 10 successful grasps

25x
speed

Optimistic (re)planning in belief space

- **control with state-dependent observation noise**:
  continuous state, action, observation spaces

- **robot grasping with tactile sensing**:
  continuous state, action, observation spaces

- **household robot with local observation**:
  mixed continuous and relational spaces

Classes of robotics problems in which:

- Problems are huge:
  - long horizon
  - many continuous dimensions
  - combinatoric discrete aspects
- No terrible outcomes
- Geometry is not intricate
- Partial observability: local but fairly reliable

Joint work with Tomás Lozano-Pérez

Symbols to Angles

Initial state known in geometric detail

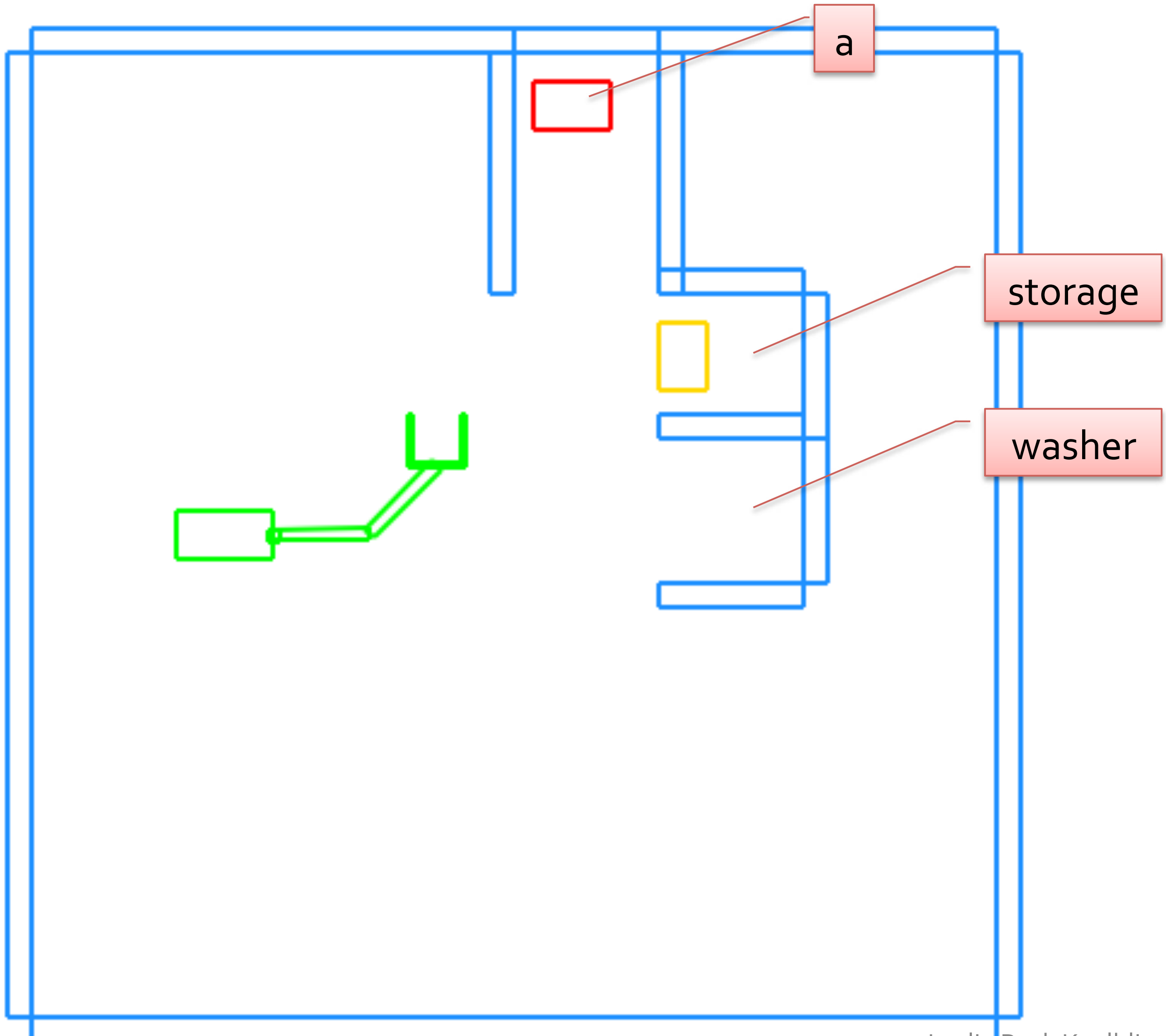
Goal set is abstract, symbolic

tidy(house) ^ charged(robot)

Operator descriptions:

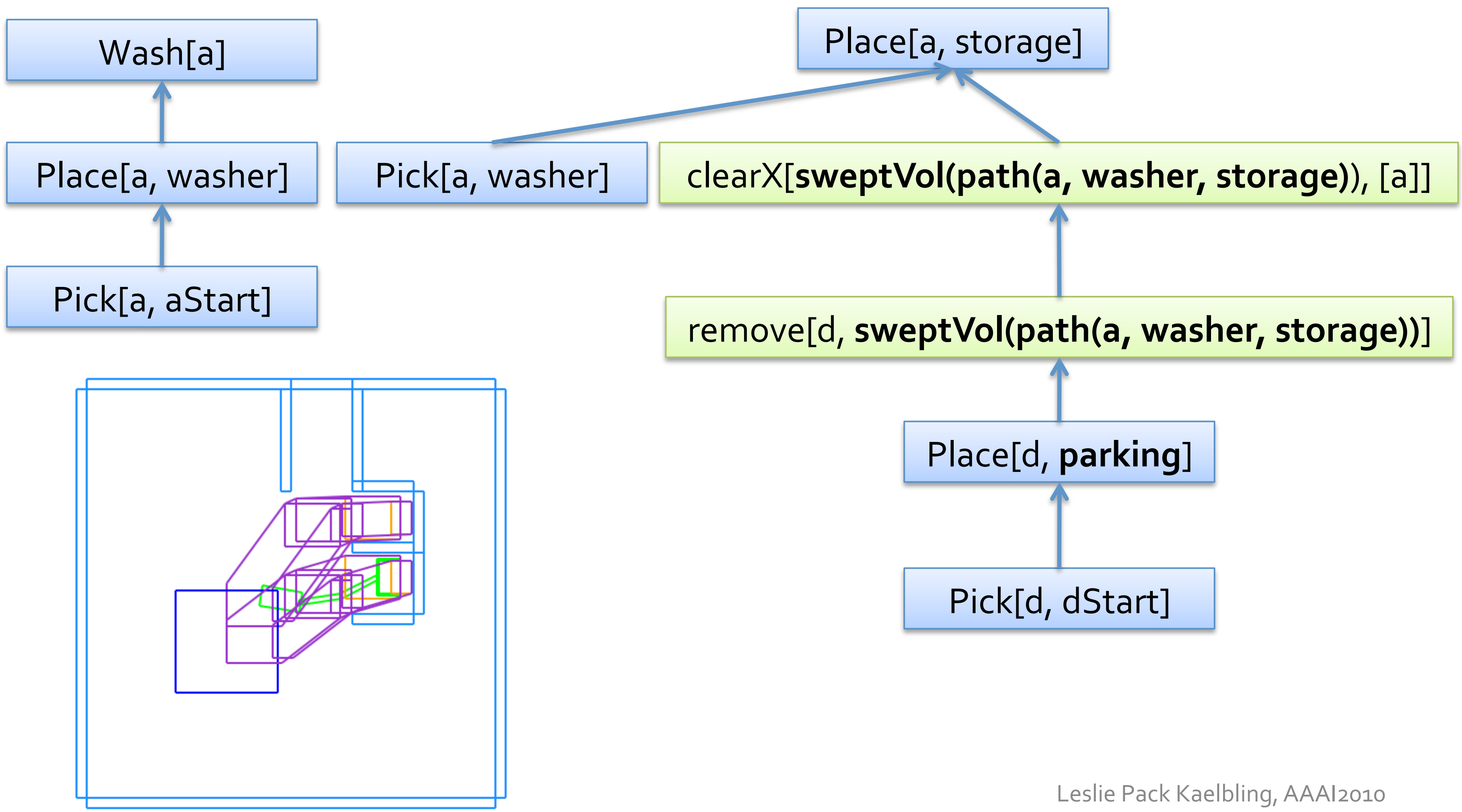
- STRIPS-like, with continuous values
- procedures suggest values for existential vars
- geometric reasoning

Wash a block and put it away

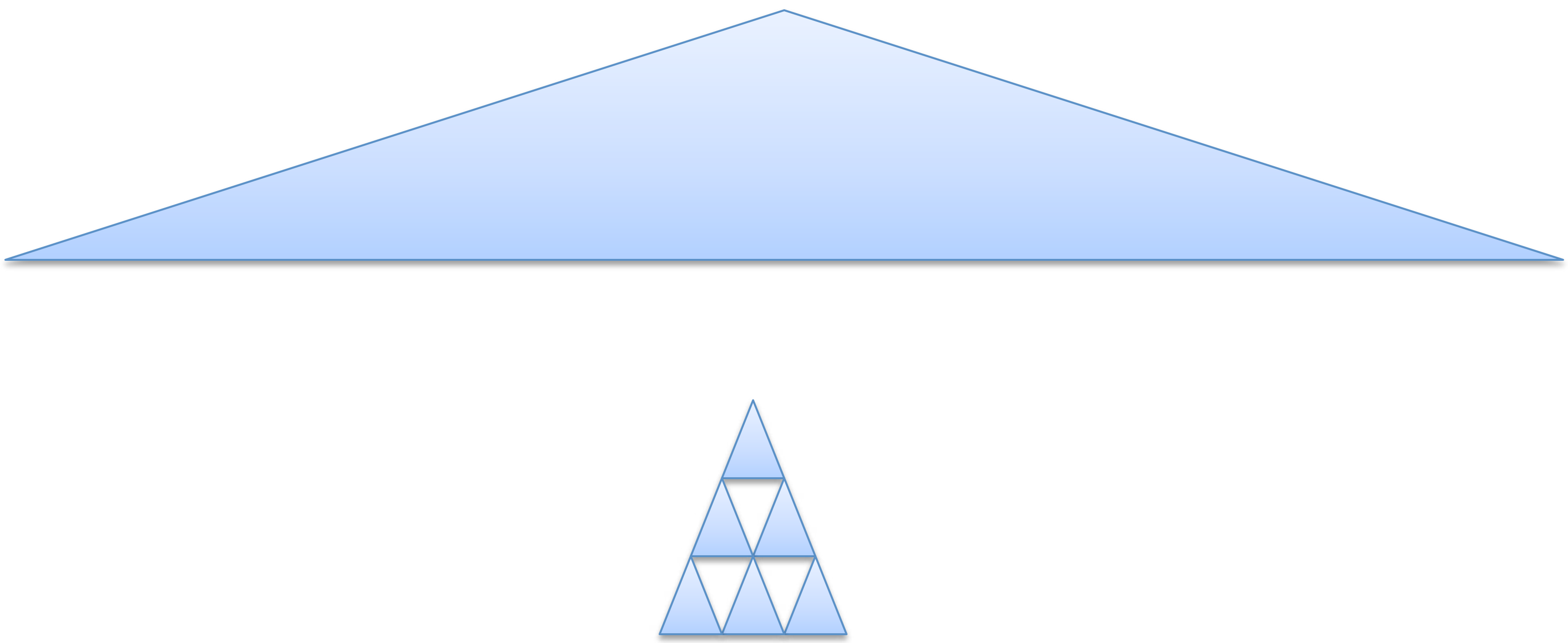


Clean(a) and In(a, storage): Regression structure

7 primitive steps; 3000 search nodes



Hierarchy crucial for large problems

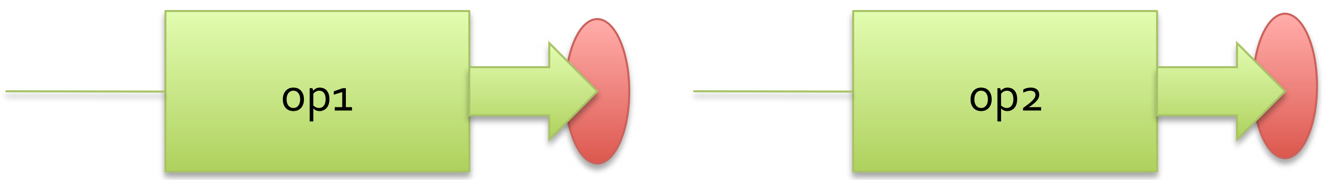


Subtrees represent **serialized subtasks**

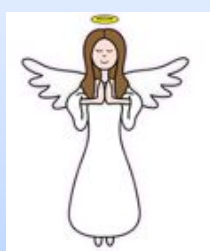
Hierarchical semantics

Subgoal is an abstract operator:

What does it mean to sequence two subgoals?



Depends on who gets to choose the outcome:

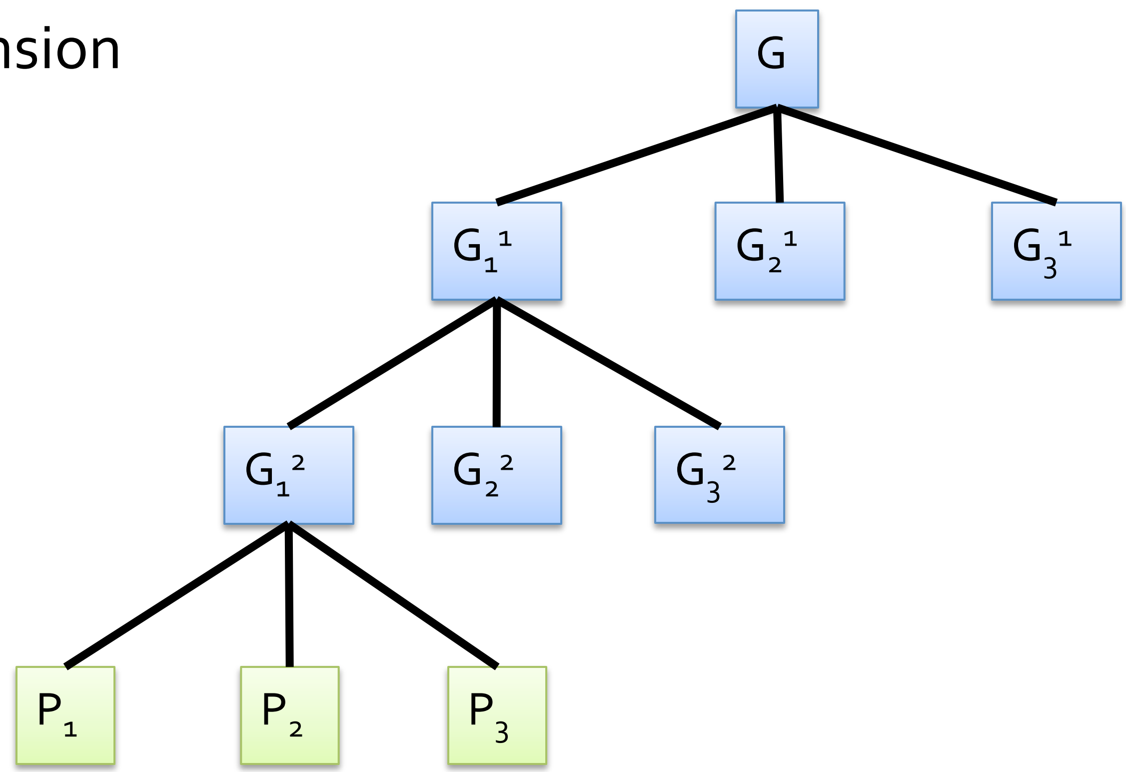

us


nature

Wolfe, Marthi, Russell

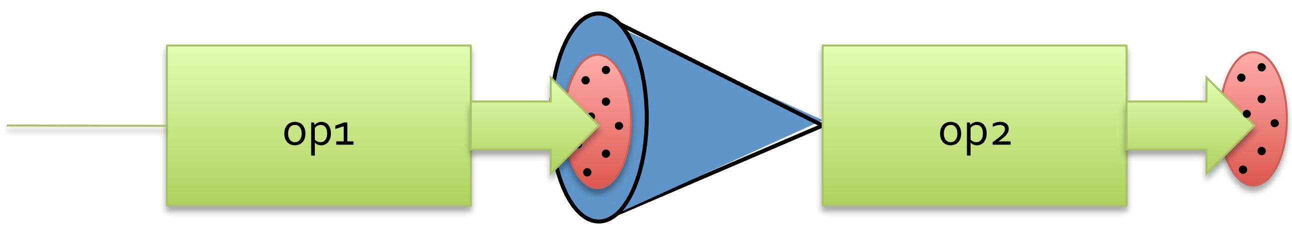
Planning in the now

- maintain left expansion of plan tree
- execute primitives
- plan as necessary

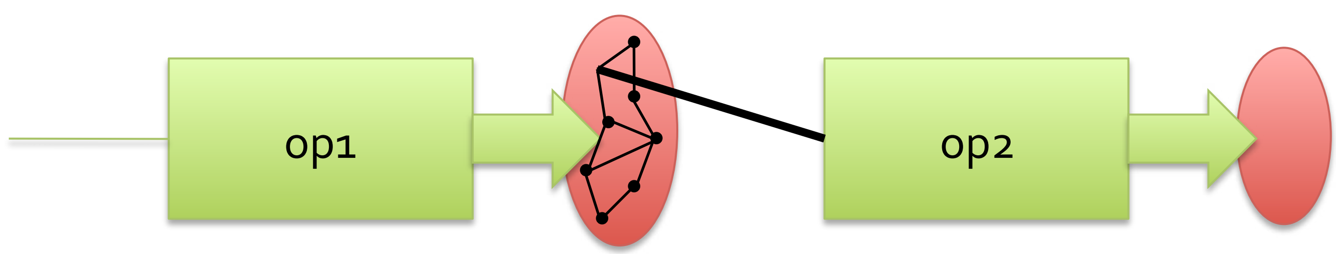


Satanic semantics

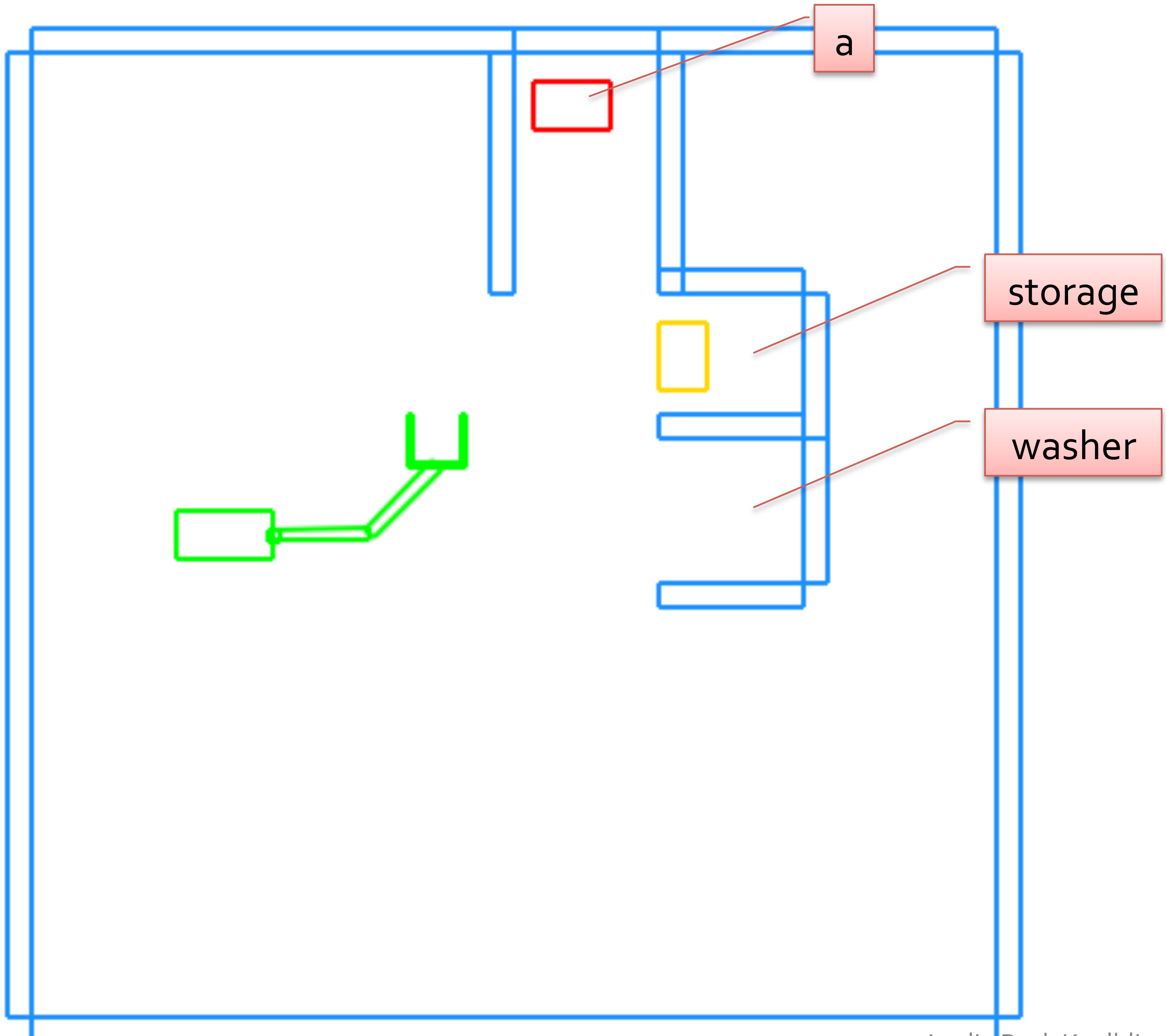
We have to handle any outcome the devil picks

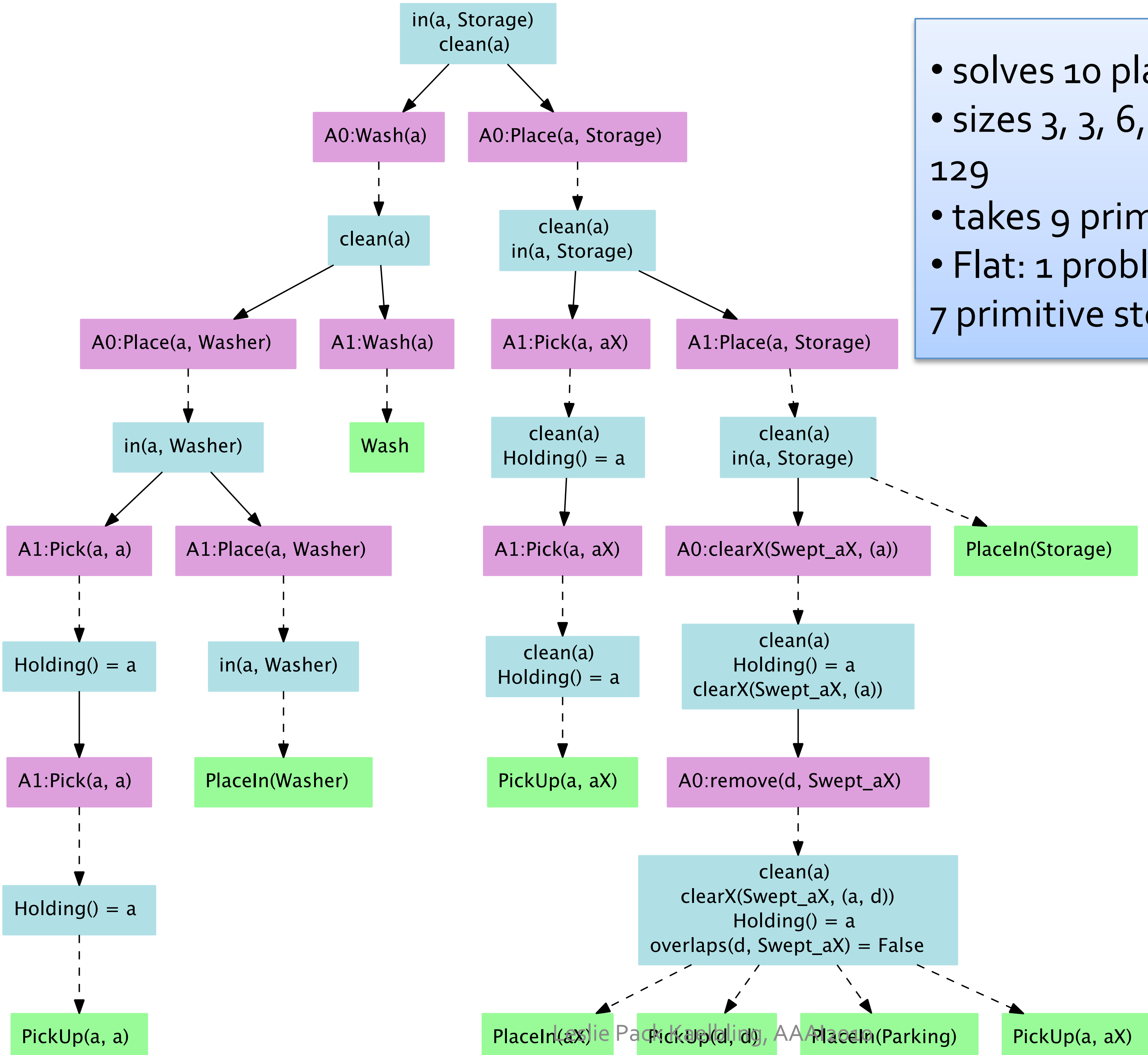


Okay if: Preconditions of op2 can be achieved from any state resulting from op1



Wash a block and put it away





- solves 10 planning problems
- sizes 3, 3, 6, 5, 2, 7, 6, 8, 13, 129
- takes 9 primitive steps
- Flat: 1 problem, 3000 nodes, 7 primitive steps

Planning in the Know

Plan in the **now in belief space**:

- Make a single plan that will succeed with high probability
- Replan on unexpected observations

Plan at the "**knowledge level**"

Moore; Petrick,Bacchus

- Traditional to plan in the powerset of the state space
- We have infinite state space
- Use explicit logical representation of knowledge and lack of knowledge

Plan at **level of abstraction** supported by current belief state

Going on a tiger hunt

move(Room):

res: robotLoc = Room

listen:

pre: robotLoc = hall

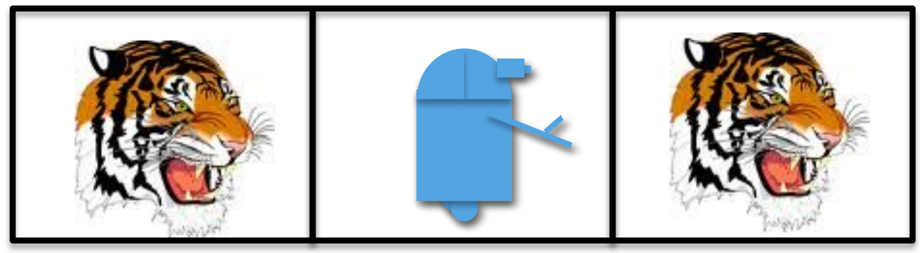
result: KV(tigerLoc)

shoot:

pre: robotLoc = tigerLoc

result: deadTiger

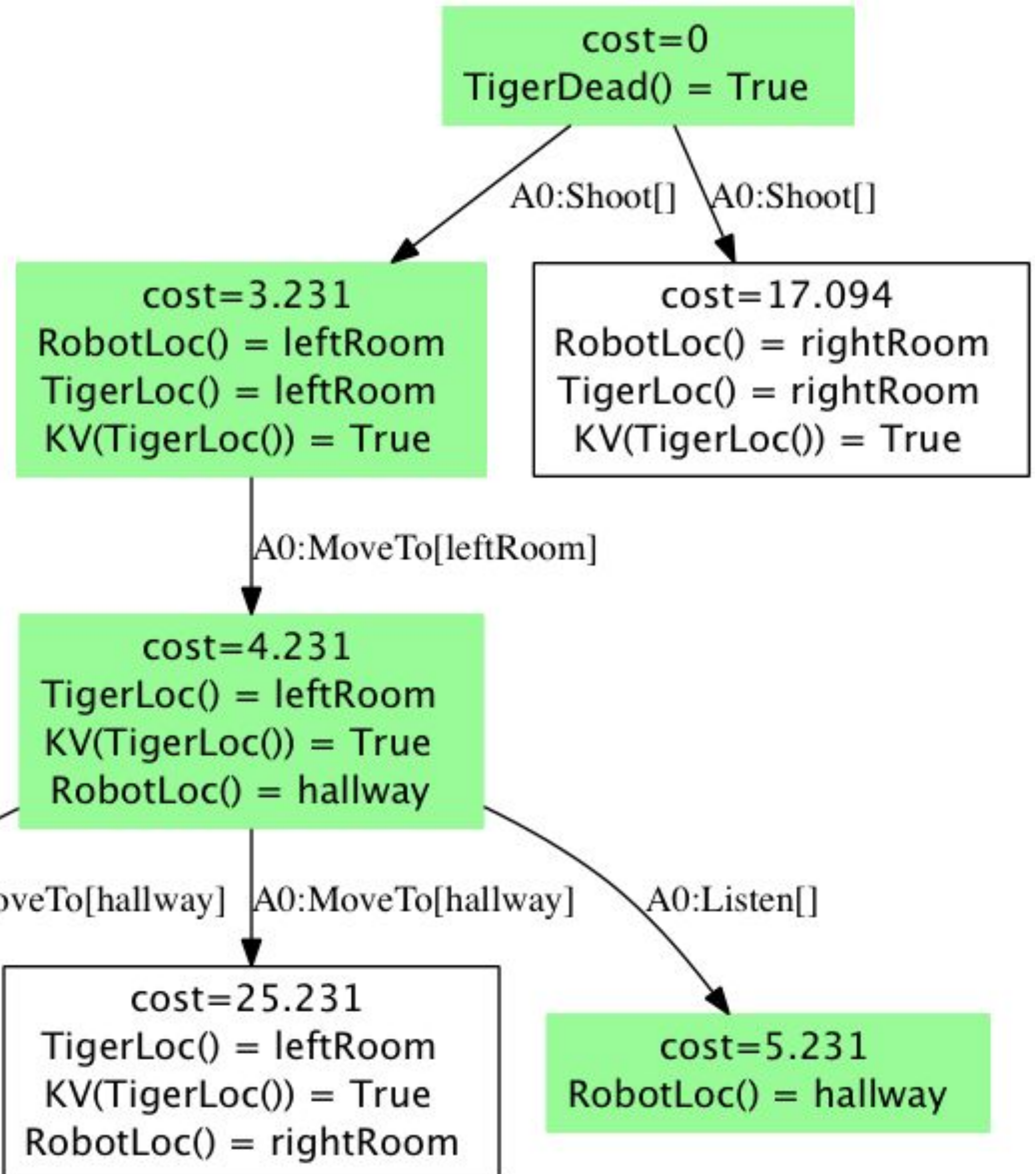
P(tigerLoc = leftRoom) = 0.8



Going on a tiger hunt: regression search tree

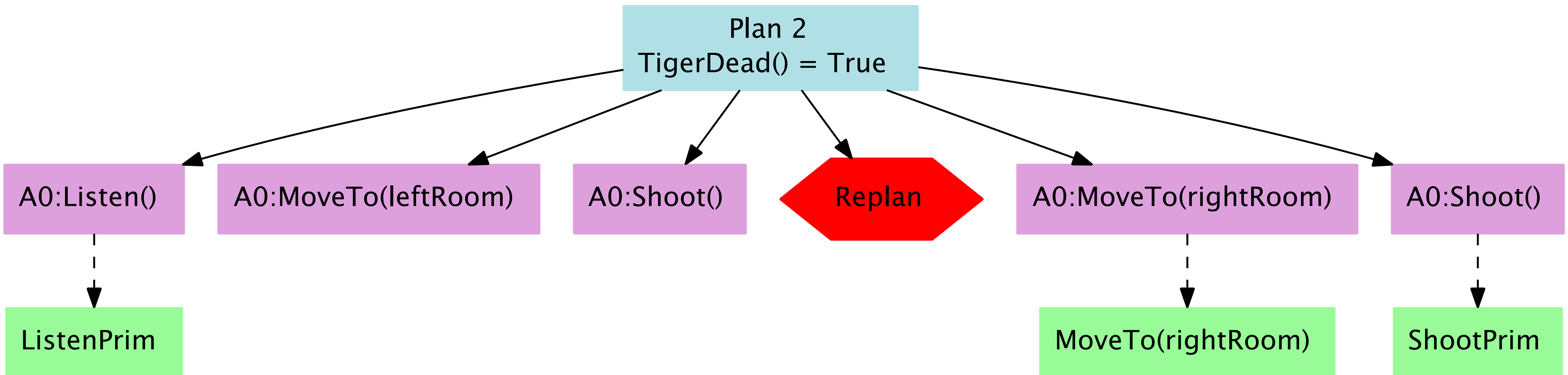
move(Room):
res: robotLoc = Room
listen:
pre: robotLoc = hall
result: KV(tigerLoc)
shoot:
pre: robotLoc = tigerLoc
result: deadTiger

P(tigerLoc = leftRoom) = 0.8



Leslie Pack Kaelbling, AAAI2010

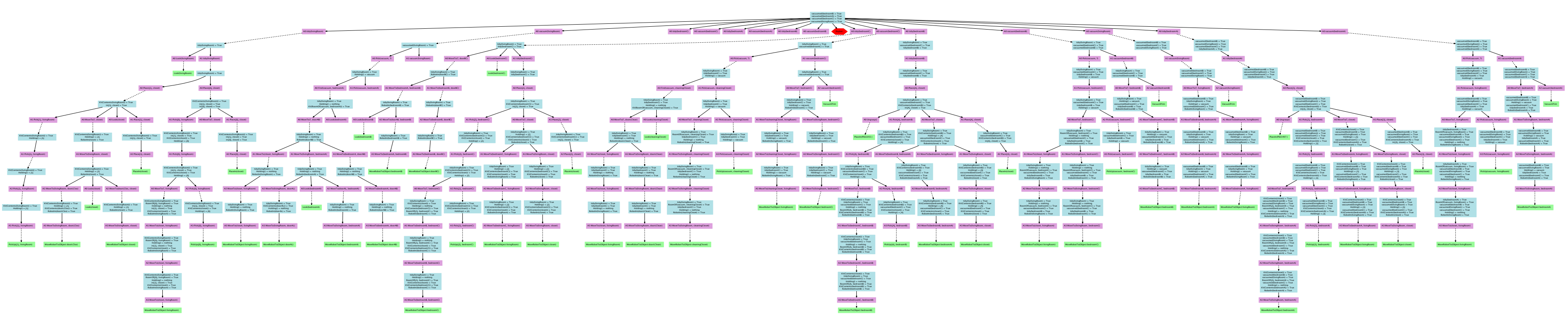
Monitor execution and replan

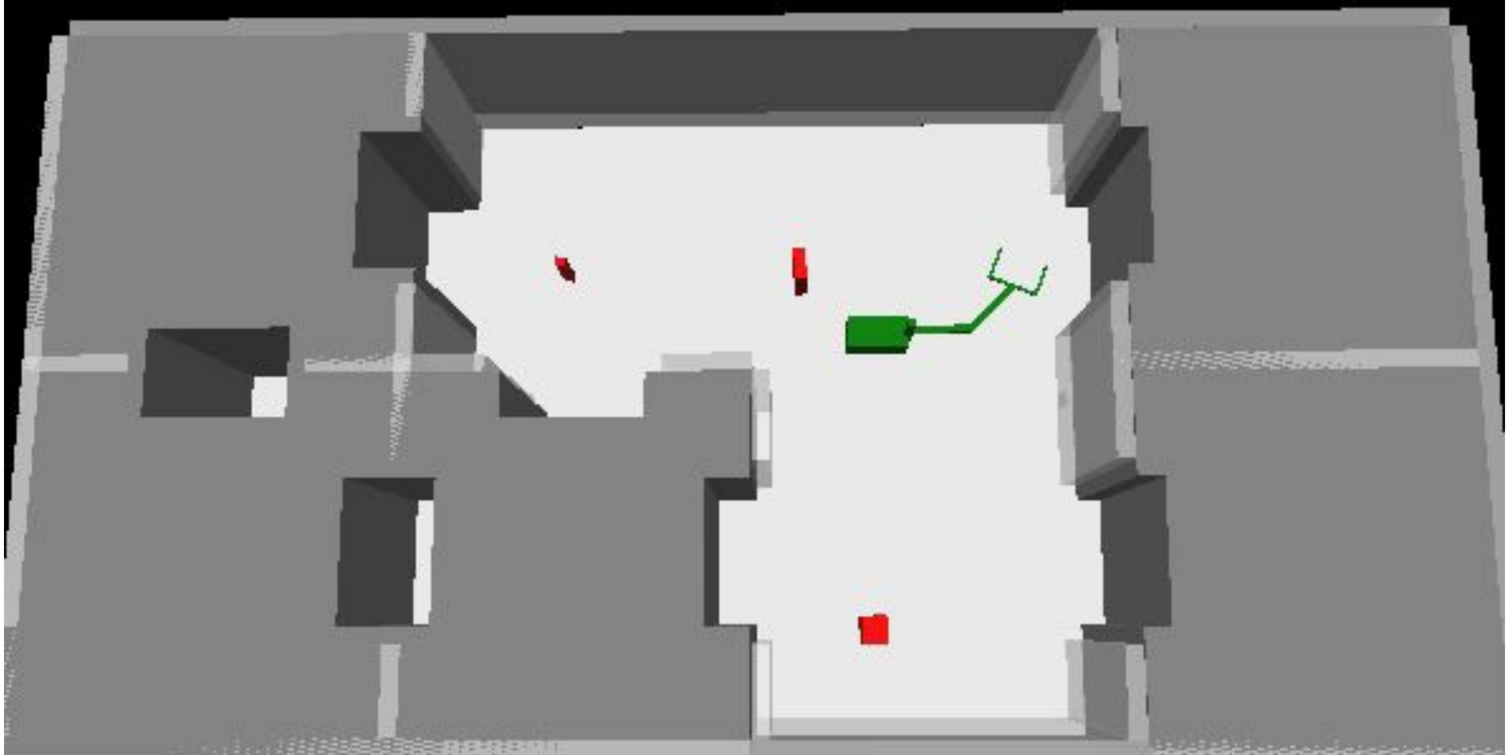


Cleaning house

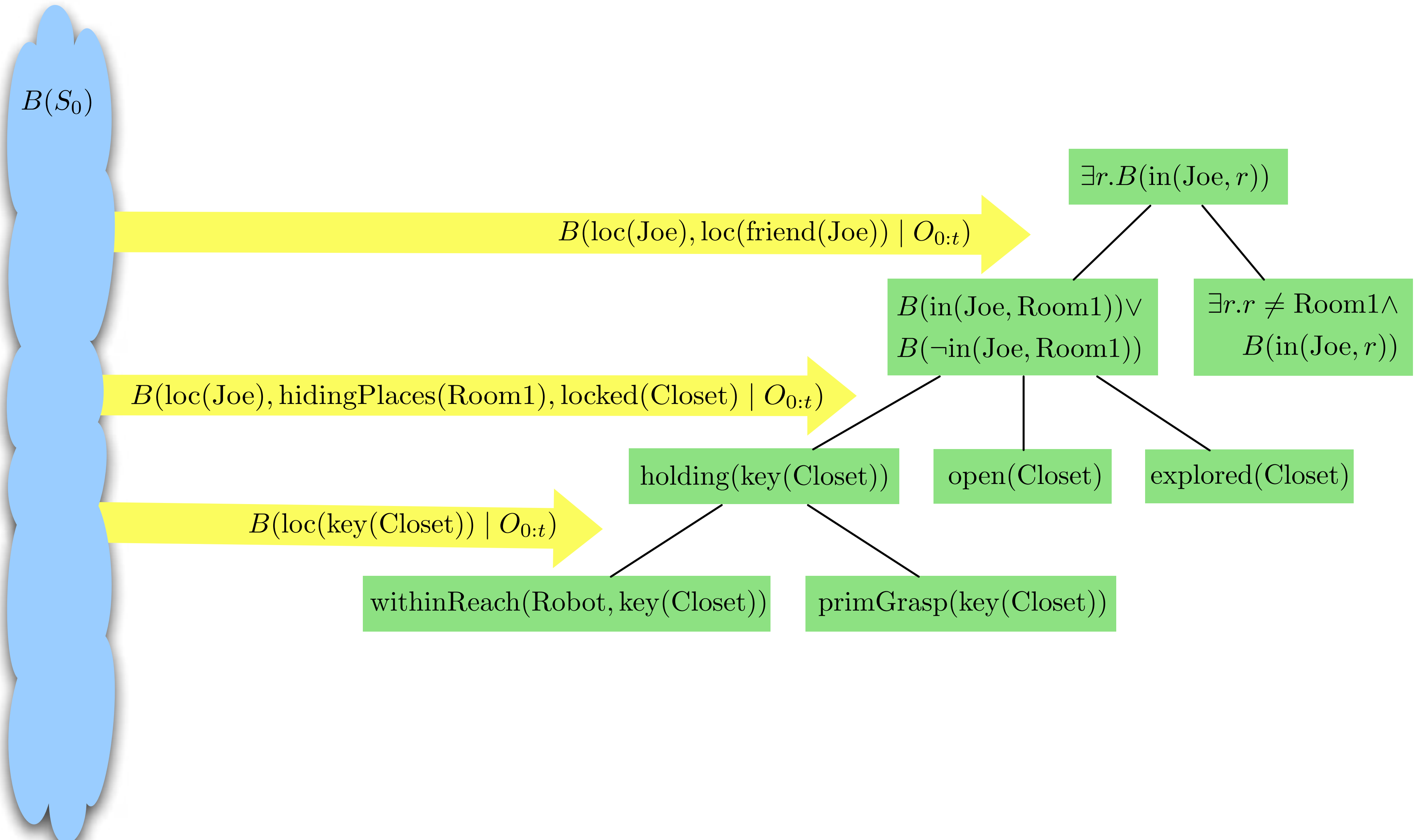
Goal: vacuum four of the rooms in the house

- have to put away junk items before vacuuming
- location of junk is unknown
- location of vacuum is unknown

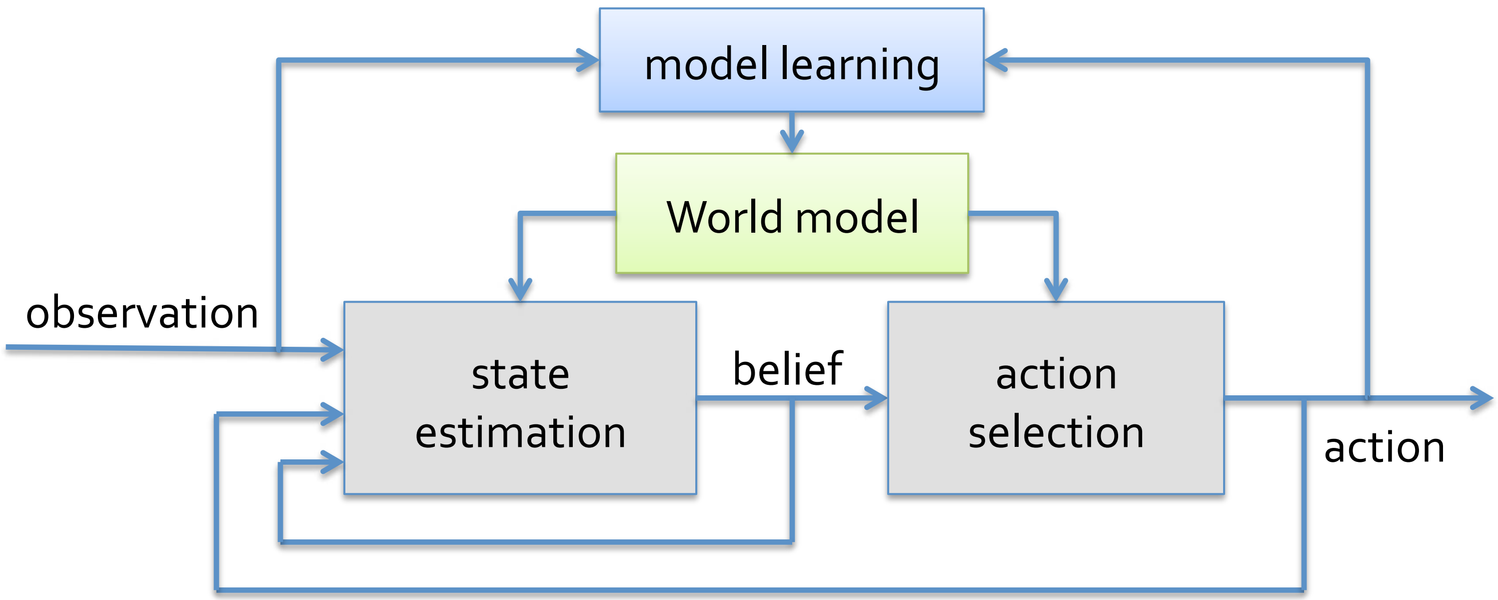




Plan hierarchy can pose small filtering problems

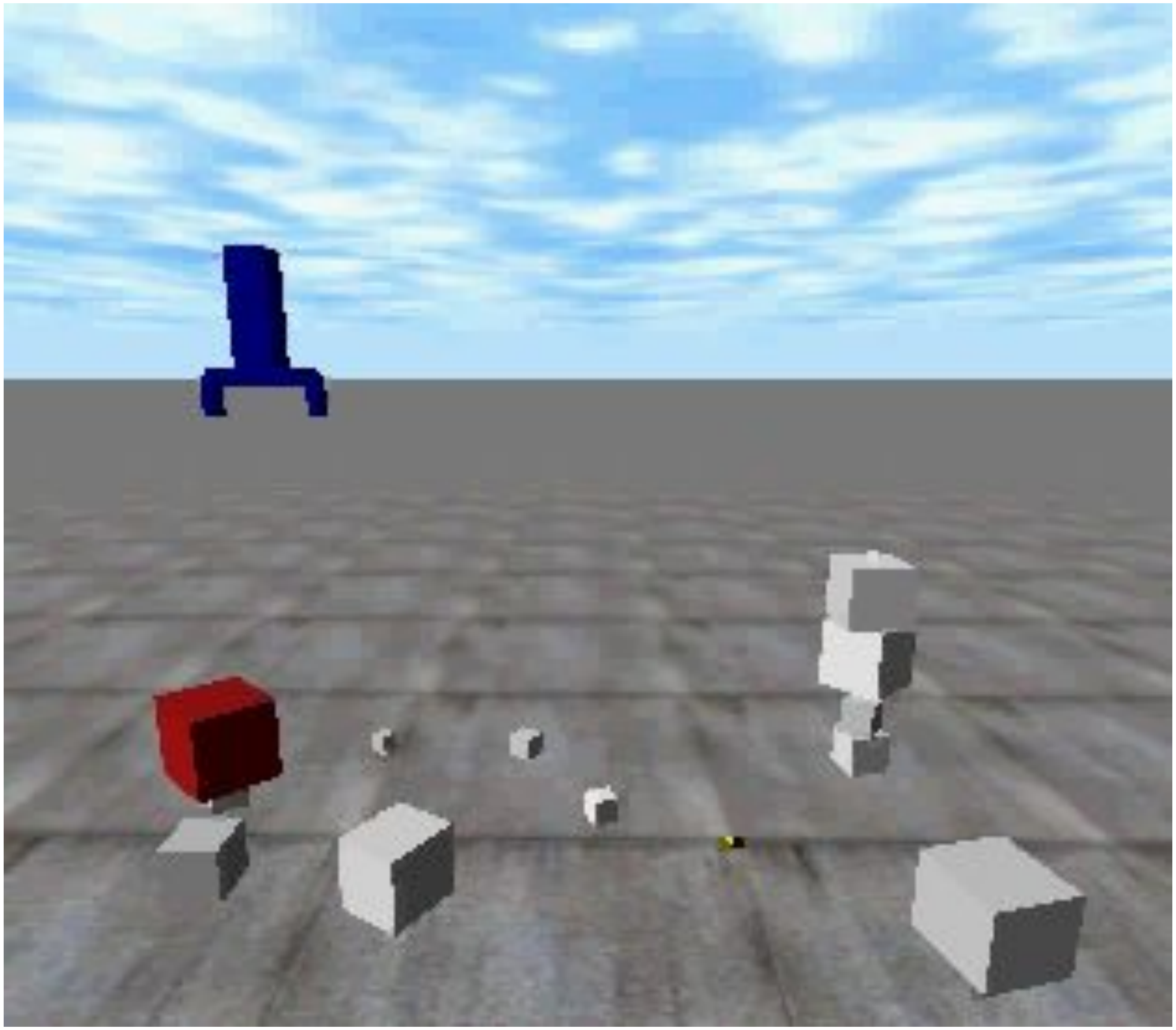


Learning a model



Joint work with Hanna Pasula and Luke Zettlemoyer

Blocks with physics



Representing a world model

Probabilistic state transition dynamics:

$$\Pr(s_t \mid s_{t-1}, a)$$

Representation should:

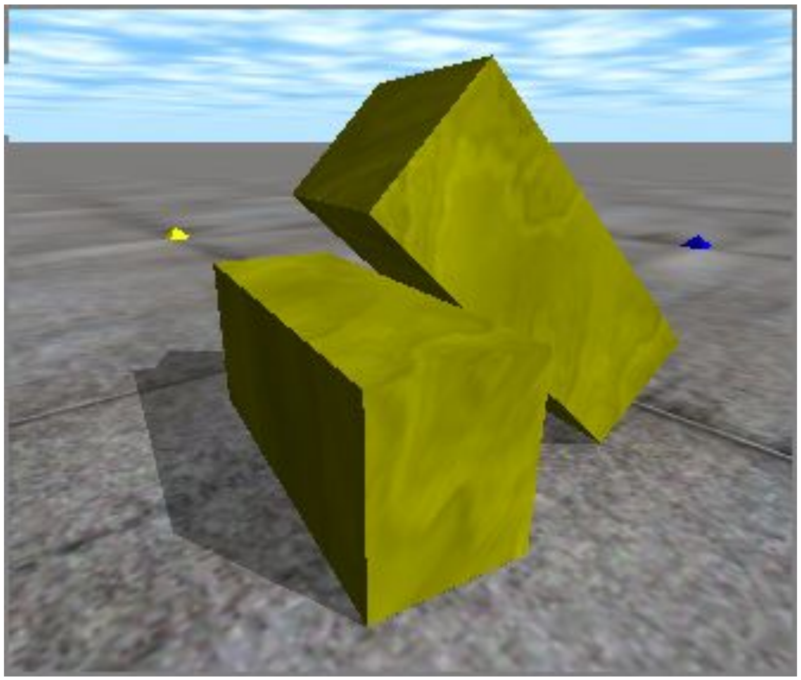
- allow effective generalization
- be useful for planning
- be efficiently learnable

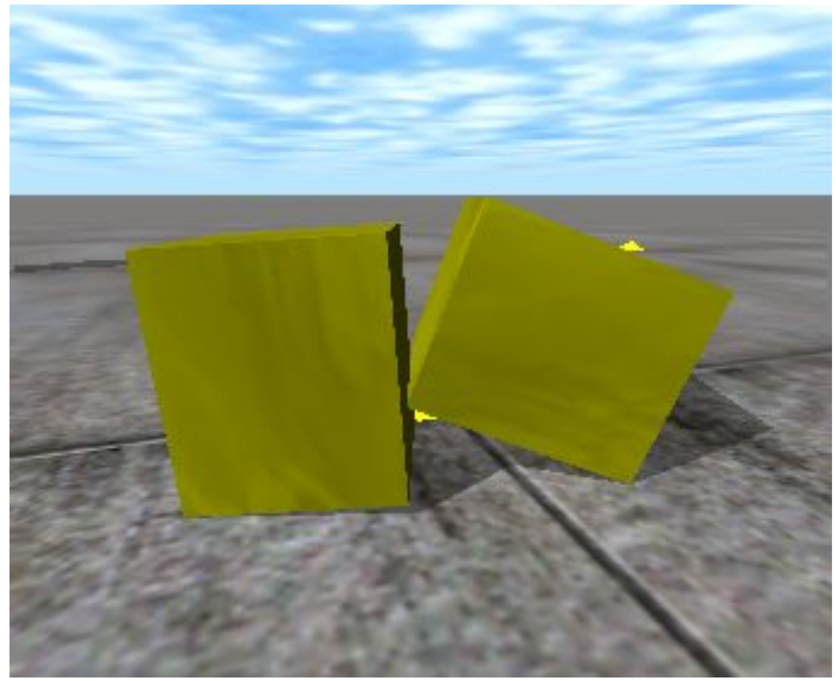
Probabilistic dynamic rules

Combine logic and probability to model effects of actions in complex, uncertain domains

```
pickup(X): {Y: on(X,Y)}
    clear(X), inhand-nil, size(X)>2, size(X)<7 →
      0.803 :¬on(X,Y)
      0.093 : no change
```

Is X *on Y*?





Useful symbolic vocabulary should be learned

Neoclassical learning

Given experience, $\{\langle s_t, a_t, s_{t+1} \rangle\}$

Find rule set that optimizes

$$\text{score}(R) = \sum_t \log \Pr(s_{t+1} \mid s_t, a_t, R) - \alpha |R|$$

Start with one default rule: "stuff happens"

- Symbolic: add, delete rule; change rule conditions
  - Greedy: choose set of outcomes
    - Convex optimization: find maximum likelihood probabilities

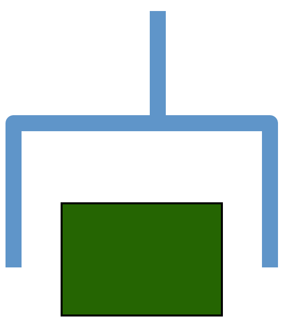
Concept invention

New concepts allow predictive theory to be expressed more compactly and learned from less data

`p1(X) :- ¬∃Y. on(X,Y)`	X is in the hand
`p2() :- ¬∃Z. p1(Z)`	nothing is in the hand
`p3(X) :- ¬∃Y. on(Y,X)`	X is clear
`p4(X,Y) :- on(X,Y)*`	X is above Y
`p5(X,Y) :- p3(X) ∧ p4(X,Y)`	X is on the top of the stack containing Y
`f6(X) :- #Y. p4(X,Y)`	the height of X

Rules learned from data

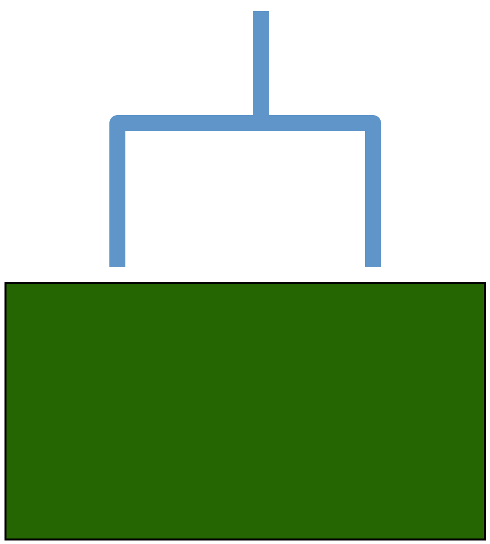
```
pickup(X): {Y: on(X,Y)}
    clear(X), inhand-nil, size(X)>2, size(X)<7→
      0.803 :¬on(X,Y)
      0.093 : no change
```



picking up middle-sized blocks usually works

Rules learned from data

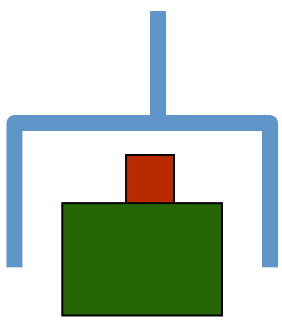
```
pickup(X):
    clear(X), inhand-nil, ¬size(X)<7 →
      0.906 : no change
```



it’s impossible to pick up very big blocks

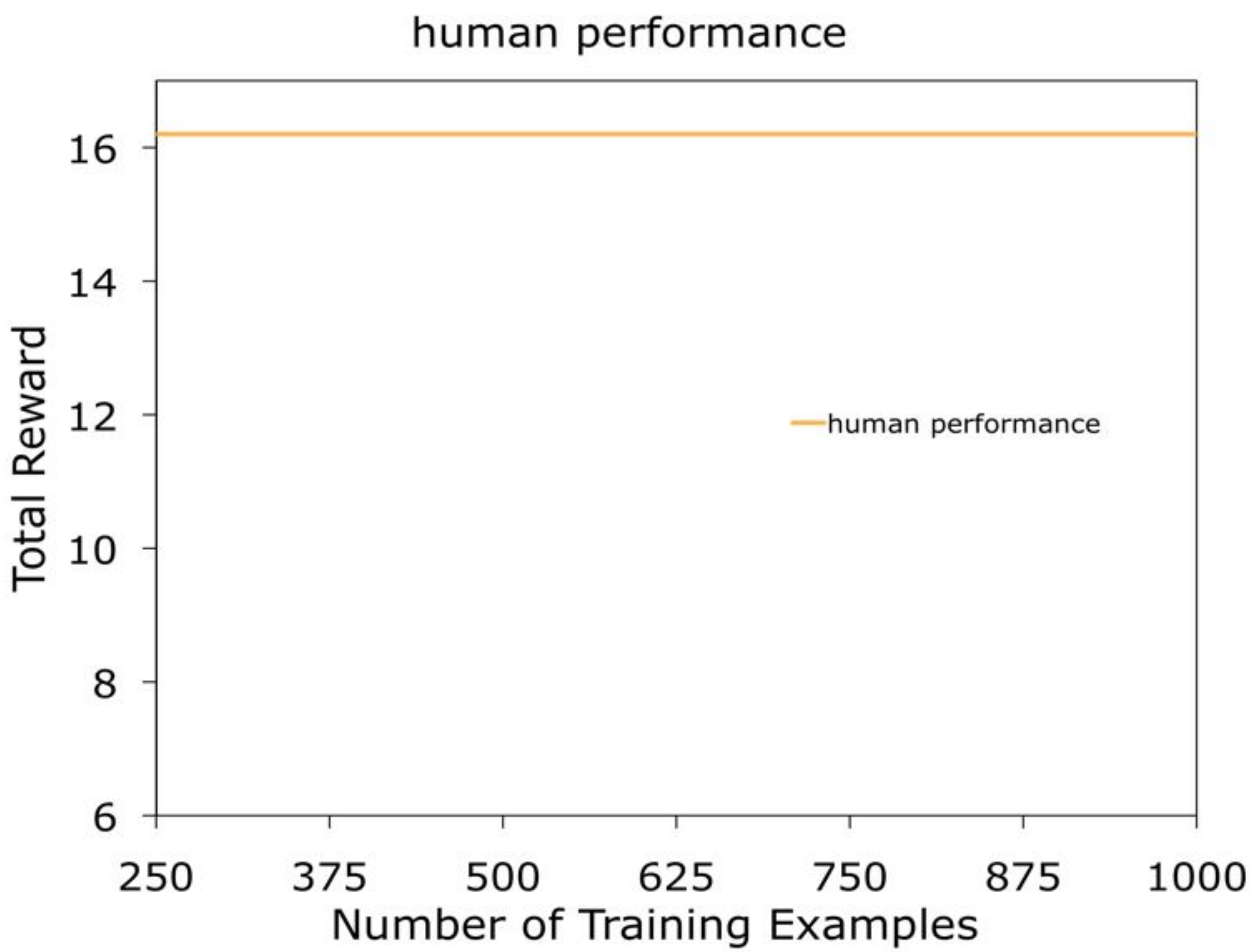
Rules learned from data

```
pickup(X): {T: table(T)}, {Y: on(X,Y), on(Y,T)}
   clear(X), inhand-nil, size(X)<2  →
      0.105 :¬on(X,Y)
      0.582 :¬on(Y,T)
      0.312 : no change
```

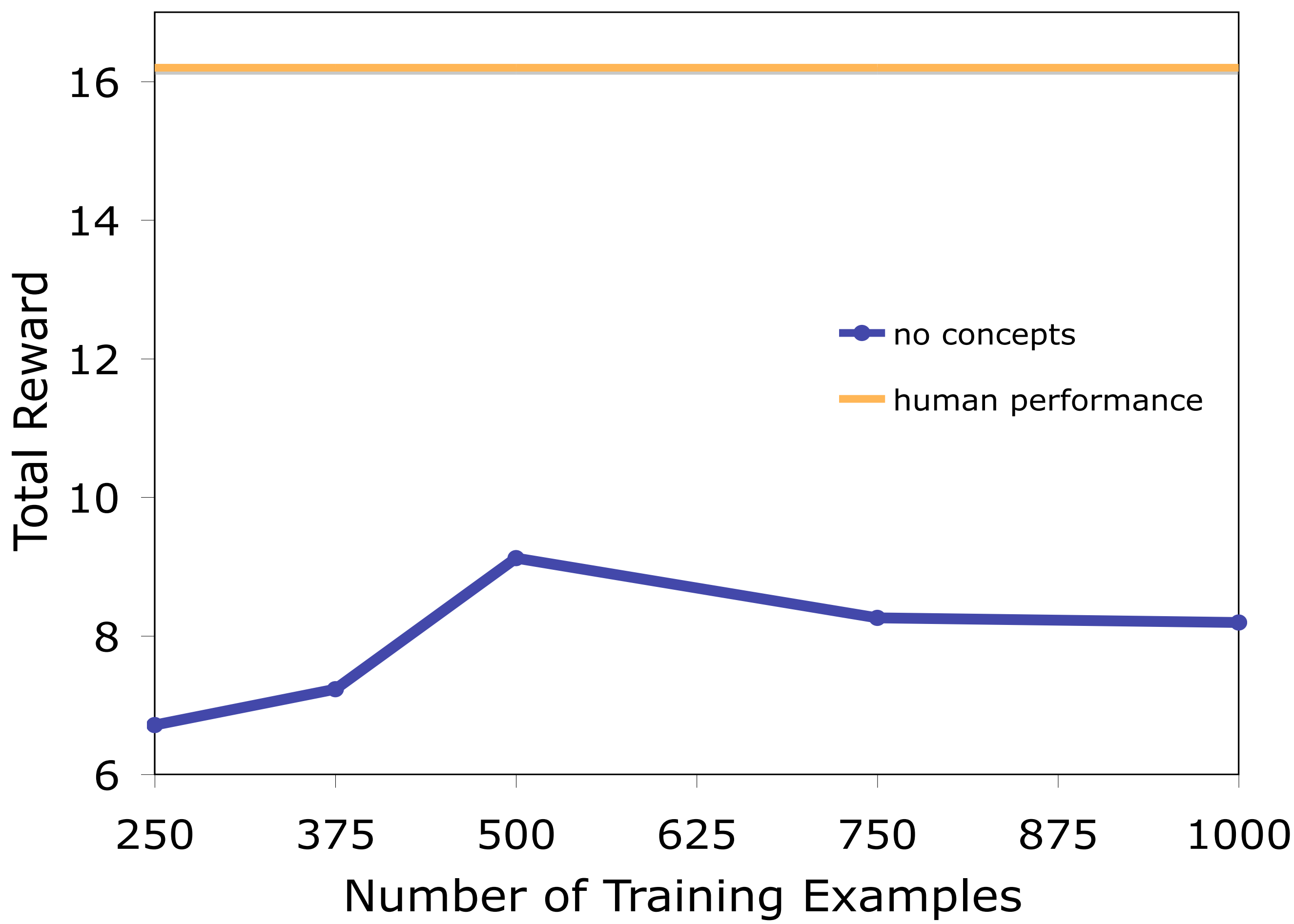


if a tiny block is on another block that is on the table, and we try to pick up the tiny block, we'll often pick up the other block as well, or fail

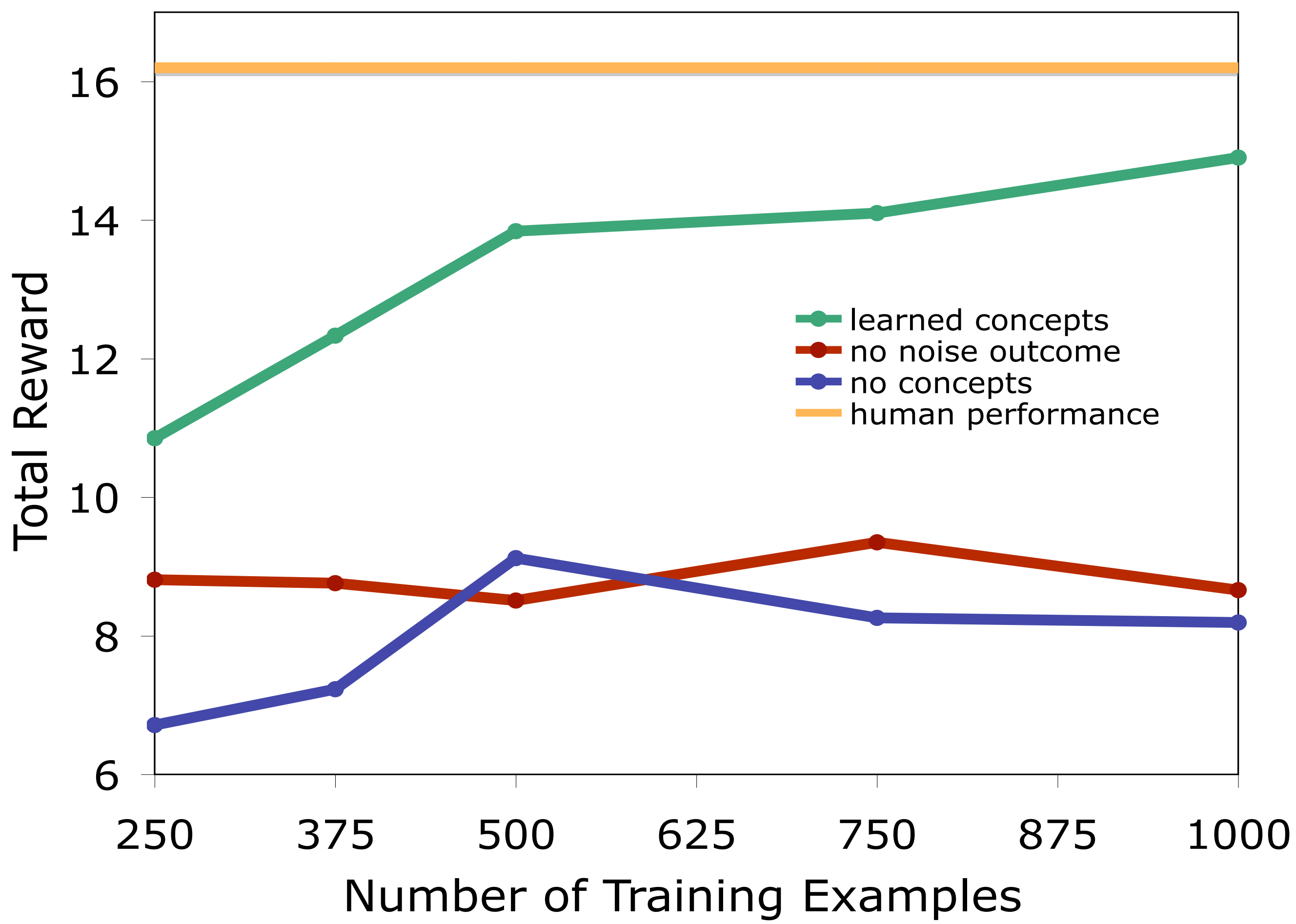
Planning with learned rules

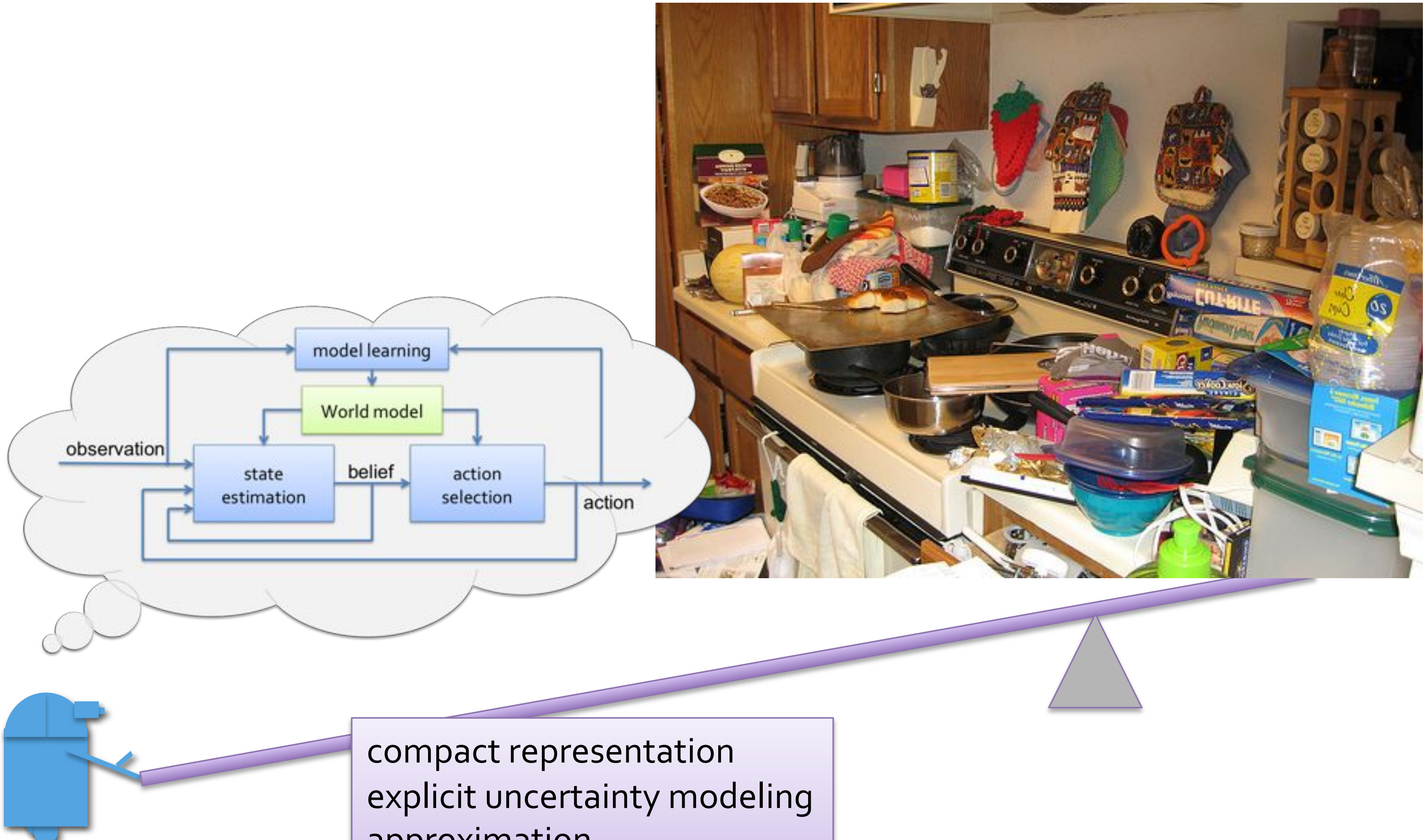


Planning with learned rules



Planning with learned rules





compact representation
explicit uncertainty modeling
approximation

What should we be doing?

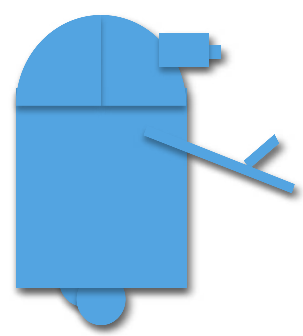
Thinking hard about representation in open, uncertain domains

- What do you know about your house?

Everything else: planning, learning, reasoning, …

Talking to each other

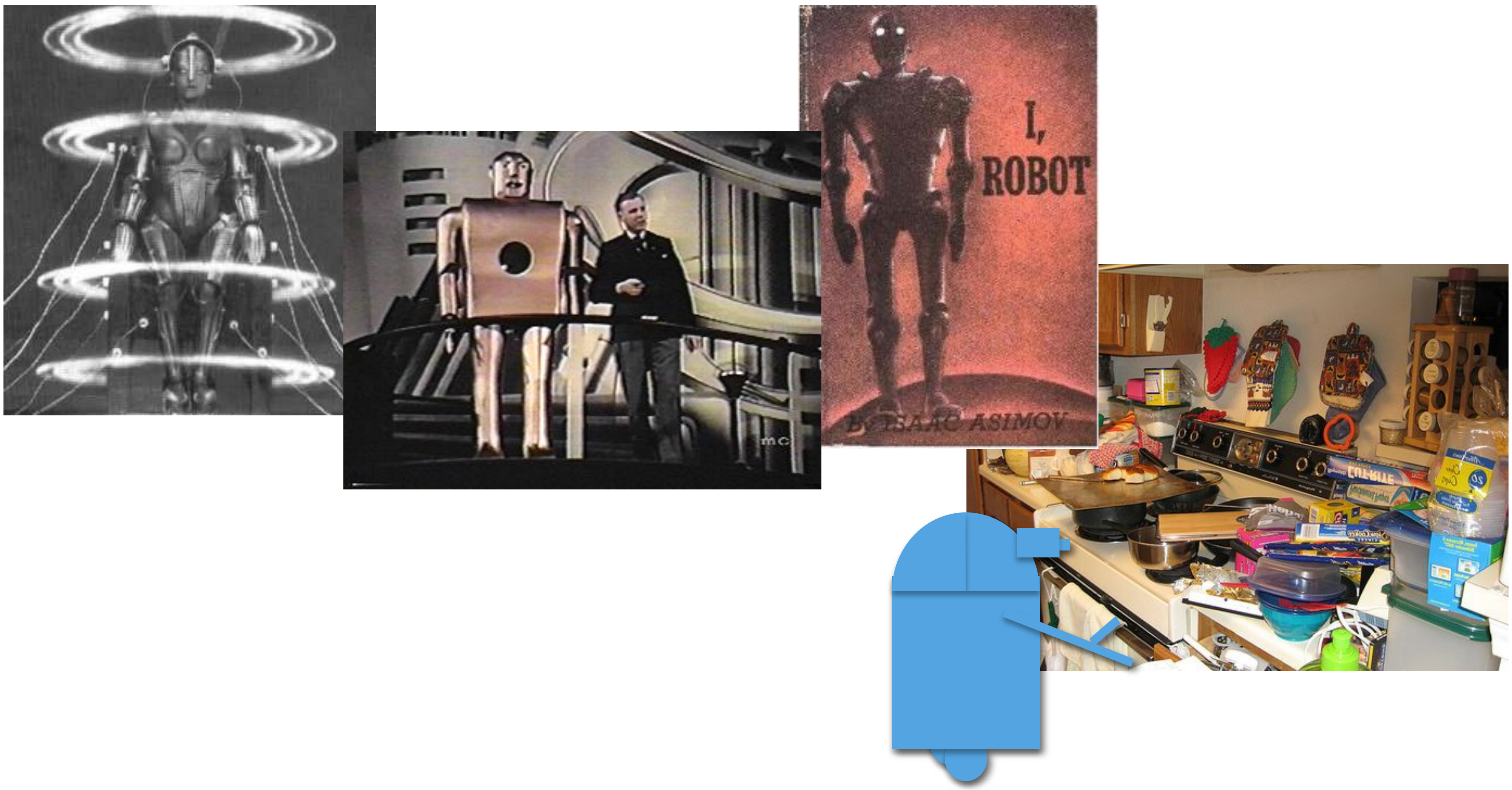
- vision, natural language, robotics, logic, probability, learning, …



Thanks!

Collaborators: Stan Rosenschein, Tom Dean, Tomas Lozano-Perez, Michael Littman, Tony Cassandra, Hagit Shatkay, Jim Kurien, Nicolas Meuleau, Milos Hauskrecht, Jak Kirman, Ann Nicholson, Bill Smart, Luis Ortiz, Leon Peshkin, Mike Ross, Kurt Steinkraus, Yu-Han Chang, Paulina Varshavskaya, Sarah Finney, Kaijen Hsiao, Luke Zettlemoyer, Han-Pang Chiu, Natalia Hernandez, James McLurkin, Emma Brunskill, Meg Aycinena Lippow, Tim Oates, Terran Lane, Georgios Theocharous, Kevin Murphy, Bruno Scherrer, Hanna Pasula, Brian Milch, Bhaskara Marthi, Kristian Kersting, Sam Davies, Dan Roy, Jenny Barry, Selim Temizer, Rob Platt, Russ Tedrake

Funders: NSF, DARPA, AFOSR, ONR, NASA, Singapore-MIT Alliance, Ford, Boeing, BAE Systems


I,
ROBOT
BY ISAAC ASIMOV