Welcome to the 8th Bellairs Workshop on Reinforcement Learning

and thoughts on this year's topic

Planning in Reinforcement Learning

by Rich Sutton
Reinforcement Learning and Artificial Intelligence Lab
University of Alberta

Participants

- Doina Precup, McGill University
- Joseph Modayil, University of Alberta
- Elliot Ludvig, Princeton University
- Rich Sutton, University of Alberta

- Matt Botvinick, Princeton University
- Michael Littman, Brown University
- Marc Bellemare, University of Alberta
- Subramanian Ramamoorthy, University of Edinburgh
- Tobias Jung, University of Liege, Belgium
- Erik Talvitie, Franklin and Marshall College
- Tom Schaul, Courant Institute, NYU
- Claudia Clopath, Columbia University
- Samuel Barrett, University of Texas, Austin
- Wouter Caarls, University of Delft
- Byron Boots, University of Washington
- Harm van Seijen, University of Alberta
- Dave Silver, University College London
- John O'Doherty, California Institute of Technology
- Yuan-Chang Leong, Princeton University
- Michael Bowling, University of Alberta
- Patrick Pilarski, University of Alberta
- Csaba Szepesvari, University of Alberta
- Scott Sanner, NICTA, Australia
- Mausam, University of Washington
- Andrey Kolobov, University of Washington
- Richard Dearden, University of Birmingham, UK
- Pascal Poupart, University of Waterloo
- Russ Greiner, University of Alberta
- Christopher Lin, University of Washington
- Alec Solway, Princeton University
- Dan Weld, University of Washington
- Daan Wierstra, Deepmind Technologies
- Clement Gehring, McGill University
- Pierre-Luc Bacon, McGill University

http://barbados2012.rl-community.org

Program – Barbados 2013

barbados2013.rl-community.org/program

Barbados 2013

Search this site

- Home
- Theme
- Participants
- Accomodation
- Barbados Fun
- **Program**
- Prior Meetings

2 days until Workshop

www.rl-community.org

Program

Sunday, April 28

Morning:

Where do models come from?

9:00 - 9:45	Rich Sutton - *Planning in RL*
9:45 -10:30	Byron Boots - *Hilbert Space Embeddings of Predictive State Representations*
10:30 -11:00	Break
11:00 -11:45	Erik Talvitie - *Learning to make predictions in high-dimensional, partially observable domains*
11:45 -12:15	Scott Sanner -*Symbolic Dynamic Programming for Hybrid MDPs*
12:15 -12:45	Marc Bellemare - *Pixels and Priors: Learning a Generative Model of Atari 2600 Games*

Evening:

7:00 - 8:00	Mausam and Andrey Kolobov -- *An Overview of MDP Planning Research at ICAPS* [Lecture Room]
8:10 - 9:30	Discussion on *Representations for Planning* led by Scott Sanner & Michael Littman [Dinnner Area]

Q: What is planning?

A: Planning is any computation from

1) a model of the world's dynamics, and

2) a goal

to

3) a fast way of making good decisions

Planning has been changing to become more interactive

- Classical AI
  - model is deterministic, tabular, correct, and complete
  - start and goal states
  - plan is a deterministic path
  - sensing is unnecessary, execution is irrelevant

Planning is one-shot, divorced from acting, learning, and sensing

- Modern issues
  - stochastic models
  - learned models
  - incomplete state
  - function approximation
  - temporal abstraction

Planning is continual, interacts with acting, learning, and sensing

RL view of planning

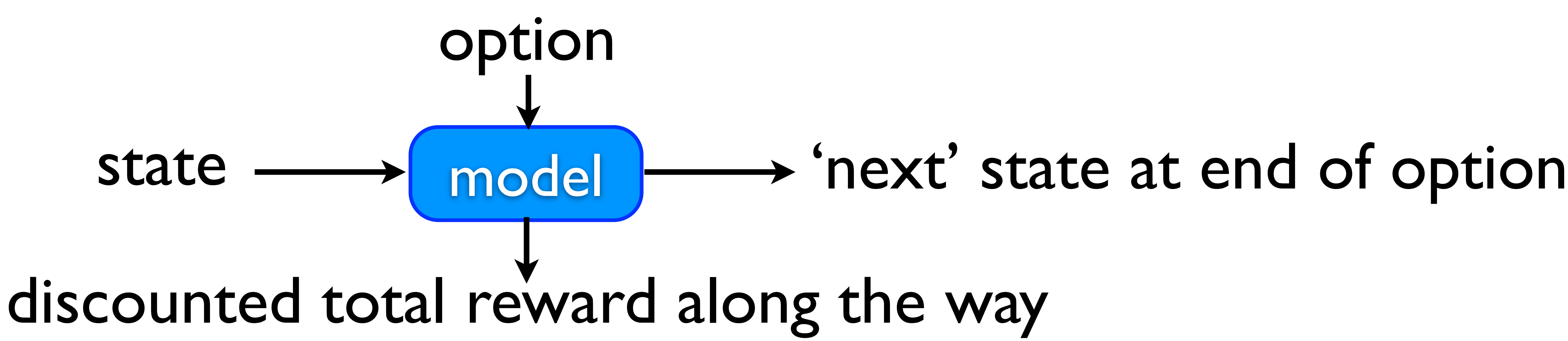
- Model is state-based, predictive, and stochastic
- Goal is reward
- Plan is policy / value function
- Planning is essentially DP-style backups
- Model may be continually learned
- Planning and model-free RL are alternative paths to the same goal
- Function approximation in value fn, policy, model
- Model may be temporally abstract

Big questions in planning

- How does planning interact with action taking?
- What sort of models are needed for planning?
  - How important is it to include *partial observability*?
  - How important is it to include *temporal abstraction*?
- How can we learn such models?
- How can we elegantly include *planning to subgoals*?
- Planning is never complete; how can we *order the computations* efficiently and robustly?
- Does *policy-gradient RL* allow a better interplay of planning and action?

Temporally abstract models

- Option models (option = policy + termination fn)
- Can be learned efficiently by off-policy methods

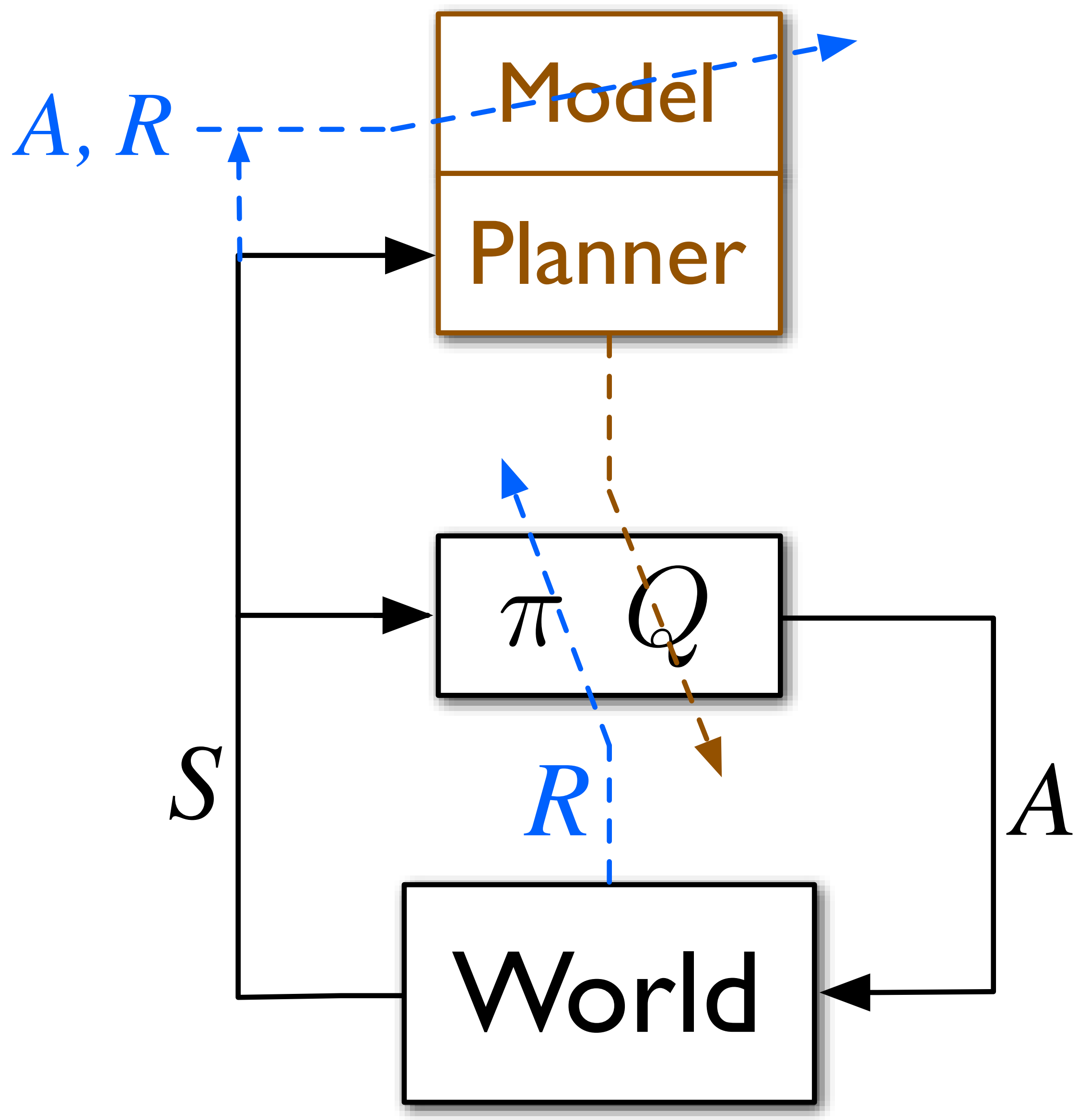


- Plug compatible with conventional 1-step models in Bellman equations and DP value-fn backups
- Can be constructed from general value functions

Foreground/background
a fundamental architectural decomposition

- Planning is inherently computation intensive, thus slow, incremental, incomplete
- Interaction should be fast, as fast as possible
- Some things have to be done at the speed of interaction (the foreground), all other things, including planning, should be in the background

Foreground-background architecture



Planning is in the background

Interaction and learning are in the foreground

Big questions in planning

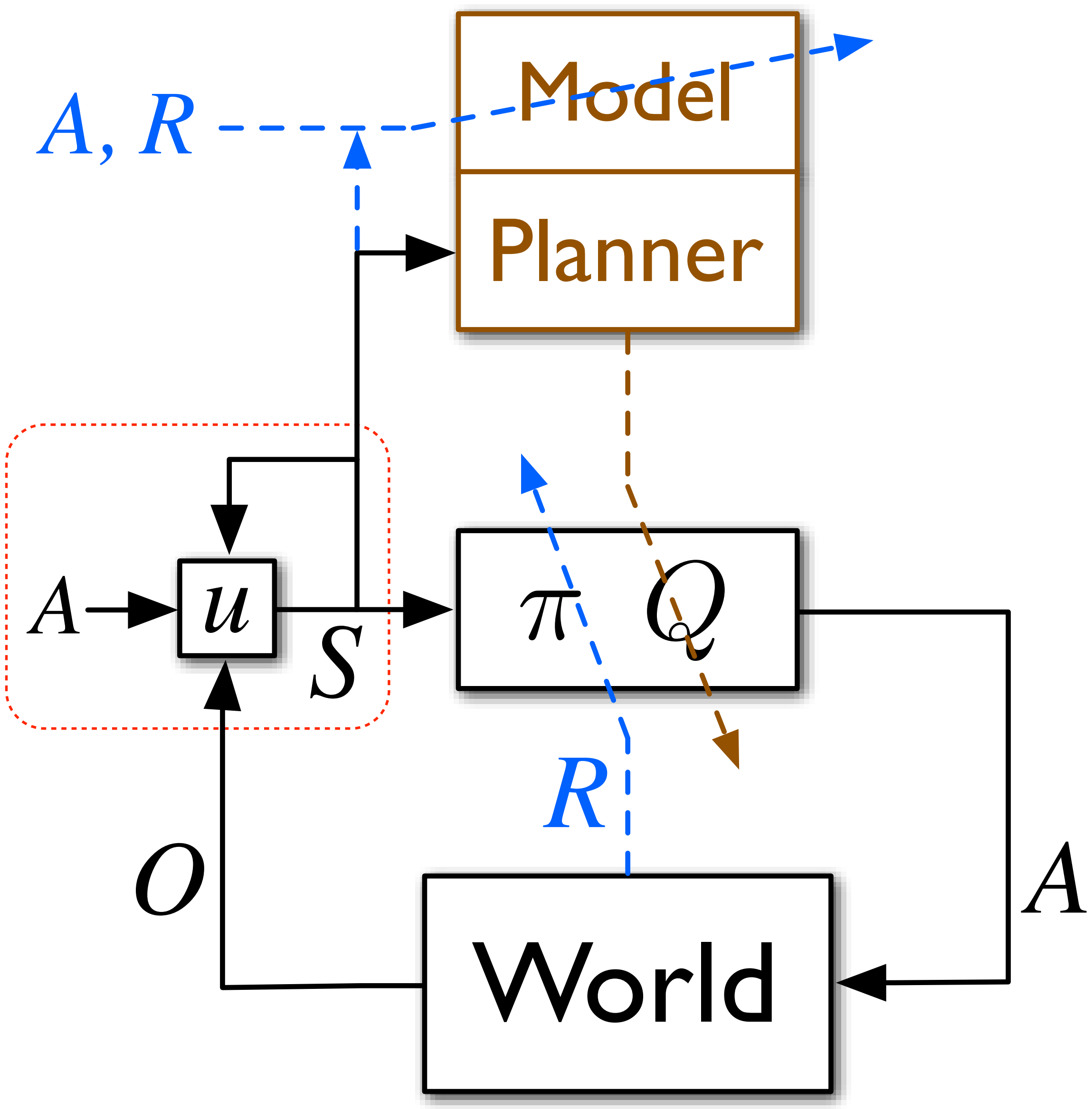
- How does planning interact with action taking?
  - Via the policy and/or value fn used by the foreground
- What sort of models are needed for planning?
  - How important is it to include *partial observability*?
  - How important is it to include *temporal abstraction*?

Background on

Partial observability

- The input from the world is an observation rather than a ‘state’
- The agent must construct its own state representation to use as state (agent state, belief state)
- Extensive theory of POMDPs
  - Bayesian belief state
  - Planning by simulating observations and state updates in response to them
  - Computationally complex

Foreground-background architecture *with partial observability*

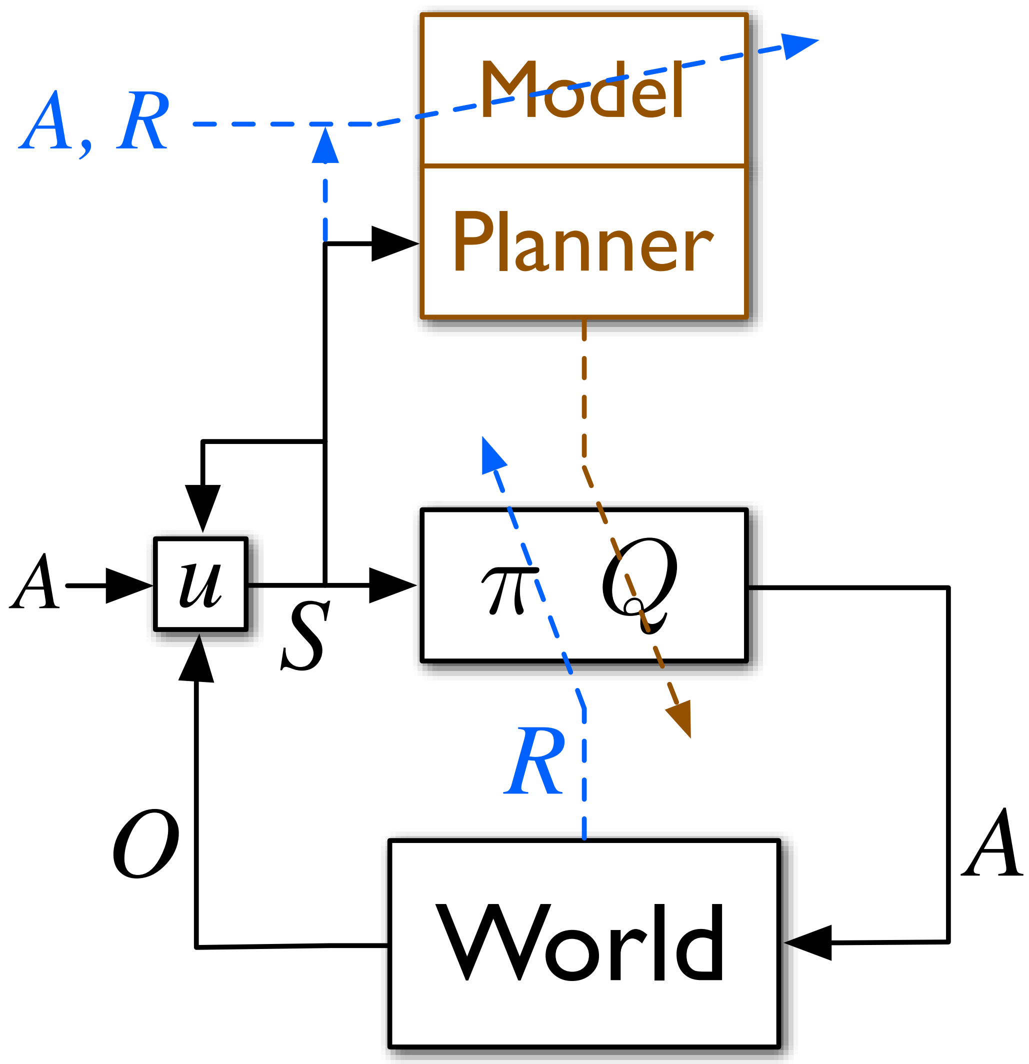


Planning is in the background

State update is also in the foreground

Interaction and learning are in the foreground

Agent state and its update

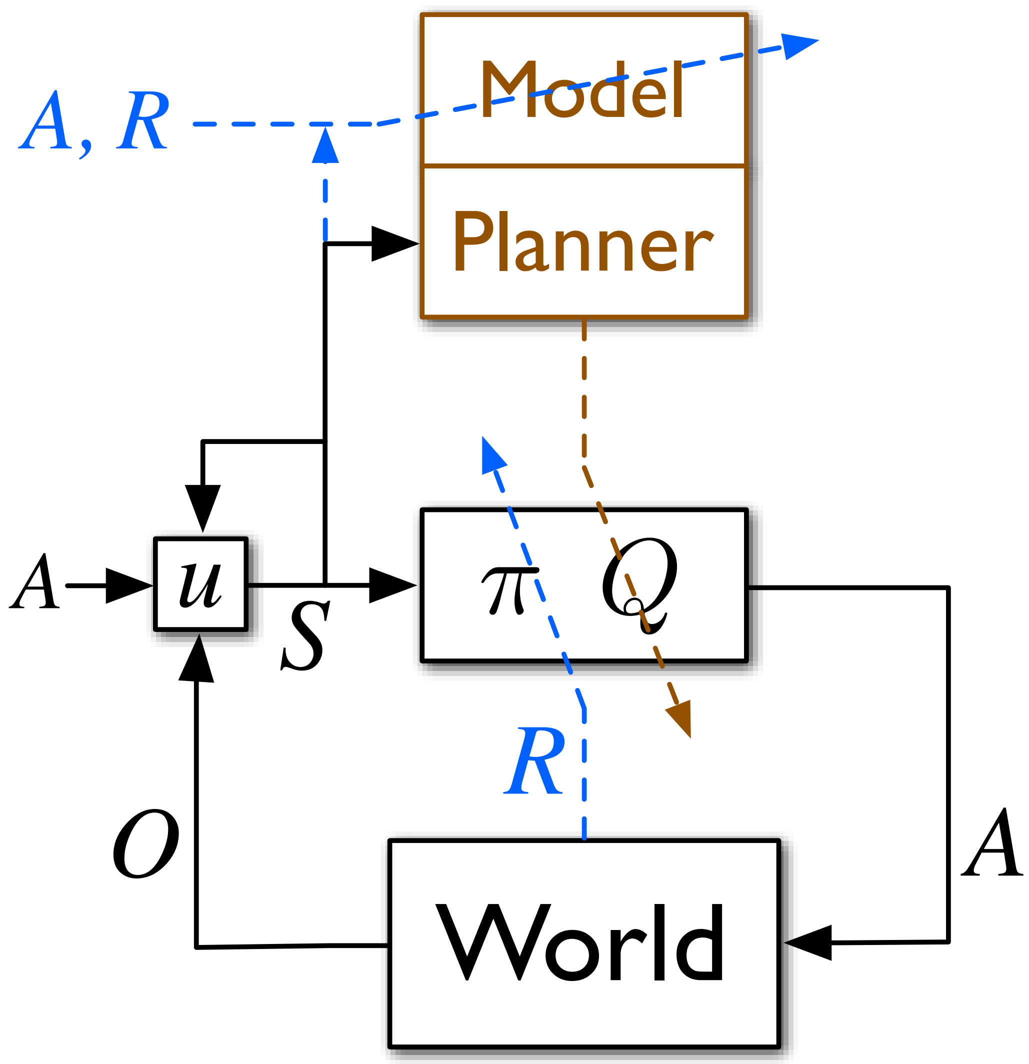


- Agent state is whatever the agent uses as state
  - in policy, value fn, model...
  - may differ from env state and information state
- State update:

  $$S_{t+1} = u(S_t, A_t, O_{t+1})$$

  - e.g., Bayes rule, k-order Markov (history), PSRs, predictions

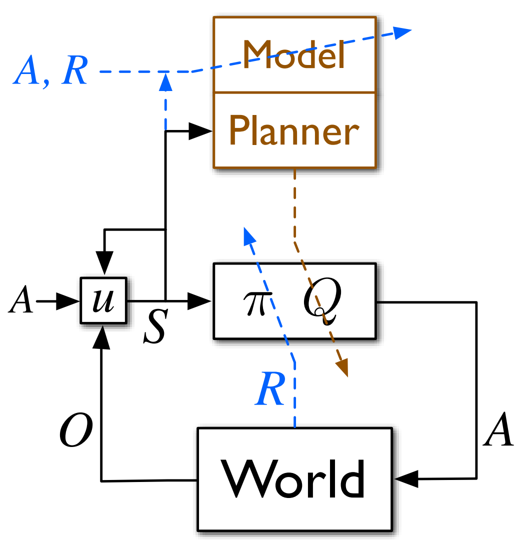
Planning should be state-to-state



$$S_{t+1} = u(S_t, A_t, O_{t+1})$$

- State update is in the foreground!
  - Planner and model see *only states*, never observations
- We lost this with POMDPs; Why?
  - Classical and MDP planning were always state-to-state
  - Planning can always be state-to-state in information state
- Function approximation makes planning in the info state a natural, flexible, and scalable approach

Why partial observability should be separated from planning



$$S_{t+1} = u(S_t, A_t, O_{t+1})$$

- Because it can be! There is just no reason to treat this case specially
- It's simpler; we save one branching and collecting step
  - In the temporally abstract case we save much more
- We actually have the data available to learn the model
- Approximation is natural and robust

Conclusions

Big answers in planning

- How does planning interact with action taking?
  - Via the policy and/or value fn used by the foreground
- What sort of models are needed for planning?
  - option models with function approximation
- How important is it to include *partial observability*?
  - Important in the foreground,
    not at all in planning or models
- How important is it to include *temporal abstraction*?
  - Important in planning and models,
    not at all in the foreground
- How can we learn such models?
  - By off-policy TD methods such as GTD, GQ, and HTD