Hierarchical POMDPs

Tuesday, January 20, 2009

Singapore

Pascal Poupart

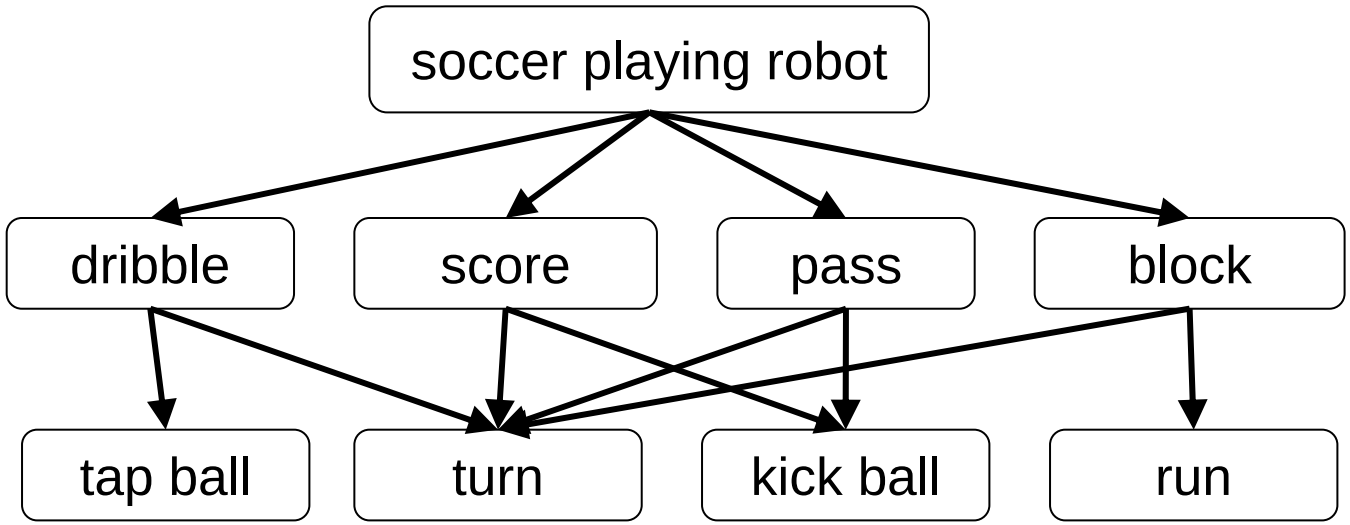
University of Waterloo

Outline

- What is a hierarchy?
- Action hierarchies
  - Policy contingent abstractions
  - Hierarchical controllers
  - Recursive controllers
  - Hierarchy discovery
- State hierarchies
  - Temporal & Spatial abstraction
  - Hierarchical HMMs
  - HPOMDPs

What is a hierarchy?

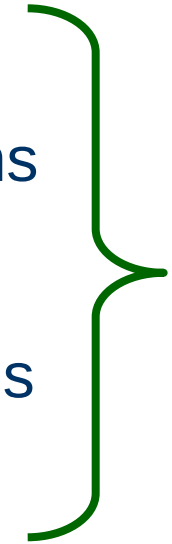
- Idea: task/process decomposed into subtasks/subprocesses arranged hierarchically



- Robot control: action hierarchy
- Behaviour recognition: state hierarchy

Why hierarchies?

- Temporal abstraction
  - Abstract actions: sequence of actions
- Spatial abstraction
  - Abstract states: aggregation of states
- Sub-policy/process reuse



Improved efficiency

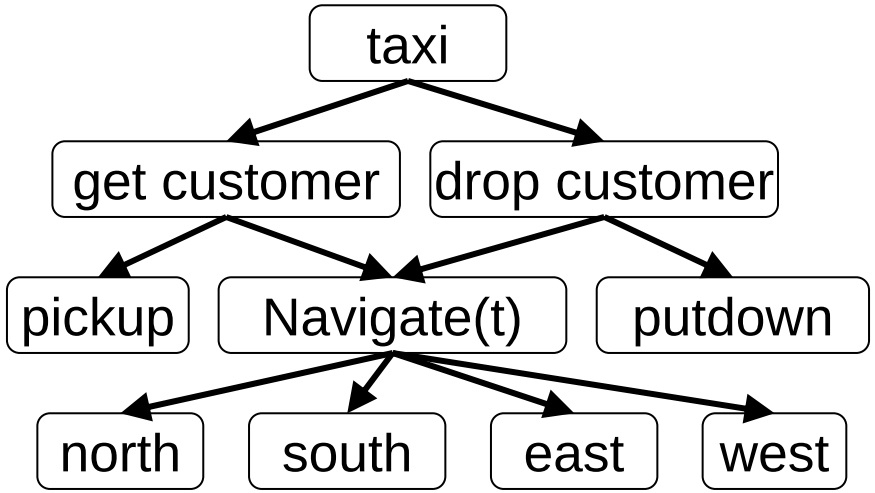
- More intuitive representation

Action Hierarchies in RL

- Augment actions with options aka macro actions
  - Policy π : S x A → [0,1]
  - Termination condition β : S → [0,1]
  - Input set I $\subseteq$ S (pre-condition)
- Semi-Markov decision process
  - $V^*(s) = \max_a R(s,a) + \Sigma_{s'} \gamma^t Pr(s',t|s,a) V^*(s')$
- Benefit: learn faster (less exploration)
  - Assuming the options/macro actions are good
  - Exploit temporal abstraction

Action Hierarchies in MDPs

- No learning… but can we speed up computation?
  - Yes: exploit state abstraction
- Design hierarchy of tasks



- Task h:
  - Subset of actions (leaves of subtree)
  - local reward function $R_h(s,a)$ (optional)
  - termination condition $\beta_h : S \rightarrow [0,1]$ (optional)

Action Hierarchies in MDPs

- Policy optimization
  - Full optimality: $V^*(s) \geq V^{\pi}(s) \quad \forall \pi$
  - Hierarchical optimality: $V^*(s) \geq V^{\pi}(s) \quad \forall$ hierarchical π
  - Recursive optimality: $V^*(s) \geq V^{\pi}(s) \quad \forall$ recursively built π

- Most common: aim for recursive optimality
  - Optimize sub-policies, bottom up
- Advantages:
  - Simple: solve sequence of small MDPs
  - State abstraction: policy-contingent abstraction (PolCA, Pineau et al.)

State Abstraction

- Idea: some state features may not be necessary

- PolCA (Pineau et al.)
- Aggregate equivalent states (Dean & Givan 97)
  - $s_1, s_2 \in C_i$ (cluster) iff
    - $R(s_1,a) = R(s_2,a) \quad \forall a$
    - $\Sigma_{s' \in C} \Pr(s'|s_1,a) = \Sigma_{s' \in C} \Pr(s'|s_2,a) \; \forall a,C$

- Could also use algebraic decision diagrams (ADDs)

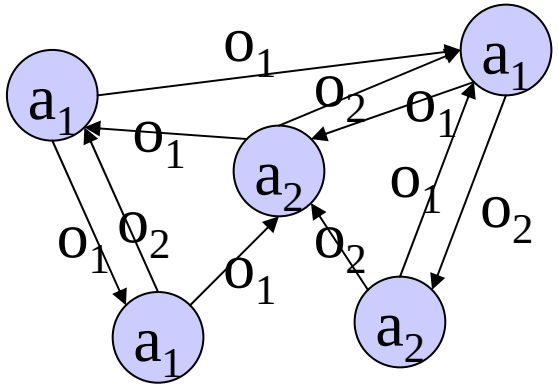
Action Hierarchies in POMDPs

- PolCA+ (Pineau et al.)
- Hierarchy of subtasks where each task:
  - subset of actions
  - local reward function
  - no termination condition (states are not observable)

- State abstraction (policy contingent abstraction)
  - $s_1, s_2 \in C_i$ (cluster) iff
    - $R(s_1,a) = R(s_2,a) \quad \forall a$
    - $\Sigma_{s' \in C} \Pr(s'|s_1,a) \Pr(o|a,s') = \Sigma_{s' \in C} \Pr(s'|s_2,a) \Pr(o|a,s') \quad \forall a,o,C$
  - Could also use ADDs

Hierarchical Controllers

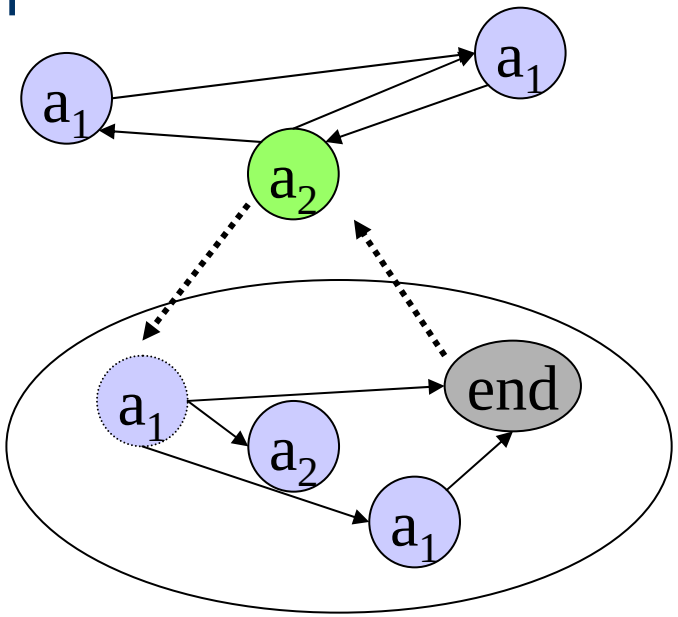
- Alternative policy representation: controllers
  - Action mapping: α: N → A or Pr(a|n)
  - Next node mapping: σ: N x O → N or Pr(n'|a,n)



- Can we use hierarchies with controllers?
  - Yes: Hansen and Zhou, 2003

Hierarchical Controllers

- Idea: let some nodes be sub-controllers
  - Action mapping: α: N → A or Pr(a|n)
  - Next node mapping: σ: N x O → N or Pr(n'|n,o)
  - Child mapping (abstract nodes): ϕ: N → N or Pr(n'|n)
  - Local reward function
  - Special exit nodes



Hierarchical Controllers

- Transition prob of abstract nodes: Pr(s'|â,s)
  - Discounted occupancy frequency
  - $f(s',n') = Pr(n'|n^{par}) + \gamma \Sigma_{snao'} f(s,n) Pr(a|n) Pr(s'|s,a) Pr(o'|a,s') Pr(n'|n,o')$

- Policy optimization
  - Bottom up optimization
  - Policy iteration
    - Needs less nodes per subtask
    - Can also exploit state abstraction

Recursive Controllers

- Can we let sub-controllers call themselves?
  - Yes: Charlin, Poupart & Shioda 2006
- Recursive controllers:
  - Infinite hierarchy
  - Could be useful in natural language processing tasks
  - Note that
    - Hierarchical controllers ⇔ regular expressions
    - Recursive controllers ⇔ context-free grammars
- Policy optimization
  - Can't use bottom up optimization
  - Must optimize all levels simultaneously

Hierarchy Discovery

- Could we discover the hierarchy?
  - Yes: Charlin, Poupart & Shioda 2006
- Policy optimization: aim for hierarchical optimality
  - Non-convex quartic optimization problem
  - Use non-convex solvers and/or approximate problem
    - Not scalable

Hierarchy Discovery

e 2: Non-convex quarticly constrained optimization problem for hierarchy and policy discovery
ounded stochastic recursive controllers.

$$\max_{x,y,z} \quad \sum_{s\in S} b_0(s)\underbrace{V_{n_0}(s)}_{y} \tag{3}$$

$$\underbrace{V_n(s)}_{y} = \sum_{a,n'}\Big[\underbrace{\Pr(n',a|n,o_k)}_{x} R(s,a) + \sum_{s',o}\Pr_\gamma(s'|s,a)\Pr(o|s',a)\underbrace{\Pr(n',a|n,o)}_{x}\underbrace{V_{n'}(s')}_{y}\Big] \quad \forall s,n \tag{4}$$

$$\underbrace{V_{\bar{n}}(s)}_{y} = \sum_{n_{beg}}\underbrace{\Pr(n_{beg}|\bar{n})}_{z}\Big[\underbrace{V_{n_{beg}}(s)}_{y} + \sum_{s_{end},a,n'}\underbrace{\mathrm{oc}(s_{end},n_{end}|s,n_{beg})}_{w}\Big[\underbrace{\Pr(n',a|\bar{n},o_k)}_{x} R(s_{end},a)$$
$$+\sum_{s',o}\Pr_\gamma(s'|s_{end},a)\Pr(o|s',a)\underbrace{\Pr(n',a|\bar{n},o)}_{x}\underbrace{V_{n'}(s')}_{y}\Big]\Big] \quad \forall s,\bar{n} \tag{5}$$

$$\underbrace{oc(s',n'|s_0,n_0)}_{w} = \delta(s',n',s_0,n_0) + \sum_{s,o,a}\Big[\tag{6}$$

$$\sum_{n}\underbrace{oc(s,n|s_0,n_0)}_{w}\Pr_\gamma(s'|s,a)\Pr(o|s',a)\underbrace{\Pr(n',a|n,o)}_{x}\Big] \Big\} \; n \text{ concrete (6a)}$$

$$+\sum_{s_{end},n_{beg},\bar{n}}\underbrace{oc(s,\bar{n}|s_0,n_0)}_{w}\Pr_\gamma(s'|s_{end},a)\Pr(o|s',a)$$
$$\underbrace{oc(s_{end},n_{end}|s,n_{beg})}_{w}\underbrace{\Pr(n',a|\bar{n},o)}_{x}\underbrace{\Pr(n_{beg}|\bar{n})}_{z}\Big] \quad \forall s_0,s',n_0,n' \Big\} \; \bar{n} \text{ abstract (6b)}$$

$$\Pr(\bar{n}'|\bar{n}) = 0 \text{ if } label(\bar{n}') \le label(\bar{n}), \forall \bar{n},\bar{n}' \tag{7}$$

Hierarchy Discovery

- Why discover the hierarchy since the problem is the same as not having a hierarchy?
  - Search in a different policy space
  - Bias the search towards hierarchical policies
  - Sub-policy reuse
  - Reveal interesting structure

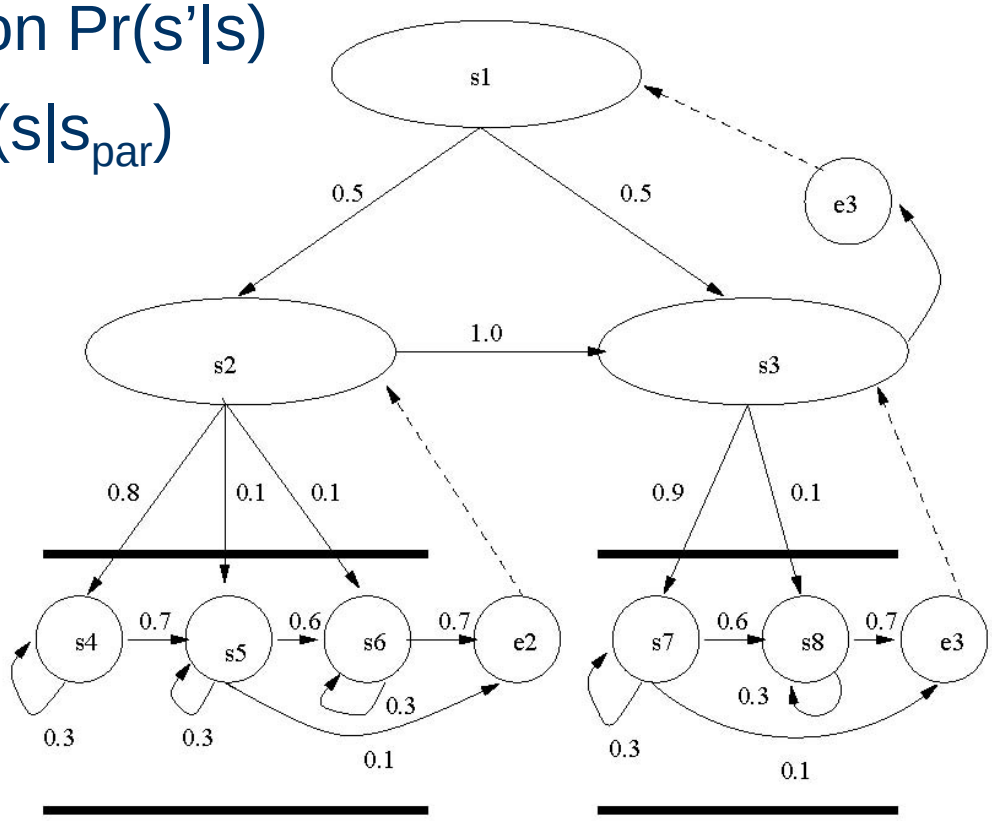
Outline

- What is a hierarchy?
- Action hierarchies
  - Policy contingent abstractions
  - Hierarchical controllers
  - Recursive controllers
  - Hierarchy discovery
- **State hierarchies**
  - **Temporal & Spatial abstraction**
  - **Hierarchical HMMs**
  - **HPOMDPs**

Hierarchies in HMMs

- Finn, Singer & Tishby 1998: state hierarchy
  - Emission distribution Pr(o|s)
  - Next state distribution Pr(s'|s)
  - Child distribution $Pr(s|s_{par})$
  - Exist states

- Similar to hierarchical controllers



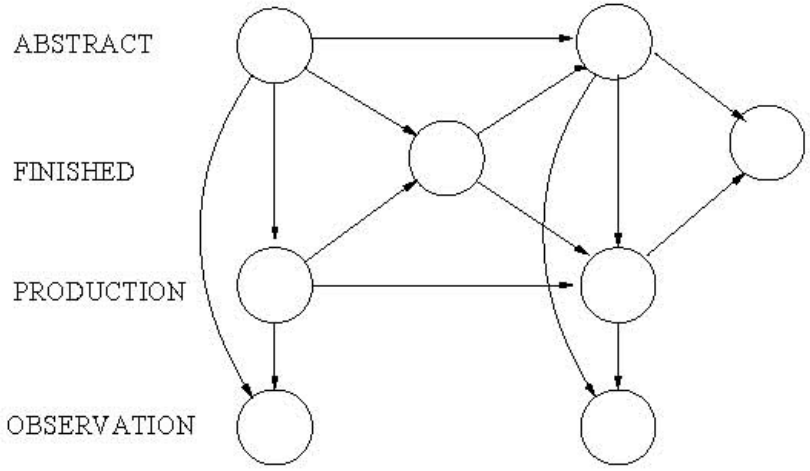
Hierarchies in HMMs

- Motivation
  - Model to naturally capture hierarchical structure
  - Fewer parameters: faster learning

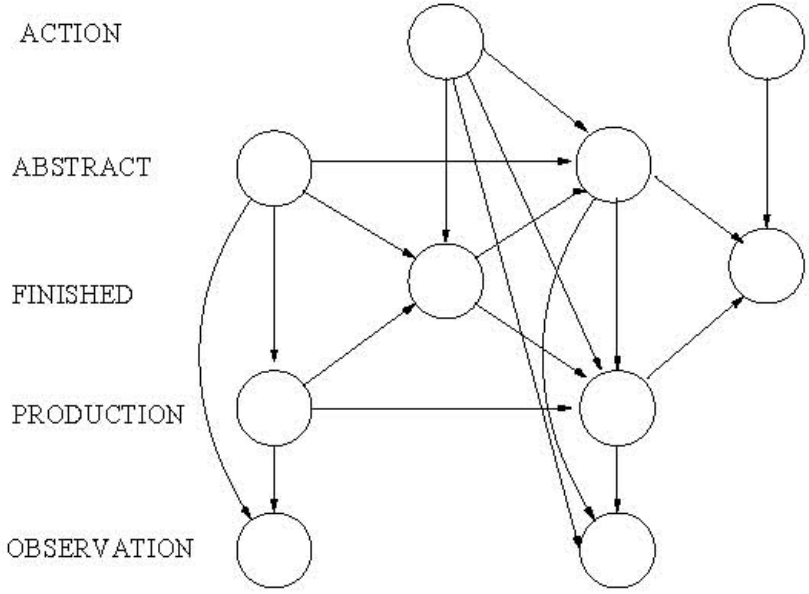
- Benefits
  - Spatial abstraction
  - Temporal abstraction

HHMMs as DBNs

- Murphy 2001
- DBN representation
- Equivalent parameterization



(a)



(b)

HHMMs as DBNs

- Factored representation
  - Can use any Bayes net algorithm
  - Common factored structure for all hierarchies

- Inference:
  - DBN: linear in time and quadratic in states
  - HHMM: cubic in time and linear in states

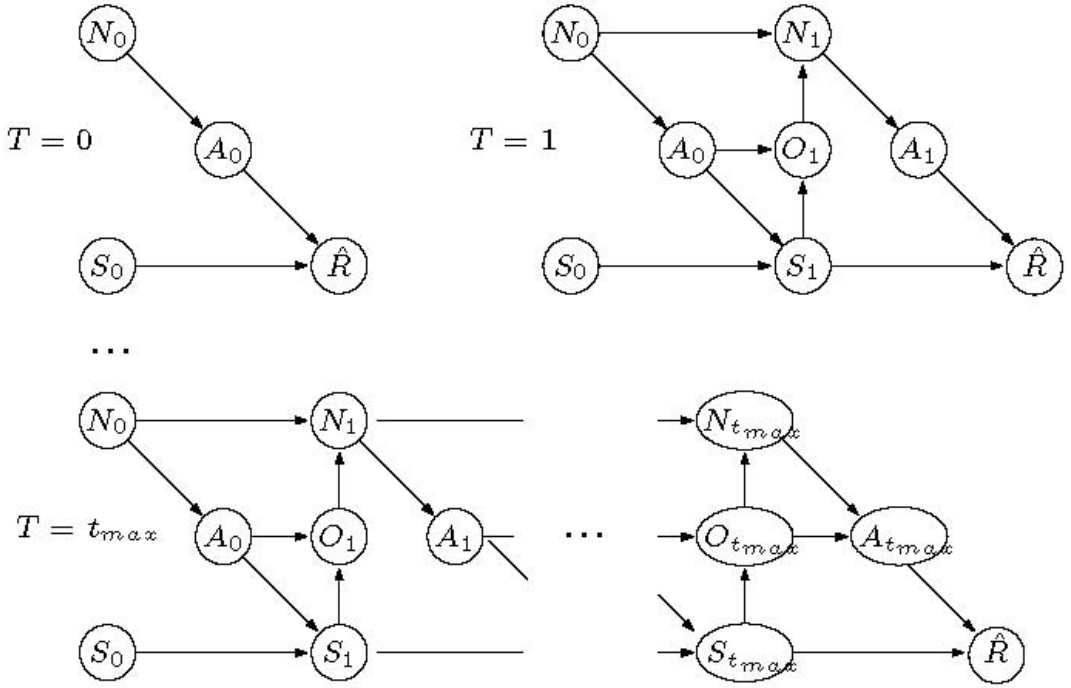
HPOMDPs

- Theocharous et al. 2001, 2004
- HHMM with actions and rewards

- Design or learn macro action for the exit state of each abstract state
- Question: do state hierarchy induce action hierarchies and vice-versa?

Controllers as DBNs

- Toussaint et al. 2006: controller ⇔ DBN mixture
  - Normalize reward in [0,1] to be a random variable
  - One DBN per horizon t with reward at the end
  - Pr(t) = t^{γ} (1 - γ)



Controllers as DBNs

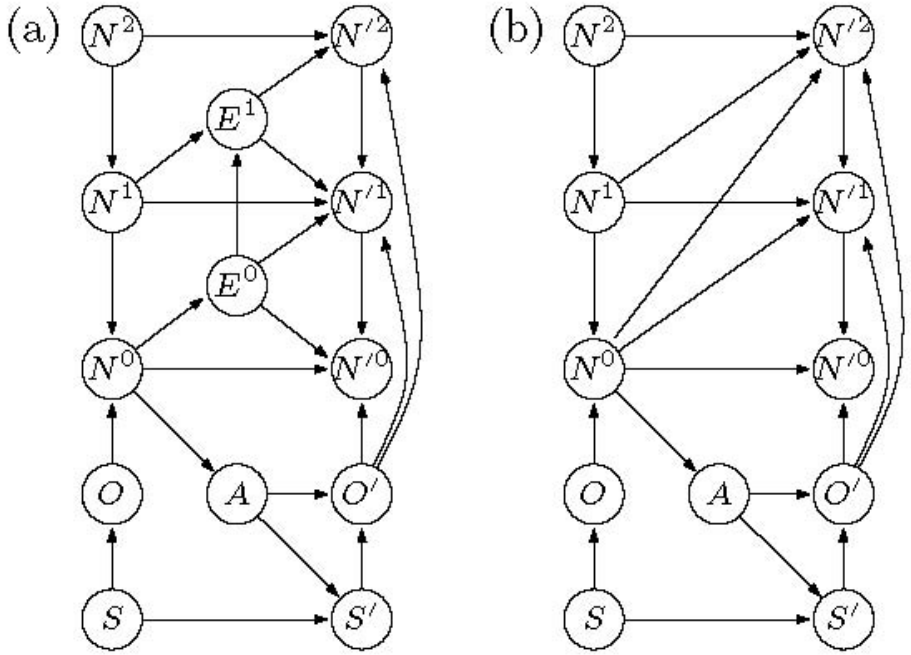
- Policy optimization
  - Maximize likelihood of R=1
  - Expectation maximization

- **Advantage: any inference algorithm can be used**

Factored Controllers

- Toussaint, Charlin & Poupart 2008
  - Hierarchical controllers ⇔ DBN mixture
  - More generally: factored controllers

Hierarchical controller



Factored controller

Factored Controllers

- Hierarchy discovery & policy optimization
  - Maximize likelihood of R=1
  - Expectation Maximization

- Same problem as non-convex quartic opt. prob.
  - EM much faster than optimization-based technique
  - But EM gets stuck in local optima more easily

Applications

- Nursebot project
  - polCA
- Robot navigation
  - HPOMDP
- Any other?

Summary

- Action hierarchies
- State hierarchies

- Advantages
  - Temporal and spatial abstraction
  - Fewer parameters to learn
  - Reuse of sub-policies/processes
  - Different search space

Questions?

- Are there synergies to be exploited by combining action and state hierarchies?
  - Does a state hierarchy imply an action hierarchy and vice versa?
- How much more abstraction can we gain from hierarchies over non-hierarchical state abstraction?