Defining Object Type Using MDP Homomorphisms

Alicia Peregrin Wolfe and Andrew G. Barto
pippin@cs.umass.edu, barto@cs.umass.edu

Autonomous Learning Laboratory
Department of Computer Science
University of Massachusetts, Amherst
Outline

- Introduction: Object Type
- CMP Homomorphisms
- Object Homomorphisms
- Object Options
- Subtypes
- Discussion
Are green blocks the same as yellow blocks?
Could the same policy be used to move both?
Is a block the same as a plate?
Is a block the same as a plate?

Can they be stacked the same way?
Related Work

- Givan, R., Dean, T., & Greig, M. *Equivalence Notions and Model Minimization in Markov Decision Processes*. Artificial Intelligence, 2003
  - stochastic bisimulation

  - MDP Homomorphisms

- CMP Homomorphisms (Wolfe, Barto, AAAI 2006)
  - If you are going to bother to build a model, use it for multiple tasks
Controlled Markov Processes

- Controlled Markov Process: \((S, A, T)\)
- \(S\): State set, \(A\): Action set, \(T: S \times A \times S \rightarrow [0, 1]\)
Controlled Markov Processes

- Controlled Markov Process: $(S, A, T)$
- $S$: State set, $A$: Action set, $T : S \times A \times S \rightarrow [0, 1]$
- Add output variable: $(S, A, T, y)$
- $y : S \rightarrow Y$
- Model which predicts one specific output variable
- Transitions occur between abstract states
- Can build policies for supported reward functions
  \( r \circ y \)
Partition of state and action spaces, with constraints:

\[ y(f(s), g_s(a)) = y(s, a) \]

\[ T(f(s_i), g_s(a), f(s_j)) = \sum_{s_k | f(s_j) = f(s_k)} T(s_i, a, s_k) \]
Partition of state and action spaces, with constraints:

\[
y(f(s), g_s(a)) = y(s, a)
\]

\[
T(f(s_i), g_s(a), f(s_j)) = \sum_{s_k | f(s_j) = f(s_k)} T(s_i, a, s_k)
\]
Partition of state and action spaces, with constraints:

\[
y(f(s), g_s(a)) = y(s, a)
\]

\[
T(f(s_i), g_s(a), f(s_j)) = \sum_{s_k | f(s_j) = f(s_k)} T(s_i, a, s_k)
\]
Object CMPs

- Output is $z \circ w_o$ where $w_o$ singles out object $o$, and $z$ singles out a feature
- What if multiple objects have the same model for $z$?
Object CMPs

- Output is $z \circ w_o$ where $w_o$ singles out object $o$, and $z$ singles out a feature.
- What if multiple objects have the same model for $z$?
Generalization

- Plates, blocks ∈ stackable objects type
- Only have to be the same with respect to the output variable
Generalization

- Plates, blocks ∈ stackable objects type
- Only have to be the same with respect to the output variable
Generalization

- Plates, blocks $\in$ stackable objects type
- Only have to be the same with respect to the output variable
Lifting Policies

- Policy specifies action in abstract model
Lifting Policies

- Policy specifies action in abstract model
- Reverse mapping to find the corresponding action in the CMP
Object Options

- Subgoal option:
  - reward function $r$
  - termination function $\beta$

- Object option: both are function of $z$

- Only need to find policies for types, not specific objects
What if all blue and green blocks stick to blocks of the same color, but yellow do not?

Sample states:
What if all blue and green blocks stick to blocks of the same color, but yellow do not?

Sample states:
What if all blue and green blocks stick to blocks of the same color, but yellow do not?

Sample states:
Object CMPs

- Equivalence criteria:
  - \( \forall \) CMPs \( M_k \)
  - \( h_i \) the reduction of \( M_k, z \circ w_{o_i} \)
  - \( \exists h_j, M_l, h_j \) a reduction of \( M_l, z \circ w_{o_j} \)
  - Such that \( h_i(M_k, z \circ w_{o_i}) = h_j(M_l, z \circ w_{o_j}) \)
  - Then \( o_j \preceq o_i \) under the output \( z \)
Discussion

- View environment from point of view of a single object
  - could be another agent
- Alternate method: add "pointer" to state space
  - one large model over all types
- HM framework does not generalize to more objects
  - Can’t use reduction for 3 blocks to learn about 4
  - Find the relations which will generalize from examples of reductions
  - Build a generic reduction