SMDP Homomorphisms: An Algebraic Approach to Abstraction in Semi-Markov Decision Processes

Balaraman Ravindran and Andrew G. Barto
Department of Computer Science
University of Massachusetts
Amherst, MA, U. S. A.
{ravi|barto}@cs.umass.edu

Abstract

To operate effectively in complex environments learning agents require the ability to selectively ignore irrelevant details and form useful abstractions. In this article we consider the question of what constitutes a useful abstraction in a stochastic sequential decision problem modeled as a semi-Markov Decision Process (SMDPs). We introduce the notion of SMDP homomorphism and argue that it provides a useful tool for a rigorous study of abstraction for SMDPs. We present an SMDP minimization framework and an abstraction framework for factored MDPs based on SMDP homomorphisms. We also model different classes of abstractions that arise in hierarchical systems. Although we use the options framework for purposes of illustration, the ideas are more generally applicable. We also show that the conditions for abstraction we employ are a generalization of earlier work by Dietterich as applied to the options framework.

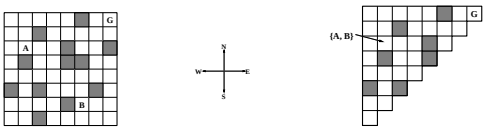
1 Introduction

The ability to form abstractions is one of the features that allows humans to operate effectively in complex environments. We systematically ignore information that we do not need for performing the immediate task at hand. While driving, for example, we may ignore details regarding clothing and the state of our hair. Researchers in artificial intelligence (AI), in particular machine learning (ML), have long recognized that applying computational approaches to operating in complex and real-world domains requires that we incorporate the ability to handle and form useful abstractions. In this article we consider the question of what constitutes a useful abstraction. Since this is a difficult problem when stated in general terms, much of the work in this field is specialized to particular classes of problems or specific modeling paradigms. In this work we will focus on Markov decision processes (MDPs), a formalism widely employed in modeling and solving stochastic sequential decision problems and semi-Markov decision processes (SMDPs), an extension to MDPs recently employed in modeling hierarchical systems [Sutton *et al.*, 1999; Dietterich, 2000; Parr and Russell, 1997].

Our objective in this article is to introduce the concept of an *SMDP homomorphism* and argue that it provides a unified view of key issues essential for a rigorous treatment of abstraction for stochastic dynamic decision processes. The concept of a homomorphism between dynamic systems, sometimes called a "dynamorphism" [Arbib and Manes, 1975], has played an important role in theories of abstract automata [Hartmanis and Stearns, 1966], theories of modeling and simulation [Zeigler, 1972], and it is frequently used by researchers studying model checking approaches to system validation [Emerson and Sistla, 1996]. Although those studying approximation and abstraction methods for MDPs and SMDPs have employed formalisms that implicitly embody the idea of a homomorphism, they have not made explicit use of the appropriate homomorphism concept. We provide what we claim is the appropriate concept and give examples of how it can be widely useful as the basis of the study of abstraction in a dynamic setting.

Informally, the kind of homomorphism we consider is a mapping from one dynamic system to another that eliminates state distinctions while preserving the system's dynamics. We present a definition of homomorphisms that is appropriate for SMDPs. In earlier work [Ravindran and Barto, 2002] we developed an MDP abstraction framework based on our notion of an MDP homomorphism. This framework extended the MDP minimization framework proposed by [Dean and Givan, 1997] and enabled the accommodation of redundancies arising from symmetric equivalence of the kind illustrated in Figure 1. While we can derive reduced models with a smaller state set by applying minimization ideas, we do not necessarily simplify the description of the problem in terms of the number of parameters required to describe it. But MDPs often have additional structure associated with them that we can exploit to develop compact representations. By injecting structure in the definition of SMDP homomorphism we can model abstraction schemes for structured MDPs. In this paper we present one simple example of such an abstraction scheme and mention other factored abstraction methods that can be modeled by suitable structured homomorphisms.

In the second part of the paper we extend the notion of SMDP homomorphism to hierarchical systems. In particular, we apply homomorphisms in the options framework introduced by [Sutton *et al.*, 1999] and show that this facilitates employing different abstractions at different levels of



(a) (b)

Figure 1: (a) A symmetric gridworld problem. The goal state is G and there are four deterministic actions. This gridworld is symmetric about the NE-SW diagonal. For example, states A and B are equivalent since for each action in A, there is an equivalent action in B. Taking action E, say, in state A is equivalent to taking action N in state B, in the sense that they go to equivalent states that are each one step closer to the goal. (b) An equivalent reduced model of the gridworld in (a). The states A and B in the original problem correspond to the single state $\{A, B\}$ in the reduced problem. A solution to this reduced gridworld can be used to derive a solution to the full problem.

a hierarchy. We also argue that homomorphisms allow us to model aspects of symbolic AI techniques in representing higher level task structure. Finally, we show that the SMDP homomorphism conditions generalize the "safe" state abstraction conditions for hierarchical systems introduced by [Dietterich, 2000].

After introducing some notation (Section 2), we define SMDP homomorphisms and discuss modeling symmetries (Section 3). Then we present a brief overview of our model minimization framework (Section 4), discuss abstraction in factored MDPs (Section 5) and abstraction in hierarchical systems (Section 6). We conclude with some discussion of directions for future research (Section 7).

2 Notation

A *Markov Decision Process* is a tuple $\langle S, A, \Psi, P, R\rangle$, where $S = \{1, 2, \cdots, n\}$ is a set of states, A is a finite set of actions, $\Psi \subseteq S \times A$ is the set of admissible state-action pairs, $P : \Psi \times S \to [0, 1]$ is the transition probability function with $P(s, a, s')$ being the probability of transition from state s to state s' under action a, and $R : \Psi \to \mathbb{R}$ is the expected reward function, with $R(s, a)$ being the expected reward for performing action a in state s. Let $A_s = \{a|(s, a) \in \Psi\} \subseteq A$ denote the set of actions admissible in state s. We assume that for all $s \in S$, A_s is non-empty.

A discrete time semi-Markov decision process (SMDP) is a generalization of an MDP in which actions can take variable amounts of time to complete. As with an MDP, an SMDP is a tuple $\langle S, A, \Psi, P, R\rangle$, where S, A and Ψ are the sets of states, actions and admissible state-action pairs; $P : \Psi \times S \times \mathbb{N} \to [0, 1]$ is the transition probability function with $P(s, a, s', N)$ being the probability of transition from state s to state s' under action a in N time steps, and $R : \Psi \times \mathbb{N} \to \mathbb{R}$ is the expected discounted reward function, with $R(s, a, N)$ being the expected reward for performing action a in state s and completing it in N time steps.[1]

A *stochastic policy*, π, is a mapping from Ψ to the real interval $[0, 1]$ with $\sum_{a\in A_s} \pi(s, a) = 1$ for all $s \in S$. For any $(s, a) \in \Psi$, $\pi(s, a)$ gives the probability of executing action a in state s. The *value* of a state-action pair (s, a) under policy π is the expected value of the sum of discounted future rewards starting from state s, taking action a, and following π thereafter. When the SMDP has well defined terminal states, we often do not discount future rewards. In such cases an SMDP is equivalent to an MDP and we will ignore the transition times. The *action-value function*, Q^π, corresponding to a policy π is the mapping from state-action pairs to their values. The solution of an MDP is an *optimal policy*, $\pi^\star$, that uniformly dominates all other possible policies for that MDP.

Let B be a partition of a set X. For any $x \in X$, $[x]_B$ denotes the block of B to which x belongs. Any function f from a set X to a set Y induces a partition (or equivalence relation) on X, with $[x]_f = [x']_f$ if and only if $f(x) = f(x')$.

3 SMDP Homomorphisms

A homomorphism from a dynamic system $\mathcal{M}$ to a dynamic system $\mathcal{M}'$ is a mapping that preserves $\mathcal{M}$'s dynamics, while in general eliminating some of the details of the full system $\mathcal{M}$. One can think of $\mathcal{M}'$ as a simplified model of $\mathcal{M}$ that is nevertheless a valid model of $\mathcal{M}$ with respect to the aspect's of $\mathcal{M}$'s state that it preserves. The specific definition of homomorphism that we claim is most useful for MDPs and SMDPs is as follows:

Definition: An *SMDP homomorphism* h from an SMDP $\mathcal{M} = \langle S, A, \Psi, P, R\rangle$ to an SMDP $\mathcal{M}' = \langle S', A', \Psi', P', R'\rangle$ is a surjection from Ψ to Ψ', defined by a tuple of surjections $\langle f, g_1, g_2, \cdots, g_n\rangle$, with $h((s, a)) = (f(s), g_s(a))$, where $f : S \to S'$ and $g_s : A_s \to A'_{f(s)}$ for $s \in S$, such that $\forall s, s' \in S, a \in A_s$ and for all $N \in \mathbb{N}$:

$$P'(f(s), g_s(a), f(s'), N) = \sum_{s''\in[s']_f} P(s, a, s'', N), \quad (1)$$

$$R'(f(s), g_s(a), N) = R(s, a, N). \quad (2)$$

We call $\mathcal{M}'$ the *homomorphic image* of $\mathcal{M}$ under h, and we use the shorthand $h(s, a)$ to denote $h((s, a))$. The surjection f maps states of $\mathcal{M}$ to states of $\mathcal{M}'$, and since it is generally many-to-one, it generally induces nontrivial equivalence classes of states s of M: $[s]_f$. Each surjection g_s recodes the actions admissible in state s of $\mathcal{M}$ to actions admissible in state $f(s)$ of $\mathcal{M}'$. This *state-dependent* recoding of actions is a key innovation of our definition, which we discuss in more detail below. Condition (1) says that the transition probabilities in the simpler SMDP $\mathcal{M}'$ are expressible as sums of the transition probabilities of the states of $\mathcal{M}$ that f maps to that same state in $\mathcal{M}'$. This is the stochastic version of the standard condition for homomorphisms of deterministic systems that requires that the homomorphism commutes with the system dynamics [Hartmanis and Stearns, 1966]. Condition (2) says that state-action pairs that have the same image under h have the same expected reward. An MDP homomorphism is similar to an SMDP homomorphism except that the conditions (1) and (2) apply only to the states and actions and not to the transition times.

The state-dependent action mapping allows us to model symmetric equivalence in MDPs and SMDPs. For example,

[1] We are adopting the formalism of [Dietterich, 2000].

if $h = \langle f, g_1, g_2, \cdots, g_n \rangle$ is a homomorphism from the gridworld of Figure 1(a) to that of Figure 1(b), then $f(A) = f(B)$ is the state marked $\{A, B\}$ in Figure 1(b). Also $g_A(E) = g_B(N) = E$, $g_A(W) = g_B(S) = W$, and so on. Whereas [Zinkevich and Balch, 2001] defined symmetries of MDPs by employing equivalence relations on the state-action pairs, we explicitly formalize the notion of SMDP symmetries employing SMDP homomorphisms and group theoretic concepts.

Definitions: An SMDP homomorphism $h = \langle f, g_1, g_2, \cdots, g_n \rangle$ from SMDP $\mathcal{M} = \langle S, A, \Psi, P, R \rangle$ to SMDP $\mathcal{M}' = \langle S', A', \Psi', P', R' \rangle$ is an *SMDP isomorphism* from $\mathcal{M}$ to $\mathcal{M}'$ if and only if f and g_s, $s \in S$, are bijective. $\mathcal{M}$ is said to be *isomorphic* to $\mathcal{M}'$ and vice versa. An SMDP isomorphism from an SMDP $\mathcal{M}$ to itself is an *automorphism* of $\mathcal{M}$.

The set of all automorphisms of an SMDP $\mathcal{M}$, denoted by $\mathrm{Aut}\mathcal{M}$, forms a group under composition of homomorphisms. This group is the *symmetry group* of $\mathcal{M}$. In the gridworld example of Figure 1, the symmetry group consists of the identity map on states and actions, a reflection of the states about the NE-SW diagonal and a swapping of actions N and E and of actions S and W. Any subgroup of the symmetry group of an SMDP induces an equivalence relation on Ψ, which can also be induced by a suitably defined homomorphism [Ravindran and Barto, 2001]. Therefore we can model symmetric equivalence as a special case of homomorphic equivalence.

4 Minimization

The notion of homomorphic equivalence immediately gives us an SMDP minimization framework. In [Ravindran and Barto, 2001] we extended the minimization framework of Dean and Givan [Dean and Givan, 1997; Givan *et al.*, 2003] to include state-dependent action recoding and showed that if two state-action pairs have the same image under a homomorphism, then they have the same optimal value. We also showed that when $\mathcal{M}'$ is a homomorphic image of an MDP $\mathcal{M}$, a policy in $\mathcal{M}'$ can *induce* a policy in $\mathcal{M}$ that is closely related. Specifically a policy that is optimal in $\mathcal{M}'$ can induce an optimal policy in $\mathcal{M}$. Thus we can solve the original MDP by solving a homomorphic image. It is easy to extend these results to SMDP models.

Thus the goal of minimization is to derive a *minimal image* of the SMDP, i.e., a homomorphic image with the least number of admissible state-action pairs. We also adapted an existing minimization algorithm to find minimal images employing state-action equivalence. Employing state-dependent action recoding allows us to achieve greater reduction in model size than possible with Dean and Givan's framework. For example, the gridworld in Figure 1(a) is minimal if we do not consider state-dependent action mappings.

5 Abstraction in Structured MDPs

SMDP homomorphisms can also be used to model various SMDP abstraction frameworks. The definition of homomorphism in Section 3 assumed a monolithic representation of the state set. While we can derive an equivalent MDP model with a smaller Ψ, it does not follow that the description of the state set is necessarily simpler and hence might not lead to a simpler problem representation. Many classes of problems that are modeled as MDPs have some inherent structure. We define structured forms of homomorphisms that can exploit this structure in deriving simpler problem representations.

Factored MDPs are a popular way to model structure in MDPs. A factored MDP is described, as before, by the tuple $\langle S, A, \Psi, P, R \rangle$. The state set S is now given by M features or variables, $S \subseteq \prod_{i=1}^{M} S_i$, where S_i is the set of permissible values for feature i. Thus any $s \in S$ is of the form $s = \langle s_1, \ldots, s_M \rangle$, where $s_i \in S_i$ for all i. The elements of S are still uniquely labeled 1 through n. The transition probabilities P are often described by a two-slice *temporal Bayesian network* (2-TBN) [Dean and Kanazawa, 1989]. A 2-TBN is a two layer directed acyclic graph, whose nodes are $\{s_1, \ldots, s_M\}$ and $\{s'_1, \ldots, s'_M\}$. Here s_i denotes the value of feature i in the present state and s'_i denotes the value of feature i in the resulting state. An arc from node s_i to node s'_j indicates that the s'_j depends on s_i. Many classes of structured problems may be modeled by a 2-TBN in which the arcs are restricted to go from nodes in the first set to those in the second. The state-transition probabilities can be factored as:

$$P(s, a, s') = \prod_{i=1}^{M} \text{Prob}(s'_i | \text{Pre}(s'_i, a)),$$

where Pre(s'_i, a) denotes the parents of node s'_i in the 2-TBN corresponding to action a and each of the Prob(s'_i|Pre(s'_i, a)) is given by a conditional probability table (CPT) associated with node s'_i. The reward function may be similarly represented.

Structuring the state space representation allows us to consider morphisms that are structured, i. e. surjections from one structured set to another. An example of a structured morphism is a simple projection onto a subset of features. Here the state set of the homomorphic image is described by a subset of the features, while the rest are ignored. We introduce some notation, after [Zeigler, 1972], to make the following definitions easier. Given a structured set $X \subseteq \prod_{i=1}^{M} X_i$, the *i-th projection* of X is a mapping $\rho_i : X \to X_i$, defined by $\rho_i(\langle x_1, \ldots, x_M \rangle) = x_i$. We extend this definition to that of a projection on a subset of features. Given a set $J \subseteq \{1, \ldots, M\}$ the *J-projection* of X is a mapping $\rho_J : X \to \prod_{j \in J} X_j$, defined by $\rho_J = \prod_{j \in J} \rho_j$.

Definition: A *simple projection homomorphism* h from a structured MDP $\mathcal{M} = \langle S, A, \Psi, P, R \rangle$ to a structured MDP $\mathcal{M}' = \langle S', A', \Psi', P', R' \rangle$ is a surjection from Ψ to Ψ', defined by a tuple of surjections $\langle f, g_1, g_2, \cdots, g_n \rangle$, with $h(s, a) = (f(s), g_s(a))$, where $f = \rho_F : S \to S'$, where $F \subseteq \{1, \ldots, M\}$ and $g_s : A_s \to A'_{f(s)}$ for $s \in S$, such that $\forall s, s' \in S, a \in A_s$:

$$P'(f(s), g_s(a), f(s')) = \prod_{j \in F} \text{Prob}(s'_j | \text{Pre}(s'_j, a)) \quad (3)$$

$$R'(f(s), g_s(a)) = R(s, a). \quad (4)$$

The first condition implies that the subset F should be chosen such that the features chosen are sufficient to describe the block transition dynamics of the system. In other words, the

subgraph of the 2-TBN described by the projection should include all the incoming arcs incident on the chosen nodes. The second condition requires that all the parents and the incoming arcs to the reward node are also included in the subgraph. To find such homomorphic projections, we just need to work back from the reward node including arcs and nodes, until we reach the desired subgraph. Such an algorithm will run in time polynomial in the number of features. It is evident that the space of such simple projections is much smaller than that of general maps and in general may not contain a homomorphism reducing a given MDP. Without suitable constraints, often derived from prior knowledge of the structure of the problem, searching for general structured homomorphisms results in a combinatorial explosion. Abstraction algorithms developed by Boutilier and colleagues can be modeled as converging to constrained forms of structured morphisms assuming various representations of the CPTs—when the space of morphisms is defined by boolean formulae of the features [Boutilier and Dearden, 1994], when it is defined by decision trees on the features [Boutilier *et al.*, 1995] and when it is defined by first-order logic formulae [Boutilier *et al.*, 2001].

6 Abstraction in Hierarchical Systems

In the previous section we showed that SMDP homomorphisms can model various abstraction schemes in "flat" MDPs and SMDPs. SMDP homomorphisms are a convenient and powerful formalism for modeling abstraction schemes in hierarchical systems as well. Before we explore various abstraction approaches we first introduce a hierarchical architecture that supports abstraction.

6.1 Hierarchical Markov Options

Recently several hierarchical reinforcement learning frameworks have been proposed [Parr and Russell, 1997; Sutton *et al.*, 1999; Dietterich, 2000] all of which use the SMDP formalism. In this article the hierarchical framework we adopt is the *options framework* [Sutton *et al.*, 1999], though the ideas developed here are more generally applicable. Options are actions that take multiple time steps to complete. They are usually described by: the policy to follow while the option is executing, the set of states in which the option can begin execution, and the termination function β which gives the probability with which the option can terminate in a given state. The resulting systems are naturally modeled as SMDPs with the transition time distributions induced by the option policies. We present an extension to the options framework that readily facilitates modeling abstraction at multiple levels of the hierarchy using SMDP homomorphisms.

We consider the class of options known as Markov options, whose policies satisfy the Markov property and that terminate on achieving a certain sub-goal. In such instances it is possible to implicitly define the option policy as the solution to an *option MDP*, or an option SMDP if the option has access to other options. Accordingly we define a hierarchical Markov sub-goal option:

Definition: A *hierarchical Markov sub-goal option* of an SMDP $\mathcal{M} = \langle S, A, \Psi, P, R\rangle$ is the tuple $O = \langle \mathcal{M}_O, \mathcal{I}, \beta\rangle$, where $\mathcal{I} \subseteq S$ is the initiation set of the option, $\beta : S \to [0, 1]$, is the termination function and $\mathcal{M}_O$ is the option SMDP.

The state set of $\mathcal{M}_O$ is a subset of S and constitutes the *domain* of the option. The action set of M_O is a subset of A and may contain other options as well as "primitive" actions in A. The reward function of $\mathcal{M}_O$ is chosen to reflect the sub-goal of the option. The transition probabilities of $\mathcal{M}_O$ are induced by P and the policies of lower level options. We assume that the lower-level options are following fixed policies which are optimal in the corresponding option SMDPs. The option policy π is obtained by solving $\mathcal{M}_O$, treating it as an episodic task with the possible initial states of the episodes given by $\mathcal{I}$ and the termination of each episode determined by the option's termination function β.

For example, in the gridworld task shown in Figure 2(a), an option to pick up the object and exit room 1 can be defined as the solution to the problem shown in 2(b), with a suitably defined reward function. The domain and the initiation set of the option are all states in the room and the option terminates on exiting the room with or without the object.

6.2 Option Specific Abstraction

The homomorphism conditions (1) and (2) are very strict and frequently we end up with trivial homomorphic images when deriving abstractions based on a non-hierarchical SMDP. But it is often possible to derive non-trivial reductions if we restrict attention to certain sub-problems, i.e., certain sub-goal options. In such cases we can apply the ideas discussed in Sections 4 and 5 to an option SMDP directly to derive abstractions that are specific to that option. The problem of learning the option policy is transformed to the usually simpler problem of learning an optimal policy for the homomorphic image.

6.3 Relativized Options

Consider the problem of navigating in the gridworld environment shown in Figure 2(a). The goal is to reach the central corridor after collecting all the objects in the environment. No non-trivial homomorphic image exists of the entire problem. But there are many similar components in the problem, namely, the five sub-tasks of getting the object and exiting $room_i$. We can model these similar components by a "partial" homomorphic image—where the homomorphism conditions are applicable only to states in the rooms. One such partial image is shown in Figure 2(b). Employing such an abstraction lets us compactly represent a related family of options, in this case the tasks of collecting objects and exiting each of the five rooms, using a single option MDP. We refer to this compact option as a *relativized option*. Such abstractions are an extension of the notion of relativized operators introduced by [Iba, 1989]. Formally we define a relativized option as follows:

Definition: A *relativized option* of an SMDP $\mathcal{M} = \langle S, A, \Psi, P, R\rangle$ is the tuple $O = \langle h, \mathcal{M}_O, \mathcal{I}, \beta\rangle$, where $\mathcal{I} \subseteq S$ is the initiation set, $\beta : S' \to [0, 1]$ is the termination function and $h = \langle f, g_1, g_2, \cdots, g_n\rangle$ is a partial homomorphism from the SMDP $\langle S, A, \Psi, P, R_O\rangle$ to the option SMDP $\mathcal{M}_O$ with R_O chosen based on the sub-task.

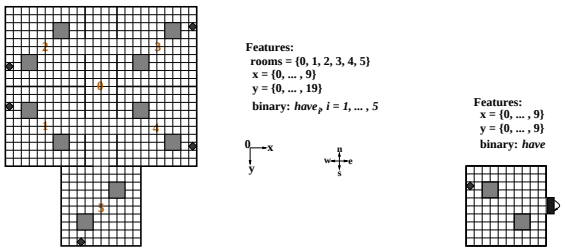


Figure 2: (a) A simple rooms domain with similar rooms and usual stochastic gridworld dynamics. The task is to collect all 5 objects (black diamonds) in the environment and reach the central corridor. The shaded squares are obstacles. (b) The option MDP corresponding to a *get-object-and-leave-room* option.

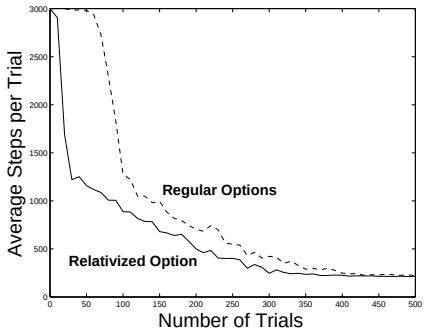


Figure 3: Comparison of performance of learning agents employing regular and relativized options on the task shown in Figure 2.

Here the state set of $\mathcal{M}_O$ is $S' = f(S_O)$, where S_O is the domain of the option, and the admissible state-action set is $h(\Psi)$. Going back to our example in Figure 2(a), we can now define a single *get-object-and-leave-room* relativized option using the option MDP of Figure 2(b). The policy learned in this option MDP can then be suitably lifted to $\mathcal{M}$ to provide different policy fragments in the different rooms. Figure 3 demonstrates the speed-up in performance when using a single relativized option as opposed to five regular options. In this experiment the option policies and the higher level policy were learned simultaneously.[2]

6.4 Modeling Higher Level Structure

SMDP homomorphisms have the power to model a broader class of abstractions, those not supported by the base level dynamics of the system but which can be induced at some intermediate levels of the hierarchy. For example, consider a multi-link assembly line robot arm that needs to move objects from one assembly line to another. The state of the system is described by the joint angles and velocities, *object-shape*, *object-orientation*, and Boolean variables indicating whether the object is *firmly-grasped* and whether the object is *at-target-location*. The primitive actions are various joint torques. Depending on the configuration of the arm and the shape and orientation of the object, this task requires different sequences of actions, or policies, even though conceptually the higher level task has the same "structure"—grasp object, move object, and place object.

We can model this higher level task structure using a relativized option. First, we define suitable grasp-object options, such as *grasp-cylinder*, *grasp-block* etc. , and similar options for moving and placing objects. Then we form a partial homomorphic image of the resulting SMDP—the state set of the image is described by the two Boolean features and the action set consists of *grasp*, *move* and *place* actions. The partial homomorphism, which applies only to the admissible state-option pairs of the SMDP, consists of $f = \rho_J$, where $J = \{$ *firmly-grasped*, *at-target-location* $\}$ and $g_{\langle \cdot, object-shape \rangle}$(*grasp-object-shape*) = *grasp* for all *object-shape* and similarly for the *move-object* and *place-object* options. A relativized option with this partial image and partial homomorphism as the option MDP and option homomorphism, respectively, captures the desired conceptual structure. Executing the optimal policy for this option results in action sequences of the form: *grasp*, *move*, *place*, ... Depending on the *object-shape* these abstract actions get bound to different lower level options. While techniques in symbolic AI have long been able to model such higher level task structure, the reinforcement learning community lacked an obvious mechanism to do the same. Our work gives us new tools to model conceptual task structure at various temporal and spatial scales.

6.5 Relation to MaxQ Safe State-abstraction

[Dietterich, 2000] introduced safe state-abstraction conditions for the MaxQ architecture, a hierarchical learning framework related to the options framework. These conditions ensure that the resulting abstractions do not result in any loss of performance. He assumes that the sub-problems at different levels of the hierarchy are specified by factored MDPs. The MaxQ architecture employs a form of value function decomposition and some of the safe abstraction conditions apply only to this form of decomposition. The following condition is more universal and applies to the hierarchical Markov options framework as well:

Definition: A Projection ρ_J is *safe* if: (i) for all (s, a) in Ψ and s' in S, $P(s, a, s', N) = Prob(\rho_J(s'), N|\rho_J(s), a) \times Prob(\rho_{M-J}(s')|s, a)$, and (ii) for all $(s, a), (t, a)$ in Ψ, if $\rho_J(s) = \rho_J(t)$, then $R(s, a, N) = R(t, a, N)$.

Condition (i) states that the transition probability can be expressed as a product of two probabilities, one of which describes the evolution of a subset of the features and depends only on that subset. Condition (ii) states that if two states project to the same abstract state, then they have the same immediate reward. From equations 1-4, it is evident that the above conditions are equivalent to the SMDP homomorphism conditions if we restrict our attention to simple projection homomorphisms and do not consider action remapping. Thus the SMDP homomorphism conditions introduced here generalize Dietterich's safe state-abstraction condition as applicable to the hierarchical Markov option framework.

[2] In [Ravindran and Barto, 2002] we have reported more detailed experiments in this setting.

7 Discussion

The equivalence classes induced by SMDP homomorphisms satisfy the stochastic version of the substitution property [Hartmanis and Stearns, 1966]. This property is also closely related to *lumpability* in Markov chains [Kemeny and Snell, 1960] and *bisimulation homogeneity* [Givan *et al.*, 2003] in MDPs. We chose the SMDP homomorphism as our basic formalism because we believe that it is a simpler notion and provides a more intuitive explanation of various abstraction schemes.

The homomorphism conditions (1) and (2) are very strict conditions that are often not met exactly in practice. One approach is to relax the homomorphism conditions somewhat and allow small variations in the block transition probabilities and rewards. We have explored this issue [Ravindran and Barto, 2002], basing our approximate homomorphisms on the concept of *Bounded-parameter MDPs* developed by [Givan *et al.*, 2000]. We are currently working on extending approximate homomorphisms to hierarchical systems so as to accommodate variations in transition-time distributions.

Although SMDP homomorphisms are powerful tools in modeling abstraction, finding a minimal image of a given SMDP is an NP-hard problem. While taking advantage of structure allows us to develop efficient algorithms in special cases, much work needs to be done to develop efficient general purpose algorithms. Currently we are investigating methods that allow us to determine homomorphisms given a set of candidate transformations in a hierarchical setting.

In this article we presented a novel definition of SMDP homomorphism that employs state-dependent recoding of actions. This allows us to extend existing minimization and abstraction methods to a richer class of problems. We developed notions of equivalence specialized to factored representations and hierarchical architectures. We have shown that SMDP homomorphism can serve as the basis for modeling a variety of abstraction paradigms.

Acknowledgments

We wish to thank Dan Bernstein and Mike Rosenstein for many hours of useful discussion and Bob Givan and Matt Greig for clarifying certain ideas from their work. This material is based upon work supported by the National Science Foundation under Grant No. ECS-0218125 to Andrew G. Barto and Sridhar Mahadevan. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[Arbib and Manes, 1975] M. A. Arbib and E. G. Manes. *Arrows, Structures and Functors*. Academic Press, New York, NY, 1975.

[Boutilier and Dearden, 1994] C. Boutilier and R. Dearden. Using abstractions for decision theoretic planning with time constraints. In *Proceedings of the AAAI-94*, pages 1016–1022. AAAI, 1994.

[Boutilier *et al.*, 1995] C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy construction. In *Proceedings of International Joint Conference on Artificial Intelligence 14*, pages 1104–1111, 1995.

[Boutilier *et al.*, 2001] Craig Boutilier, Ray Reiter, and Robert Price. Symbolic dynamic programming for first-order mdps. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 541–547, 2001.

[Dean and Givan, 1997] T. Dean and R. Givan. Model minimization in Markov decision processes. In *Proceedings of AAAI-97*, pages 106–111. AAAI, 1997.

[Dean and Kanazawa, 1989] Thomas Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computer Intelligence*, 5(3):142–150, 1989.

[Dietterich, 2000] T. G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Artificial Intelligence Research*, 13:227–303, 2000.

[Emerson and Sistla, 1996] E. A. Emerson and A. P. Sistla. Symmetry and model checking. *Formal Methods in System Design*, 9(1/2):105–131, 1996.

[Givan *et al.*, 2000] R. Givan, S. Leach, and T. Dean. Bounded-parameter Markov decision processes. *Artificial Intelligence*, 122:71–109, 2000.

[Givan *et al.*, 2003] R. Givan, T. Dean, and M. Greig. Equivalence notions and model minimization in Markov decision processes. To appear in Artificial Intelligence, 2003.

[Hartmanis and Stearns, 1966] J. Hartmanis and R. E. Stearns. *Algebraic Structure Theory of Sequential Machines*. Prentice-Hall, Englewood Cliffs, NJ, 1966.

[Iba, 1989] Glenn A. Iba. A heuristic approach to the discovery of macro-operators. *Machine Learning*, 3:285–317, 1989.

[Kemeny and Snell, 1960] J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Van Nostrand, Princeton, NJ, 1960.

[Parr and Russell, 1997] Ronald Parr and Stuart Russell. Reinforcement learning with hierarchies of machines. In *Proceedings of Advances in Neural Information Processing Systems 10*, pages 1043–1049. MIT Press, 1997.

[Ravindran and Barto, 2001] B. Ravindran and A. G. Barto. Symmetries and model minimization of Markov decision processes. Technical Report 01-43, University of Massachusetts, Amherst, 2001.

[Ravindran and Barto, 2002] Balaraman Ravindran and Andrew G. Barto. Model minimization in hierarchical reinforcement learning. In Sven Koenig and Robert C. Holte, editors, *Proceedings of the Fifth Symposium on Abstraction, Reformulation and Approximation (SARA 2002), Lecture Notes in Artificial Intelligence 2371*, pages 196–211, New York, NY, August 2002. Springer-Verlag.

[Sutton *et al.*, 1999] Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and Semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[Zeigler, 1972] Bernard P. Zeigler. On the formulation of problems in simulation and modelling in the framework of mathematical system theory. In *Proceedings of the Sixth International Congress on Cybernetics*, pages 363–385. Association Internationale de Sybernétique, 1972.

[Zinkevich and Balch, 2001] M. Zinkevich and T. Balch. Symmetry in Markov decision processes and its implications for single agent and multi agent learning. In *Proceedings of the 18th International Conference on Machine Learning*, pages 632–640, San Francisco, CA, 2001. Morgan Kaufmann.