

# Dynamic probabilistic logic models for effective abstractions in RL

Harsha Kokel,<sup>1</sup> Arjun Manoharan,<sup>2</sup> Sriraam Natarajan,<sup>1</sup>  
Balaraman Ravindran,<sup>2</sup> Prasad Tadepalli<sup>3</sup>

<sup>1</sup>The University of Texas at Dallas,

<sup>2</sup>Robert Bosch Centre for Data Science and Artificial Intelligence at Indian Institute of Technology Madras,

<sup>3</sup>Oregon State University

{hkobel,Sriraam.Natarajan}@utdallas.edu, {arjunman,ravi}@cse.iitm.ac.in, tadepall@eecs.oregonstate.edu

## Abstract

State abstraction enables sample-efficient learning and better task transfer in complex reinforcement learning environments. Recently, we proposed RePreL (Kokel et al. 2021), a hierarchical framework that leverages a relational planner to provide useful state abstractions for learning. We present a brief overview of this framework and the use of a **dynamic probabilistic logic model** to design these state abstractions. Our experiments show that RePreL not only achieves better performance and efficient learning on the task at hand but also demonstrates better generalization to unseen tasks.

## 1 Introduction

Planning and Reinforcement Learning have been two major thrusts of AI aimed at sequential decision making. We recently presented (Kokel et al. 2021) an integrated architecture we call “RePreL,” which combines relational planning (RP) and reinforcement learning (RL) in a way that exploits their complementary strengths and not only speeds up the convergence compared to a traditional RL solution but also enables effective transfer of the solutions over multiple tasks. In many real world domains, e.g., driving, the state space of offline planning is rather different from the state space of online execution. Planning typically occurs at the level of deciding the route, while online execution needs to take into account dynamic conditions such as locations of other cars and traffic lights. The agent typically does not have access to the dynamic part of the state at the planning time, e.g., future locations of other cars, nor does it have the computational resources to plan an optimal policy in advance that works for all possible traffic events.

The key principle that enables agents to deal with these informational and computational challenges is *abstraction*. In the driving example, the high level state space consists of coarse locations such as “O’hare airport” and high level actions such as take “Exit 205,” while the lower level state space consists of a more precise location and velocity of the car and actions such as turning the steering wheel by some amount and applying brakes. Importantly, excepting occasional unforeseen failures, the two levels operate independently of each other and depend on different kinds of information available at different times. This allows the agent to

tractably plan at a high level without needing to know the exact state at the time of the execution, and behave appropriately during plan execution by only paying attention to a small dynamic part of the state.

To achieve the integration of planner and RL effectively, we adapt first-order conditional influence (FOCI) statements (Natarajan et al. 2008) to specify bisimilarity conditions of MDPs (Givan, Dean, and Greig 2003), which in turn help justify safe and effective abstractions for reinforcement learning (Dietterich 2000b). This is the key contribution of this extended abstract – *highlighting the need for an effective StaRAI representation in defining appropriate abstractions for relational MDPs*. We hypothesize and verify empirically that such effective abstractions can result in efficient learning and in many cases, efficient and effective transfer.

## 2 Preliminaries

We define goal-directed relational MDP (GRMDP) as  $\langle S, A, P, R, \gamma, G \rangle$ , which is an extension of RMDP definition of Fern, Yoon, and Givan (2006) for goal-oriented domains by adding a set of goals  $G$  that the agent may be asked to achieve. A problem instance for a GRMDP is defined by a pair  $\langle s \in S, g \in G \rangle$ , where  $s$  is a state and  $g$  is a goal condition, both represented using sets of literals, i.e., positive and/or negative atoms. A solution is a policy that starts from  $s$  and ends in a state satisfying  $g$  with probability 1.

The RePreL framework proposes that the GRMDPs can be solved using a combination of planning and RL in 3 stages:

1. **Planning:** Use the hierarchical planner to decompose the goal of the GRMDP to smaller sub-tasks.
2. **Abstraction:** Get abstract state representations specific to the sub-task.
3. **RL:** Learn RL agents to perform these sub-tasks in abstract state space.

We defer to Kokel et al. (2021) for details of the RePreL framework; here we focus on how to obtain the task-specific abstract state representations using StaRAI approach.

## 3 Abstraction

Dietterich (2000a) defines *irrelevant state variables* for an MDP as variables that never influence the reward function or the relevant state variables. Formally,

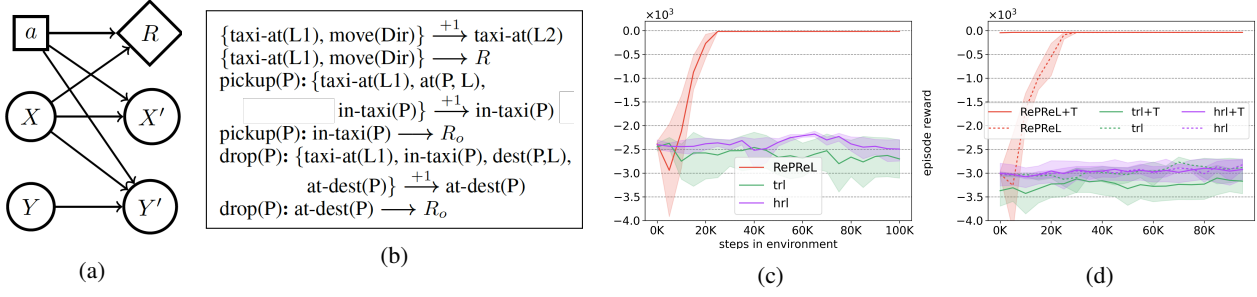


Figure 1: **(a)** A DBN representing influences in an MDP; **(b)** D-FOCI statements representing influences in a GRMDP of the Taxi domain; **(c)** Learning curve of RePreL, Taskable RL (trl), and Hierarchical RL (hrl) for task 1, transport 1 passenger, in the Taxi domain; **(d)** Learning curve with-transfer (solid lines) and without transfer (dashed lines) for task 2, transport 2 passengers, in the Taxi domain.  $X$ -axis represents steps taken in environments and  $Y$ -axis represents average episode reward.

**Definition 1.** State variables  $Y$  are irrelevant in an MDP, if state variables can be partitioned into two disjoint subsets  $X$  and  $Y$  such that

1.  $P(x', y' | x, y, a) = P(x' | x, a)P(y' | x, y, a)$
2.  $R(s, a, s') = R(\langle x, y \rangle, a, \langle x', y' \rangle) = R(x, a, x')$

Given a Dynamic Bayesian Network (DBN) representation of the transition function of an MDP, like shown in the Fig. 1a, the set of irrelevant variables can be identified by starting at the reward variable and collecting all the variables that influence the collected variables. Note that the *transition function of a GRMDP is first-order* and hence a simple DBN does not suffice. Previous work have used a 2-time slice Probabilistic Relational Model (Guestrin et al. 2003) to capture the transition function in relational MDP. However, they are not suitable for capturing GRMDPs. Hence, we use First-Order Conditional Influence (FOCI) statements (Natarajan et al. 2008). Each FOCI statement is of the form: “if condition then  $X_1$  influence  $X_2$ ”, where, condition and  $X_1$  are a set of first-order literals and  $X_2$  is a single literal. It encodes the information that literal  $X_2$  is influenced only by the literals in  $X_1$  when the stated condition is satisfied. The influence information captured by the FOCI statements is similar to that captured in many SRL models (Getoor and Taskar 2007) and additionally FOCI statements allow expressing conjunction of literals as conditions.

For GRMDPs, we develop Dynamic FOCI (D-FOCI) statements that express the influence of the literals in the current time step on the literals in the next time-step. Template of the D-FOCI statement is:

$$\text{sub-task} : X_1 \xrightarrow{+1} X_2$$

To distinguish the same time-step and next time-step influences and to capture the temporal influence, we use a +1 on the arrow. D-FOCI statement encodes the information that for a given sub-task, the literal  $X_2$  in the next time-step is directly influenced only by the literals  $X_1$  in current time-step. Following the standard DBN representation of MDP, we allow action variables and the reward variables in the two sets of literals. To represent unconditional influences between state literals, we skip the sub-task.

The D-FOCI statements can be viewed as 2 timeslice PRM and have a similar function of capturing the conditional independence relationships between domain predicates at different time steps, with additional capability of conditioning on sub-task being executed. The Fig. 1b illustrates the D-FOCI statements in a relational Taxi domain with two sub-task: `pickup` and `drop`. A task-specific model-agnostic abstraction is derived for each sub-task by iterating through the D-FOCI statements and collecting the relevant literals that influence the relevant literals starting with the literals that influence reward variables  $R$  and  $R_o$ . The Theorem 1 in Kokel et al. (2021) shows that *if the GRMDP satisfies the D-FOCI statements with any fixed depth unrolling, then the corresponding model-agnostic abstraction has the same optimal value function as the fully instantiated MDP*. We refer to the paper for details.

## 4 Evaluation

Our empirical evaluations on 4 domains show that the proposed task-specific abstraction using the D-FOCI statements have three advantages: **1.** With abstract state representation, the state space is reduced and hence RePreL achieves better sample efficiency over other methods, **2.** With task-specific abstractions, RePreL demonstrates efficient transfer across task, **3.** With the propositionalization for D-FOCI statements, RePreL illustrates zero-shot generalization capability in some cases. We reproduce the average episode rewards achieved by various agents in the relational Taxi domain on two tasks in Fig. 1c and 1d. The policies which are transferred from one task to other are indicated with ‘+T’. As seen, RePreL agents needs significantly less number of samples to reach optimal reward and also show zero-shot generalization across tasks.

## 5 Discussion

We introduced Dynamic First-Order Conditional Influence statement, an extension of a StaRAI language to capture the conditional influence information of goal-directed relational MDP. We provide an approach for extracting effective task-specific state abstractions in relational MDPs. We demonstrate its utility on learning how to act in relational domains.

## 6 Acknowledgements

This work is supported by ARO award W911NF2010224 (HK & SN), AFOSR award FA9550-18-1-0462 (SN), travel grant from RBCDSAI (AM), DARPA contract N66001-17-2-4030 (PT) and NSF grant IIS-1619433 (PT). Any opinions, findings, conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the ARO, AFOSR, NSF, DARPA or the US government.

## References

- Dietterich, T. G. 2000a. An overview of hierarchical reinforcement learning. In *International Symposium on Abstraction, Reformulation, and Approximation*. Springer.
- Dietterich, T. G. 2000b. State abstraction in MAXQ hierarchical reinforcement learning. In *NeurIPS*, 994–1000.
- Fern, A.; Yoon, S.; and Givan, R. 2006. Approximate policy iteration with a policy language bias: Solving relational Markov decision processes. *JAIR* 25: 75–118.
- Getoor, L.; and Taskar, B. 2007. *Introduction to Statistical Relational Learning*. The MIT Press.
- Givan, R.; Dean, T.; and Greig, M. 2003. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence* 147(1-2): 163–223.
- Guestrin, C.; Koller, D.; Gearhart, C.; and Kanodia, N. 2003. Generalizing plans to new environments in relational MDPs. In *IJCAI*, 1003–1010.
- Kokel, H.; Manoharan, A.; Natarajan, S.; Balaraman, R.; and Tadepalli, P. 2021. RePREL: Integrating Relational Planning and Reinforcement Learning for Effective Abstraction. *ICAPS* 31(1): 533–541.
- Natarajan, S.; Tadepalli, P.; Dietterich, T. G.; and Fern, A. 2008. Learning first-order probabilistic models with combining rules. *Ann. Math. Artif. Intell.* 54(1-3): 223–256.