# RePReL : Integrating Relational Planning and Reinforcement Learning for Effective Abstraction (Extended Abstract)

**Harsha Kokel,**[1] **Arjun Manoharan,**[2] **Sriraam Natarajan,**[1]
**Balaraman Ravindran,**[2] **Prasad Tadepalli** [3]
[1]The University of Texas at Dallas,
[2]Robert Bosch Centre for Data Science and Artificial Intelligence at Indian Institute of Technology Madras,
[3]Oregon State University
{hkokel,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. Inspired by the benefits of state abstraction in hierarchical planning and learning, we propose RePReL, a hierarchical framework that leverages a relational planner to provide useful state abstractions for learning. State abstraction is especially beneficial in relational settings, where the number and/or types of objects are not fixed apriori. Our experiments show that RePReL framework not only achieves better performance and efficient learning on the task at hand but also demonstrates better generalization to unseen tasks. It has been argued that for human-level general intelligence, the ability to detect compositional structure in the domain (Lake et al. 2017) and form task-specific abstractions (Konidaris 2019) is necessary. Our RePReL framework takes a step in that direction by formalizing the prior domain knowledge that gives rise to effective task-specific abstractions.[1]

## 1 Introduction

Planning and Reinforcement Learning have been two major thrusts of AI aimed at sequential decision making. While classical relational planning focuses on composing sequences of high level actions offline before any execution, reinforcement learning interleaves planning and execution and is typically associated with reactive domains with unknown dynamics. We describe 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. Indeed, 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.

The key contribution of the current paper is the RePReL architecture, which consists of a high level relational planner and a low level reinforcement learner. The high level planner is itself hierarchical that allows it to further take advantage of multi-level abstractions. It plans to achieve its goal using a sequence of subgoals, which are passed onto the reinforcement learning agent. The reinforcement learning agent then tries to reach its assigned sub-goal with minimum path cost. To do this 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 2000; Ravindran and Barto 2003).

## 2 Relational Planning and Reinforcement Learning

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

[1] This is an extended abstract of Kokel et al. (2021) available at https://starling.utdallas.edu/papers/RePReL.

to achieve. The reward function $R$ provides the reward (or cost) of taking a step in the environment, regardless of the goal. 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.

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 tasks.
2. **Abstraction:** Get task specific abstractions.
3. **RL:** Learn RL agents to perform these tasks in abstract state space

**Planning:** The hierarchical planner decomposes goals into subgoals recursively to generate a sequence of planning operators. *A key difference to typical hierarchical planners is that in our case, planning operators do not execute the atomic action. Instead, these operators are in turn implemented as an RL agent that learns to solve them by executing a policy.* Several prior works have explored similar idea of combining a planner and RL agents to solve complex problems which have some notion of temporally extended actions or task hierarchies (Grounds and Kudenko 2005; Yang et al. 2018; Lyu et al. 2019; Jiang et al. 2019; Eppe, Nguyen, and Wermter 2019; Illanes et al. 2020). Our RePReL framework diverges from previous work in two ways: first, we use the relational MDP representation expressed above, and second, we propose an approach to define task-specific state abstractions, an important contribution of this work.

**Abstraction:** Safe and efficient state abstraction techniques have been studied extensively in RL (Li, Walsh, and Littman 2006). They have been particularly useful for multi-task and transfer learning problems (Walsh, Li, and Littman 2006; Sorg and Singh 2009; Abel et al. 2018). We are inspired by the task-specific abstractions of MAXQ (Dietterich 2000) and adopt the bisimulation framework of Givan, Dean, and Greig (2003) and Ravindran and Barto (2003), which has been called "model agnostic abstraction" in Li, Walsh, and Littman (2006). An abstraction function is called model-agnostic when the the immediate reward distribution and the transition dynamics of the abstract MDP are the same as that of the original MDP [2].

In RMDPs, we need to reason about how the actions influence the state predicates and how rewards are influenced by goal predicates and actions to decide which literals should be included and excluded in the abstraction. We capture this knowledge using First-Order Conditional Influence (FOCI) statements (Natarajan et al. 2008), one of the many variants of statistical relational learning languages (Getoor and Taskar 2007; Raedt et al. 2016). Each FOCI statement is of the form: "if `condition` then $\mathtt{X}_1$ influence $\mathtt{X}_2$", where, `condition` and $\mathtt{X}_1$ are a set of first-order literals and $\mathtt{X}_2$ is a single literal. It encodes the information that literal $\mathtt{X}_2$ is influenced only by the literals in $\mathtt{X}_1$ when the stated `condition` is satisfied. For RePReL, we simplify the syntax and extend FOCI to dynamic FOCI (D-FOCI) statements. In addition to direct influences in the same time step, D-FOCI statements also describe the direct influences of the literals in the current time step on the literals in the next time step. To distinguish the two kinds of influences, we show a $+1$ on the arrow between the sets of literals to capture a temporal interaction, as shown below.

$$\mathtt{operator}: \{\mathrm{p}(X_1), q(X_1)\} \xrightarrow{+1} \mathrm{q}(X_1)$$

It says that, for the given `operator`, the literal $q(\mathtt{X}_1)$ in the next time step is directly influenced only by the literals $\{p(\mathtt{X}_1), q(\mathtt{X}_1)\}$. 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 predicates, we skip the `operator`.

The D-FOCI statements can be viewed as relational versions of dynamic Bayesian networks (DBNs) and have a similar function of capturing the conditional independence relationships between domain predicates at different time steps. While the planner works in relational representations, the reinforcement learning operates at a propositional level. The propositionalization proceeds by instantiating each D-FOCI statement with generic objects yielding a structure equivalent to a propositional DBN. A model-agnostic abstraction is derived for each operator by iteratively adding the literals that influence the relevant literals through all actions starting with the reward variables.Theorem 1 in Kokel et al. (2021) shows that *if the MDP 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*.

**RL:** The hierarchical planner assumes a relational deterministic model of operators, whereas the reinforcement learner allows stochastic actions. Learning of the RL agents for each planning operator can thus proceed with abstract state representation with guarantees of optimality.

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 FOCI statements, RePReL illustrates zero-shot generalization capability in some cases.

## 3 Conclusions

We presented a framework that seamlessly combines planning and RL. The key intuition is to employ the combination of a planner and a relational language to define task-specific abstractions that can be effectively and efficiently exploited by a RL agent. Our empirical results across a variety of tasks demonstrates the efficacy and generalization capabilities of the proposed approach. Extending the work to continuous state-action spaces, allowing for richer human interaction and scaling up to large tasks remain interesting future directions.

[2] cf. Kokel et al. (2021) for formal definitions

# 4 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

Abel, D.; Arumugam, D.; Lehnert, L.; and Littman, M. 2018. State abstractions for lifelong reinforcement learning. In *ICML*, volume 80, 10–19.

Dietterich, T. G. 2000. State abstraction in MAXQ hierarchical reinforcement learning. In *NeurIPS*, 994–1000.

Eppe, M.; Nguyen, P. D. H.; and Wermter, S. 2019. From Semantics to Execution: Integrating Action Planning With Reinforcement Learning for Robotic Causal Problem-Solving. *Frontiers in Robotics and AI* 6: 123.

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.

Grounds, M.; and Kudenko, D. 2005. Combining reinforcement learning with symbolic planning. In *AAMAS III*, volume 4865, 75–86.

Illanes, L.; Yan, X.; Icarte, R. T.; and McIlraith, S. A. 2020. Symbolic Plans as High-Level Instructions for Reinforcement Learning. *ICAPS* 540–550.

Jiang, Y.; Yang, F.; Zhang, S.; and Stone, P. 2019. Task-Motion Planning with Reinforcement Learning for Adaptable Mobile Service Robots. In *IROS*, 7529–7534.

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.

Konidaris, G. 2019. On the necessity of abstraction. *Current Opinion in Behavioral Sciences* 29: 1–7.

Lake, B. M.; Ullman, T. D.; Tenenbaum, J. B.; and Gershman, S. J. 2017. Building machines that learn and think like people. *Behavioral and brain sciences* .

Li, L.; Walsh, T. J.; and Littman, M. L. 2006. Towards a Unified Theory of State Abstraction for MDPs. In *ISAIM*, volume 4, 5.

Lyu, D.; Yang, F.; Liu, B.; and Gustafson, S. 2019. SDRL: Interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. In *AAAI*, 2970–2977.

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.

Raedt, L. D.; Kersting, K.; Natarajan, S.; and Poole, D. 2016. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 10(2): 1–189.

Ravindran, B.; and Barto, A. G. 2003. SMDP Homomorphisms: An Algebraic Approach to Abstraction in Semi Markov Decision Processes. In *IJCAI*, 1011–1018.

Sorg, J.; and Singh, S. 2009. Transfer via soft homomorphisms. In *AAMAS*, 741–748.

Walsh, T. J.; Li, L.; and Littman, M. L. 2006. Transferring state abstractions between MDPs. In *ICML Workshop on Structural Knowledge Transfer for Machine Learning*.

Yang, F.; Lyu, D.; Liu, B.; and Gustafson, S. 2018. PEORL: Integrating symbolic planning and hierarchical reinforcement learning for robust decision-making. *IJCAI* 4860–4866.