# Towards Knowledge-Augmented Agents for Efficient and Interpretable Learning in Sequential Decision Problems

Reem Alansary and Nourhan Ehab
*Department of Computer Science and Engineering, German University in Cairo, Cairo, Egypt*

Keywords: Neuro-Symbolic AI, Reinforcement Learning, Representation Learning.

Abstract: The advantages of neurosymbolic systems as solvers of sequential decision problems have captured the attention of reseachers in the field of AI. The combination of perception and cognition allows for constructing learning agents with memory. In this position paper, we argue that the decision-making abilities of such knowledge-augmented solvers transcend those of black-box function approximators alone as the former can generalize through inductive reasoning to behave optimally in unknown states and still remain fully or partially interpretable. We present a novel approach leveraging a knowledge base structured as a layered directed acyclic graph, facilitating reasoned generalization in the absence of complete information.

## 1 INTRODUCTION

With the onset of the third AI Summer (Kautz, 2020), Neuro-Symbolic systems have captured the attention of AI researchers, enthusiasts, and the scientific community to address problems where “solutions that work” are no longer sufficient. It is now of great importance to have solutions that are efficient and transparent, at least to a certain degree. Our contention is that a system integrating both neural and symbolic components holds added value in achieving this objective. On the one hand, data-driven systems rarely require expert interference or detailed and specific algorithms to learn a solution of some task, but in most cases they need to witness myriads of data. As a result, a system trained in this way is robust against noise, but turns out to be costly in terms of computation time and resources. Furthermore, they tend to produce solutions of an esoteric nature incapable of being deciphered simply by scrutiny and analysis. On the other hand, systems built using symbols and rules have to have an internal store of knowledge, known as a knowledge base (KB); this KB is the defining factor for the system’s behaviour as indicated by the Knowledge Representation Hypothesis (Brachman and Levesque, 2004). Nevertheless, knowledge-based programs can not gracefully handle erroneous input and the inference they employ is by itself computationally expensive.

Our intent is not to undermine the success of either neural or symbolic systems. Nonetheless, a single-type program fails to emulate human cognitive capacities, a quintessential feature of intelligent machines. Humans learn new things all the time, but they do not learn without background knowledge most of the time. They can also utilize language to describe or even explain their knowledge of the world. Thus, they use both their perception and cognition abilities. Therefore, we posit that a true emulation of the human mind demands a machine that amalgamates the strengths of both data-driven and symbolic AI as each type is comparable to a mode of thinking of the human mind (Evans and Stanovich, 2013).

This position paper specifically addresses problems inherently demanding a neuro-symbolic system, particularly sequential decision problems (Barto et al., 1989). We center our discussion on the subset of such problems that are either too big or too complex to have a complete KB. Solutions for these problems must be learned through repeated observation and prediction. However, if a learning-only approach is adopted without utilizing cognition to reason over the learned knowledge, generalization may be a lethargic process, and if a neural network is used for speeding up with increasing problem scale, interpretability of the achieved solution would be compromised. Hence, we argue that solvers of such problems should begin by learning partial solutions that can be represented symbolically in a layered directed acyclic graph where similar nodes can be grouped together and in a language that permits inductive reasoning to enable fast and interpretable generalization.

Alansary, R. and Ehab, N.
Towards Knowledge-Augmented Agents for Efficient and Interpretable Learning in Sequential Decision Problems.
DOI: 10.5220/0012430900003636
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 3*, pages 1014-1019
ISBN: 978-989-758-680-4; ISSN: 2184-433X
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

## 2 SIMILARITY REVELATIONS

Before delving into the available approaches for tackling the problem of efficient and interpretable generalization in sequential decision problems, we introduce a simple model of such problems: the Markov Decision Process (MDP). An MDP $M$ is formally defined as the 4-tuple $M = (S, T, A, R)$ where:

- $S$ is the state space,
- $T$ is the transition model,
- $A$ is the action space, and
- $R$ is the reward function.

The transition model $T$ usually has a probabilistic nature and the reward function is always defined with respect to some goal. In other words, for all states $s \in S$, where the goal is achieved, the reward is maximum. Any MDP $M$ exhibits the Markovian Property that guarantees that the current state of the world is affected only by the most recent previous state.

A solution for $M$ then would be the optimal global policy $\pi^*$ found by an abstract embodiment of the solver as a rational agent which learns by reinforcement. This agent finds the optimal global policy by repeatedly observing state-action-reward sequences in the MDP and making predictions as to which actions would lead it to its goal.

Early efforts to build a reinforcement learning (RL) agent capable of learning $\pi^*$, that were based on temporal difference methods (Sutton, 1988), resulted in the development of several analytically transparent algorithms (Yu et al., 2021). However, not much later, it was found that emergent algorithms that were model-free, such as Q-Learning or SARSA, did not scale well (Kaelbling et al., 1996). This was due to their operation without a model of the environment. That is, these algorithms disregard the importance of similarity of agent behaviour in state-action-reward sequences and the similarity of state representations. This early finding comports with our position that capitalizing on such similarity would aid in the generalization process of any rational agent.

Subsequent research explored model-based algorithms, such as Dyna and prioritized sweeping (Sutton et al., 2012), that capitalize on similarity of agent behaviour or state representation. In the next section, we delve into approaches aiming to learn representations of the underlying problem to facilitate generalization.

## 3 MODEL REPRESENTATIONS

In order to mitigate the difficulties hindering fast and interpretable generalization, researchers shifted their attention from model-free to model-based algorithms. However, not all proposed solutions revolved around representing the environment model or parts of it to discover regularities between states. Some approaches were motivated by abstracting the similarity of agent behaviour across state-action-reward sequences.

One method of abstracting granular operations transformed the underlying problem model from a regular MDP into a Semi-MDP (SMDP) (Barto and Mahadevan, 2003), where the agent may remain in one state for more than a single time step. This characteristic waiting time gives the agent the opportunity to execute multiple primitive actions. Thus, the notion of macros was introduced. Through the use of macro operators, the agent behaviour is successfully abstracted as the invocation of any macro is analogous to compounding state-action-reward sequences. This representation of the problem as an SMDP motivates hierarchical RL, where a common architecture of a solution is to segment the environment and assign an agent to each segment, such that these agents report back to a master agent that then would have "macro" information of the optimal policy of each segment. This approach can also be extended to function in partially observable MDPs (POMDPs) by adding hierarchical memory structures where agents, at each level corresponding to an abstraction degree, represent their belief states. In a POMDP, every state can be partially observable, thus an agent observes a belief state that corresponds to a probability distribution over the set of possible states that it can be in.

On a similar note, agent behaviour can be abstracted by applying a bisimulation metric to the optimal policies in different states to measure their similarity without depending on the knowledge of the reward function (Agarwal et al., 2021). Although we believe that incorporating reward information is at the heart of goal-directed generalization (Landajuela et al., 2021), this approach has shown promising results for control problems as it identifies states as close when their policies are similar. Hence, it can be argued that since the policy is used to measure similarity, then the reward function is taken into consideration indirectly.

Another relevant endeavor to generalize and enable an RL agent to learn the optimal policy faster adopted a data mining approach (Apeldoorn and Kern-Isberner, 2017) to discover the best action in any state quickly (Apeldoorn and Kern-Isberner, 2017). The goal was to represent the optimal policy as a structured KB of weighted rules, such that each abstract rule may indicate the best action in a group of states where the agent perceptions in each state are similar. The agent starts exploiting the learned knowledge according to the strategic depth measure that is based on the size of the KB.

The previous approaches experimented with learning symbolic knowledge to enable the learners to act fast in large environments. There is another research direction which assumes that background knowledge is fully or in part available (Ledentsov, 2023, for instance). While we contend that this is infeasible for every environment as knowledge may not always be in abundance, the results support our position that learning becomes faster and more interpretable when a symbolic component is present.

Recent advances in this direction include works on control problems, where deep RL is employed to learn mathematical functions as approximations for each action in the action space in a sequential manner (Landajuela et al., 2021); this approach reduces complexity of the environment by building the function approximators step-by-step and giving a neural network prior information of properties of certain functions to inhibit the generation of trivial action represenations, such as an invertible function and its inverse and applying one to the other or vice versa. This corresponds to creating an ordering of actions $(a_1, a_2, a_3, .., a_n)$ for some $n \in \mathbb{N}$ such that the function approximator for $a_{i+1}$ is searched for only when the complexity of the search space has been reduced by fixing representations for action $a_1$ through $a_i$ for $1 \leq i \leq n$.

Special attention has been accorded to the importance of model-aided learning in environments with sparse rewards, where the nature of the reward function may render the modelling of the environment as an MDP inadequate. The use of a partial model representation in the form of a declarative planning model provided to an RL agent from humans as advice was explored to simplify the underlying problem (Guan et al., 2022). The planning model helped the agent decompose the problem into sub-problems or landmarks. Thus, hierarchical RL could be employed to find the optimal solution for each landmark and the optimal policy is obtained by stitching together these individual solutions. On a different note, such environments could be regarded as completely non-Markovian and as such a different learning approach should be used. One example, which conforms with our standpoint that generalizations should be learned, learns a deterministic finite automaton (DFA) that simplifies the problem by providing a Markovian decomposition of it (Christoffersen et al., 2023). After this transformation, the DFA and observations are used in conjunction to learn the optimal policy by tabular Q-learning.

## 4 OUR PROPOSED APPROACH

Humans generalize efficiently only when they use their memory to keep track of past experiences in a language that allows for inductive reasoning and the best decision-makers are those with the best general experience. They are those with the best knowledge bases. We believe that this idea should be maintained as the heart of any solution that considers efficiency and interpretability for any sequential decision problem. Otherwise, we would be running into the old problem of having learning agents that are expected to behave optimally in short time, but on every power-on they must learn with a clean slate (Kaelbling et al., 1996).

Learning with a clean slate is not enough for large or complex environments. In order to behave optimally and in a shorter time in such environments, an agent must leverage its memory by reasoning with past experience. We argue that such behaviour is accomplished through a symbolic representation of knowledge as the agent's behaviour would be entrenched in this representation.

Means to this end, governed by the underlying model of the sequential decision problem at hand, vary depending on the nature of the domain; whether it is discrete or continuous. Within this line of research, endeavors can be categorized based on the extent of available knowledge for utilization. Some approaches assume either partial or complete access to symbolic knowledge in a defined format, employing it as a guiding framework for a learning agent. Conversely, alternative methodologies avoid such assumptions, opting to concurrently acquire the symbolic representation while navigating the path to an optimal policy.

We stand with the second view and propose a solution with the initial absence of knowledge in fully observable environments. Necessary knowledge can then be acquired through symbolic function approximators or by reasoning on some logical representation. We are currently working on systems that use tabular Q-learning for learning logical knowledge, stored in a layered graph-based KB, that can be generalized through inductive reasoning as it is our belief that this approach is best suited for preserving interpretability as well as for being efficient.

### 4.1 KB Structure

Addressing the crux of efficient learning and generalization, we propose a graph-based Knowledge Base (KB) structured into layers. Each layer corresponds to the agent's sensors, forming a hierarchy that aids

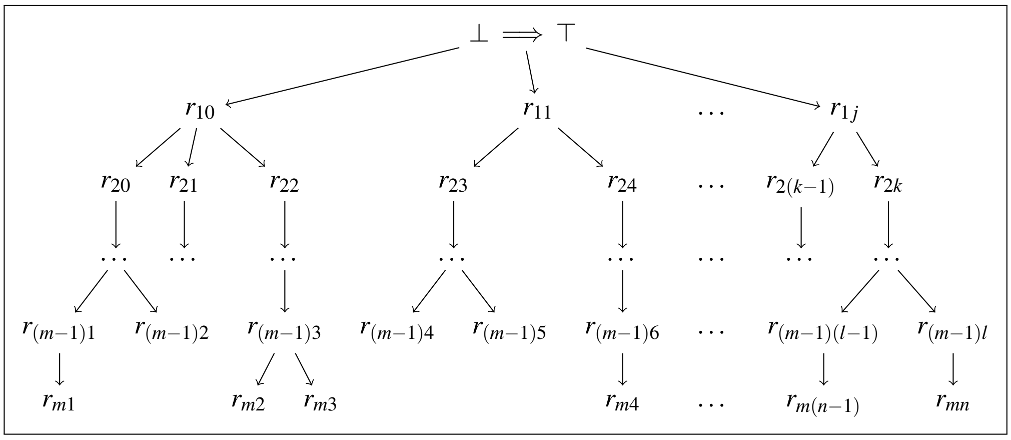


Figure 1: An abstract example of the graph-based KB where each $r_{ab}$ is a rule in level $a$ and its index in that level is $b$.

in efficient navigation and representation of learned rules. The graph data structure is chosen over the classical flat KB as it can separate the learned rules such that similar ones remain closer together. Besides, the directed links translate the relation between the rules in two connected nodes.

The zeroth level of the KB is made up of one utility node that holds the structure together but is not representative of a rule specific to any decision problem; it will always hold the material conditional

$$\perp \implies \top,$$

which serves as a formalization of the fact that any agent situated in an unknown environment without prior knowledge will believe that it can execute any action and that all actions are equally preferable with respect to its goal, embodied by the reward function. Although any valid sentence would have served the same purpose, the above sentence was favored as it conforms to the standard format of the rules and it intuitively maps to the behaviour of a learning agent starting out in any environment. Any node, other than the utility node, holds a rule consisting of a discretized state from an environment, as the antecedent, and the corresponding best action as the consequent of a material conditional. Records of the total number of application and the number of successes of a rule are kept alongside the rule in the corresponding node. Furthermore, on the grounds that each rule is obtained through an inductive process that generates a belief that could be revised not absolute knowledge, we associate with each rule a weight measuring its credibility. For nodes in the bottom level, this weight is based on the q-value of the rule, which comes directly from the learning algorithm and is a measure of how relevant this rule is for the agent's goal, and the successes and tries of this rule. Each rule in any other level is induced and its weight is computed from the weights of the rules in the nodes that participated in its induction.

The levels of the KB are finite and directly correspond to the number of sensors of the agent; if the agent has $m$ sensors, then the graph has $m+1$ levels, where the $i^{th}$ level has $i$ sensor values forming the antecedent of every rule for $0 \leq i \leq m$. Thus, the bottom level contains the rules that directly map to states and actions obtained from the learning algorithm. The rules in every other level, where $i \neq m$, are created by an inductive inference process using those in level $i+1$ or the level directly below. A graphic example is shown in figure 1, where out of all the contents of a single node only the rule is shown for simplicity. These rules can be used to interpret the behaviour of the agent and the outcome of the learning algorithm as each rule maps either a full or partial state to the perceived optimal action with respect to some goal; rules in the bottom level correspond to full states, but rules in any other level correspond to partial states as they are induced. This mapping, enforced by the rule format, helps in clarifying the agent's belief about the outcome of executing some action in some state and why the agent's beliefs may change with any change of an observation.

## 4.2 KB Operations

The KB provides two functionalities; the first is to store knowledge and the second is to utilize it. In order to accomplish both, the KB is furnished with 4 operations, briefly described below.

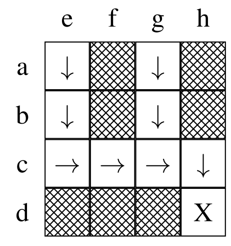
**Insert.** This is a *TELL*-type operation invoked by an agent to add knowledge to the KB. The knowledge can be a new rule or experience for an already existing rule. If a new rule is to be added, then it added directly to the last level in a node with application and successes numbers equal zero. However, if experience is to be incorporated and it is not contradictory, only the numbers of application and successes are updated. In the case of contradictory evidence, which may occur if a different best action for the same state is learned, the old rule and its credibility are updated, and the numbers of application and successes are reset.

**Infer.** This is an *ASK*-type operation to apply an inference method for detecting implicit knowledge in the KB. This operation adds the implicit knowledge to the graph in the form of nodes as described above in level $i$ if they were inferred from rules in nodes in level $i+1$, where $0 \leq i \leq m$ and $m$ is the number of sensors of the agent. The addition of a new node $n_{new}$ is accompanied by the addition of directed edges from this node to each of the nodes from which its rule was inferred. An edge is also added from the node $n_{old}$ that was already connected to these nodes, used in the inference process, to $n_{new}$, whereas the old edges are removed. Thus, the implicit knowledge becomes explicit.

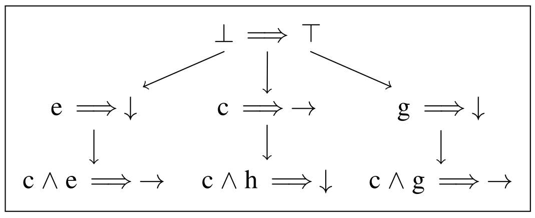
**Search.** This is an *ASK*-type operation for knowledge present explicitly in the KB. The pattern to be matched is the discretized state of the environment, which is always a superset of the antecedent of each rule. Searching the graph happens in a top-down manner, starting from the utility node and moving down level by level. If several matches are found in a single level, the tie is broken by comparing the credibility measures or the weights of each rule and only the subgraph connected to the rule with the highest credibility is further searched.

**Prune.** This is a utility operation to prevent cluttering the KB with unnecessary knowledge. A rule becomes redundant when there exists an inferred rule that subsumes it. Therefore, the node that contains the redundant rule should be removed.

Figure 2 is an example that gives a concrete illustration of the expected result of applying these operations on the learned policy by an agent in a 4x4 environment that is trying to reach the cell marked with an "X" while avoiding obstacles, where again for simplicity only the material implication in each node is shown. The agent in this environment has two sensors; one for the horizontal axis and the other for the vertical axis. The arrows representing the consequent of every rule give the direction the agent should follow when in the state represented by the antecedent of the same rule. Rules in the bottom level, which represent full states, are not pruned when they cannot be subsumed by other rules in a higher level.



(a)



(b)

Figure 2: An example of a 2D environment (2a) and its corresponding KB (2b).

Usually the most expensive operations are the ones concerned with utilizing knowledge and these correspond to the ASK-type operations: infer and search. By representing the knowledge in a layered graph that is also directed and acyclic, we can greatly improve the complexity of search, because intuitively once the search goes from a level to the one below it, many nodes would be disqualified without inspection of their contents. However, the structure does not improve the complexity of the inference procedure that much by itself; the graph will be built from the bottom layer upwards using the experience of the agent directly and at any given time when the inference procedure should be run, all nodes in a certain level will be considered. Therefore, it is our intention to invoke the infer operation at regular intervals separated by several learning episodes to prevent the inference procedure from consuming resources when the observed knowledge has not changed by much. After the graph has been partially built, it can be consulted by the agent to speed up the learning of the optimal policy.

## 5 CONCLUSIONS

Solutions for sequential decision problems that scale well and remain transparent to a degree must incorporate both components of a neurosymbolic system. The data-driven component enables learning through perception and the symbolic component endows the system with cognitive power. We have argued that the presence of knowledge in a KB allows the solver of a sequential decision problem to reason in order to generalize faster and in a more interpretable manner.

In consequence, we proposed a solution for problems that are fully observable in the form of such a system of two components. The reasoning component will make use of a KB organized as a layered directed acyclic graph and after a sufficient period of learning and generalization the learner will turn to the KB to expedite the process of learning the optimal policy. This is analogous to capitalizing on memory of past experience, that forms the cognitive abilities of the solver, in the face of the unknown, which is much better than facing the unknown with high hopes.

We are currently investigating extensions of the proposed neuro-symbolic framework to partially observable environments and the integration of additional cognitive elements in the KB. Furthermore, we are exploring enhancing the efficiency of the inferential procedures in the proposed KB structure. Additionally, we plan to conduct empirical validations across diverse problem domains to provide insights into the broader applicability and robustness of the proposed approach, paving the way for advancements in neuro-symbolic systems for artificial intelligence applications.

## REFERENCES

Agarwal, R., Machado, M. C., Castro, P. S., and Bellemare, M. G. (2021). Contrastive behavioral similarity embeddings for generalization in reinforcement learning.

Apeldoorn, D. and Kern-Isberner, G. (2017). An agent-based learning approach for finding and exploiting heuristics in unknown environments. In *International Symposium on Commonsense Reasoning*.

Apeldoorn, D. and Kern-Isberner, G. (2017). Towards an understanding of what is learned: Extracting multi-abstraction-level knowledge from learning agents. In Rus, V. and Markov, Z., editors, *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2017, Marco Island, Florida, USA, May 22-24, 2017*, pages 764–767. AAAI Press.

Barto, A. G. and Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13:41–77.

Barto, A. G., Sutton, R. S., and Watkins, C. (1989). Sequential decision problems and neural networks. *Advances in neural information processing systems*, 2.

Brachman, R. and Levesque, H. (2004). *Knowledge Representation and Reasoning*. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann, Amsterdam.

Christoffersen, P. J. K., Li, A. C., Icarte, R. T., and McIlraith, S. A. (2023). Learning symbolic representations for reinforcement learning of non-markovian behavior.

Evans, J. S. B. T. and Stanovich, K. E. (2013). Dual-process theories of higher cognition: Advancing the debate. *Perspectives on Psychological Science*, 8(3):223–241. PMID: 26172965.

Guan, L., Sreedharan, S., and Kambhampati, S. (2022). Leveraging approximate symbolic models for reinforcement learning via skill diversity.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *CoRR*, cs.AI/9605103.

Kautz, H. (2020). Rochester hci. https://roc-hci.com/announcements/the-third-ai-summer/. Accessed on January 6th, 2024.

Landajuela, M., Petersen, B. K., Kim, S., Santiago, C. P., Glatt, R., Mundhenk, N., Pettit, J. F., and Faissol, D. (2021). Discovering symbolic policies with deep reinforcement learning. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5979–5989. PMLR.

Ledentsov, A. (2023). Knowledge base reuse with frame representation in artificial intelligence applications. *IAIC Transactions on Sustainable Digital Innovation (ITSDI)*, 4(2):146–154.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Mach. Learn.*, 3(1):9–44.

Sutton, R. S., Szepesvári, C., Geramifard, A., and Bowling, M. P. (2012). Dyna-style planning with linear function approximation and prioritized sweeping. *arXiv preprint arXiv:1206.3285*.

Yu, D., Yang, B., Liu, D., and Wang, H. (2021). A survey on neural-symbolic systems. *CoRR*, abs/2111.08164.