# Application of Neurosymbolic AI to Sequential Decision Making

**Carlos Núñez-Molina**
Universidad de Granada, Spain
ccaarlos@correo.ugr.es

## Abstract

In the history of AI, two main paradigms have been proposed to solve Sequential Decision Making (SDM) problems: Automated Planning (AP) and Reinforcement Learning (RL). Among the many proposals to unify both fields, the one known as *neurosymbolic AI* has recently attracted great attention. It combines the Deep Neural Networks used in modern RL with the symbolic representations typical of AP. The main goal of this PhD is to progress the state of the art in neurosymbolic AI for SDM, developing methods for both solving these problems and learning aspects of their structure.

Sequential Decision Making (SDM) is the problem of solving Sequential Decision Processes (SDP). In an SDP, an agent must make a series of decisions in order to complete a task or achieve a goal. These decisions must be selected according to some optimality criteria, usually corresponding to the maximization of reward or the minimization of cost. SDPs provide a general framework which has been successfully applied to fields as diverse as robotics, logistics, games and finance.

AI methods for solving SDPs can be grouped in two main categories: Automated Planning (AP) and Reinforcement Learning (RL). These two paradigms mainly differ in how they obtain the solution and how they represent their knowledge. AP exploits the prior knowledge about the environment dynamics, encoded in the planning domain, to search for a plan that achieves the goals. This knowledge is usually encoded in a symbolic manner, using declarative languages. Standard RL methods learn the optimal policy, i.e., a mapping from states to actions to maximize reward, automatically from data without planning. This policy is often represented in a subsymbolic way, as a Deep Neural Network (DNN).

The main advantages of AP are the interpretability of its knowledge representation and its suitability for long-term reasoning. The main advantage of RL is its ability to learn automatically from data. Since the shortcomings of AP align with the strengths of RL and vice versa, many approaches have tried to unify these two paradigms, such as model-based RL, relational RL, methods for learning the structure of the SDP (e.g., the planning domain) and neurosymbolic AI, a novel approach that combines the DNNs of Deep Learning (DL) and Deep RL with the symbolic representations of AP.

The main goal of my PhD is to progress the state of the art in neurosymbolic AI for SDM, with the development of methods for both solving SDPs and learning aspects of their structure. This PhD contains three different lines of research.

**Study of state of the art.** The first line of research corresponds to a study of AI methods for SDM. This study has already been completed and the results summarized into a review which we intend to submit very soon. To the best of our knowledge, it is the first work that presents an overview of AI methods for both solving SDPs and learning their structure, discussing methods ranging from the symbolic approach of AP to the subsymbolic one of DL, RL and Deep RL, in addition to many hybrid approaches in between. In this work, we also discuss what properties an ideal method for SDM should possess and argue that neurosymbolic AI is currently the closest existing approach to this ideal method.

This study has helped us identify several directions for future work, two of which are explored in this PhD. The first one regards Goal Reasoning (GR), a design philosophy in which agents do not only reason about how to achieve their goals but also about which goals to pursue (in order to achieve a final goal or complete a task). Several works learn to select goals for AP from data, using either traditional RL [Jaidee *et al.*, 2012] or Supervised Learning (SL) [Bonanno *et al.*, 2016]. We propose a Deep RL method for goal selection, which is able to generalize to unseen states and goals (unlike traditional RL) and requires less prior knowledge than SL.

The second one regards planning problem generation methods. A planning problem contains the initial state and goal(s) to achieve and, along with a planning domain, defines a planning task. Generating planning problems serves two main purposes: 1) generating data for training machine learning methods (such as those that learn planning heuristics or hierarchical task network (HTN) domains) and 2) creating benchmark problems to evaluate planners (as in AP competitions). Although there exist several methods for generating planning problems, they either focus on a specific domain or are domain-general but cannot generate problems that are valid, diverse and of good quality [Fuentetaja and De la Rosa, 2012]. We propose a domain-independent method for generating planning problems, represented in the PDDL language, which will use Deep RL to guide the search towards problems with the properties described above.

**Integration of AP and Deep RL.** The second line of research is also initiated and focuses on the integration of AP and Deep RL. We have developed a neurosymbolic architecture that combines Deep Q-Learning to learn to select goals and AP to achieve them. This approach has been tested on a deterministic version of the 2D game known as *BoulderDash*, employing different levels for training and testing in order to evaluate its generalization ability. We compared our approach against both standard AP and standard Deep Q-Learning. The results obtained showed that 1) Deep Q-Learning can learn to select goals effectively and 2) our hybrid approach performs much better (when both time requirements and plan length are considered) than AP and RL on their own. The first version of this work was presented at a workshop, the second one at a conference and the third one [Núñez-Molina *et al.*, 2022] has recently been accepted for publication in a journal.

We plan to continue this research line during my PhD. Firstly, we will extend our goal selection approach to stochastic environments. To do so, the model will also be trained to predict how likely a plan is to be executed without interruptions (e.g., an obstacle suddenly appearing). This uncertainty value will be monitored during execution so that, when an obstacle appears, a new goal is pursued. This approach will make possible to apply a deterministic planner to a stochastic environment, since all the stochasticity is managed by the goal selection procedure. We will test our enhanced architecture on stochastic GVGAI games.

Secondly, we will apply our goal selection approach to a real-world problem. The goal is to manage the logistics of a company which employs trucks to transport and deliver packages. We propose to use our approach to map high-level decisions to goals and train Deep Q-Learning to select them so as to satisfy a set of criteria (e.g., time deadlines) and optimize a series of metrics (e.g., minimize fuel consumption). Then, a symbolic planner will be used to find plans to the selected goals. Additionally, the state of the execution will be monitored in order to modify the goals to achieve when needed. Since the training data will be in the form of logs, we will need to employ a DNN architecture suitable for symbolic data, such as Graph Neural Networks (GNN). Finally, the quality of the plans will be compared with the quality of those obtained with the previous approach used by the logistics company.

**Automatic generation of planning problems.** The third line of research focuses on a neurosymbolic method for generating planning problems and its possible applications to learn aspects of the SDP structure. Planning problem generation itself can be posed as a SDM task. The initial state can be generated by an iterative process that, at each step, adds a new object or a new predicate (relation) between the existing objects. The goal can be generated by executing a sequence of actions from the initial state to arrive at a final state, and then selecting a subset of the state predicates. We propose to adapt the method used in [You *et al.*, 2018] for successfully generating molecular graphs to generate planning problems for any given planning domain. A GNN will select the action to apply at each generation step and will be trained with RL to generate problems that are valid, diverse and of good quality (e.g., hard to solve). The validity and quality of the generated problems can be evaluated by solving them with an off-the-shelf planner and analyzing the resolution metrics. On the other hand, the diversity of the problems will probably need to be assessed by a human.

Once implemented and tested, we will apply our problem generation method to learn hierarchical planning domains, represented as HTNs. HTNs provide an interesting alternative to non-hierarchical planning domains, as they not only represent the environment dynamics but also encode useful strategies for achieving the goals. However, designing HTN is usually a very time-consuming endeavor. For this reason, many methods have been developed to automatically learn HTN domains from data. Despite their usefulness, they usually require plan traces as input, which may be hard to obtain. To solve this, we plan to generate planning problems with the method previously discussed and use a planner to solve them. Then, the plan traces will be provided to a state-of-the-art method for learning HTN domains. This will make possible to learn an HTN domain just from a non-hierarchical domain, without needing plan traces. The quality of the HTN domains generated will be evaluated by measuring the time difference when solving a planning task with the HTN domain instead of the non-hierarchical one. To the best of our knowledge, [Lotinac and Jonsson, 2016] is the only existing work that learns HTN domains without plan traces. Nonetheless, as the authors point out, their method does not perform well in some domains, as the learned HTNs are very general.

## Acknowledgements

This PhD is being supervised by Dr. Juan Fdez-Olivares and Dr. Pablo Mesejo, and funded by Projects PYC20-RE-049-UGR and RTI2018-098460-B-I00 with FEDER funds.

## References

[Bonanno *et al.*, 2016] David Bonanno, Mark Roberts, Leslie Smith, and David W Aha. Selecting subgoals using deep learning in minecraft: A preliminary report. In *IJCAI Workshop on Deep Learning for AI*, 2016.

[Fuentetaja and De la Rosa, 2012] Raquel Fuentetaja and Tomás De la Rosa. A planning-based approach for generating planning problems. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[Jaidee *et al.*, 2012] Ulit Jaidee, Héctor Munoz-Avila, and David W Aha. Learning and reusing goal-specific policies for goal-driven autonomy. In *ICCBR*, 2012.

[Lotinac and Jonsson, 2016] Damir Lotinac and Anders Jonsson. Constructing hierarchical task models using invariance analysis. In *ECAI*. 2016.

[Núñez-Molina *et al.*, 2022] Carlos Núñez-Molina, Juan Fdez-Olivares, and Raúl Pérez. Learning to select goals in automated planning with deep q-learning. *Expert Syst. Appl.*, 2022. In Press.

[You *et al.*, 2018] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *NeurIPS*, 2018.