FOND4LTL$_f$: FOND Planning for LTL$_f$/PLTL$_f$ Goals as a Service

Giuseppe De Giacomo[1], **Francesco Fuggitti**[1,2]
[1] University of Rome "La Sapienza", Italy
[2] York University, Toronto, ON, Canada
{lastname}@diag.uniroma1.it

Abstract

Planning is a central area in Artificial Intelligence (AI) concerned with the decision making performed by autonomous agents with the aim of achieving some goals. In the last decades, extensive literature has been produced in Fully Observable Non-Deterministic (FOND) planning for temporally extended goals when the specification is expressed using one of the several finite trace variants of LTL. Numerous applications have been developed to solve the problem, but most of them require special knowledge. In this demonstration, we present FOND4LTL$_f$, a web service tool that allows solving FOND planning for LTL$_f$/PLTL$_f$ goals as an integrated tool with editor, planners and policy visualizer. Video: https://bit.ly/3ilX7JR.

1 Introduction

Planning for *temporally extended goals* in *deterministic* and *non-deterministic* domain settings has been of increasing interest over the past decades, starting with the pioneering work on planning for temporally extended goals (Bacchus and Kabanza 1998) and on planning via model checking (Pistore, Bettin, and Traverso 2001); then, with the work on integrating LTL goals into standard planning tools (Patrizi, Lipovetzky, and Geffner 2013), and, more recently, with the work relating planning in non-deterministic domains to synthesis, often focused on the *finite trace* variants of LTL (De Giacomo and Vardi 2013, 2015; Camacho et al. 2017, 2018; De Giacomo and Rubin 2018; De Giacomo et al. 2020). In the *Fully Observable Non-Deterministic* (FOND) setting, two techniques have been mainly exploited. One uses automata-theoretic approaches; whereas, the second one integrates the automaton dynamics directly within the compactly represented FOND domain model.

Although some of these techniques have already been implemented, such implementations involve several steps and, in general, do not conform on input/output formats. Thus, on the one hand non-expert users usually find challenging to understand the whole process pipeline and might encounter several difficulties for the correct usage of such tools. On the other hand, experienced users cannot check or debug solutions with ease.

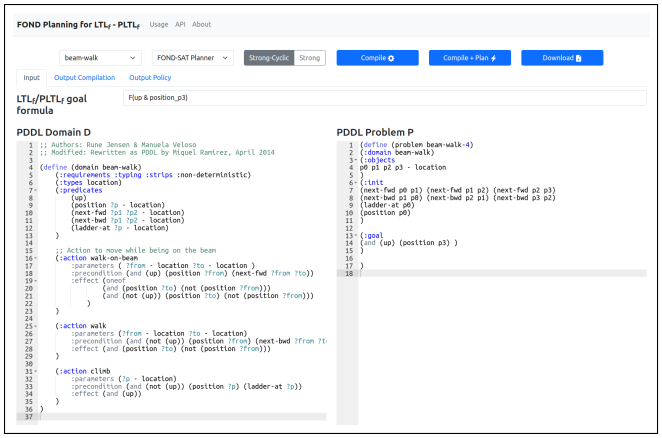

Figure 1: FOND4LTL$_f$ input section.

Therefore, this demonstration introduces FOND4LTL$_f$, an online web service able to compile FOND planning for LTL$_f$/PLTL$_f$ goals into standard FOND planning and compute the associated policy, if it exists. The web service comes with a user-friendly graphical interface, but also exposes its functionalities through RESTful APIs.

2 FOND Planning for LTL$_f$/PLTL$_f$ Goals

Although there are many variants of compilation-based approaches to solve FOND Planning for temporally extended goals, they all share the same high-level idea.

Given a FOND planning domain model $\mathcal{P}$ with initial state s_0 (both represented in PDDL), and an LTL$_f$/PLTL$_f$ goal formula φ, the approaches work as follows: (i) Transform the goal formula φ into the corresponding (DFA) using off-the-shelf translators (e.g., (Zhu et al. 2017; Fuggitti 2019; De Giacomo and Favorito 2021)); (ii) Build a new domain model $\mathcal{P}'$, by augmenting $\mathcal{P}$ with the automaton dynamics and states; (iii) Solve $\mathcal{P}'$ with any off-the-shelf FOND planner (e.g., (Mattmüller et al. 2010; Muise, McIlraith, and Beck 2012; Geffner and Geffner 2018)); (iv) Extract from $\mathcal{P}'$ a solution to $\mathcal{P}$. In this way, FOND planning for temporally extended goals is reduced to standard FOND planning. The main advantage of this solution is that any off-the-shelf FOND planner can be leveraged. Our application allows only

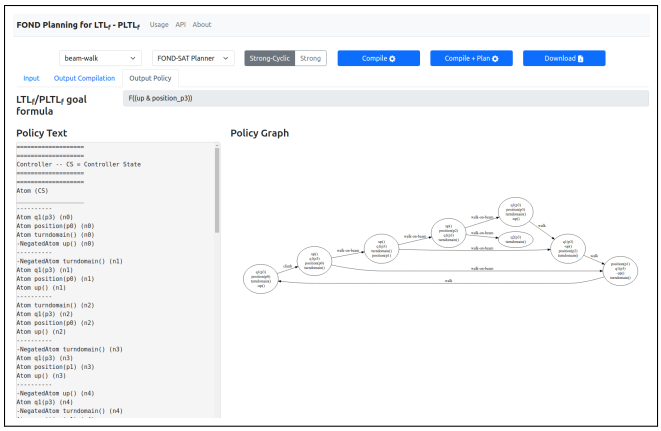

Figure 2: FOND4LTL$_f$ output policy section.

LTL$_f$ and PLTL$_f$ as formalism to specify the goal, but the system can be easily adapted to accommodate others as well.

3 FOND4LTL$_f$ Web Service

The FOND4LTL$_f$ web service presented here is based on the homonym implementation (Fuggitti 2021), developed in (Fuggitti 2018), which implements the steps presented in Section 2. Such an implementation might be inefficient when the number of DFA states is large.

The web service comes with a graphical user interface and exposes RESTful APIs. A capture of the user interface is shown in Figure 1. At the beginning, the application requires the user to provide the PDDL domain and problem specifications along with an LTL$_f$/PLTL$_f$ goal. Users can either directly write their own models using the given editor or they can choose a preloaded working example.

Once the input is given, the web service allows two main options: *Compile* or *Compile + Plan*. The former computes the DFA for the goal formula with the LTL$_f$2DFA (Fuggitti 2019) tool and integrates its dynamics within the PDDL domain model, returning a new domain and a new problem. The latter, instead, first performs the compilation just described and then calls an off-the-shelf FOND Planner to find a policy for the planning problem. The user can choose among the three main state-of-the-art planners, i.e., FOND-SAT, MyND and PRP. Moreover, based on the chosen planner, the user can also decide to ask for *strong-cyclic* or *strong* policies. If the planner finds a policy within a given timeout, the output policy is shown to the user and, from such a policy, the system computes the corresponding graph. In this way, the user can look at the policy details in the textual representation or can just have an overview with the policy graph. A capture of a policy output is shown in Figure 2. In any case, at the end of the process, the application allows the user to download all the produced content. The tool is available online at https://fond4ltlf.herokuapp.com.

Conclusion We described FOND4LTL$_f$, an easy-to-use web service tool that compiles FOND domain models for LTL$_f$/PLTL$_f$ goals into standard FOND planning problems and computes the policy with publicly available FOND planners.

Acknowledgments

Project partially supported by the ERC AG WhiteMech (No. 834228) and the EU ICT-48 2020 TAILOR (No. 952215).

References

Bacchus, F.; and Kabanza, F. 1998. Planning for temporally extended goals. *Annals of Mathematics and Artificial Intelligence* 22(1-2).

Camacho, A.; Baier, J.; Muise, C.; and McIlraith, S. 2018. Finite LTL Synthesis as Planning. In *ICAPS*.

Camacho, A.; Triantafillou, E.; Muise, C.; Baier, J.; and McIlraith, S. 2017. Non-Deterministic Planning with Temporally Extended Goals: LTL over Finite and Infinite Traces. In *AAAI*.

De Giacomo, G.; Di Stasio, A.; Fuggitti, F.; and Rubin, S. 2020. Pure-Past Linear Temporal and Dynamic Logic on Finite Traces. In *IJCAI*. Survey Track.

De Giacomo, G.; and Favorito, M. 2021. Compositional Approach to Translate LTLf/LDLf into Deterministic Finite Automata. In *ICAPS*.

De Giacomo, G.; and Rubin, S. 2018. Automata-Theoretic Foundations of FOND Planning for LTLf and LDLf Goals. In *IJCAI*.

De Giacomo, G.; and Vardi, M. 2013. Linear Temporal Logic and Linear Dynamic Logic on Finite Traces. In *IJCAI*.

De Giacomo, G.; and Vardi, M. 2015. Synthesis for LTL and LDL on finite traces. In *IJCAI*.

Fuggitti, F. 2018. *LTL and Past LTL on Finite Traces for Planning and Declarative Process Mining*. Master's thesis, Sapienza University of Rome.

Fuggitti, F. 2019. LTLf2DFA. doi:10.5281/zenodo.3888410. URL https://doi.org/10.5281/zenodo.3888410.

Fuggitti, F. 2021. FOND4LTLfPLTLf. doi:10.5281/zenodo.4876281. URL https://doi.org/10.5281/zenodo.4876281.

Geffner, T.; and Geffner, H. 2018. Compact Policies for Fully Observable Non-Deterministic Planning as SAT. In *ICAPS*.

Mattmüller, R.; Ortlieb, M.; Helmert, M.; and Bercher, P. 2010. Pattern Database Heuristics for Fully Observable Nondeterministic Planning. In *ICAPS*.

Muise, C.; McIlraith, S. A.; and Beck, J. C. 2012. Improved Non-deterministic Planning by Exploiting State Relevance. In *ICAPS*.

Patrizi, F.; Lipovetzky, N.; and Geffner, H. 2013. Fair LTL Synthesis for Non-Deterministic Systems using Strong Cyclic Planners. In *IJCAI*.

Pistore, M.; Bettin, R.; and Traverso, P. 2001. Symbolic techniques for planning with extended goals in non-deterministic domains. In *ECP*.

Zhu, S.; Tabajara, L. M.; Li, J.; Pu, G.; and Vardi, M. Y. 2017. Symbolic LTLf Synthesis. In *IJCAI*.