Reactive Synthesis of Linear Temporal Logic on Finite Traces

Shufang Zhu

University of Oxford, UK
shufang.zhu@cs.ox.ac.uk

Reactive synthesis promises to automatically generate a verifiably correct program from a high-level specification [13]. A popular such specification language is Linear Temporal Logic (LTL) [12]. Unfortunately, synthesizing programs from *general* LTL formulas, which rely on first constructing a game arena and then solving the game, remains challenging [8, 11]. Nevertheless, the synthesis problem of a finite trace variant of LTL, which is LTL_f [9], has shown to be much simpler than LTL synthesis [7]. The key idea is that synthesizing LTL_f formulas only involves games on finite traces instead of infinite traces as for LTL, though both problems share the same worst-case complexity of 2EXPTIME-complete.

In this paper, we will review an evolving journey motivated by this idea. We start from an attempt to devise a symbolic LTL_f synthesis framework [17], which consists of a backward reachability game on the constructed Deterministic Finite Automaton (DFA) of the corresponding LTL_f formula and has demonstrated its significant efficiency in various application scenarios. Then, the journey evolves into a forward LTL_f synthesis technique that synthesizes a strategy while constructing the DFA, thus being possible to avoid the 2EXPTIME worst-case complexity [15, 6]. Next, we study LTL_f synthesis under environment specifications, which are constraints on the environment that rule out certain environment behaviours [1, 2]. A key observation is that even if we consider an agent with LTL_f tasks on finite traces, environment specifications need to be expressed over infinite traces since accomplishing the agent tasks may require an unbounded number of environment actions [1, 2]. While a naive solution to LTL_f synthesis under environment specifications expressed in LTL would be reducing the problem to LTL synthesis, which remains challenging [1, 2], we show in this paper that we can avoid the detour to LTL synthesis and keep the simplicity of LTL_f synthesis in interesting cases. More specifically, we consider the following certain environment specifications: safety [3], simple fairness and stability [16], and Generalized-Reactivity(1) [4]. Furthermore, we show that even when the environment specifications are expressed in general LTL, we can still partially avoid the full detour to LTL synthesis [5].

In conclusion, reactive synthesis on LTL_f has been an exciting research problem. Our work on this problem spans from standard LTL_f synthesis to synthesis concerning environment specifications with efficient solution techniques. In the future, we would like to consider environment specifications expressed in different languages, e.g., PDDL [10], a popular specification language in planning. Furthermore, planning is highly related to synthesis, since both concern games between the environment and the agent. One advantage of utilizing PDDL is that the corresponding state space obtained from PDDL only leads to a single-exponential blowup [14] instead of double-exponential as LTL. An interesting question is how to integrate the synergies of PDDL better-expressed planning and LTL/LTL_f synthesis.

Acknowledgments

We thank the contributions of all the co-authors (in the order of publications): Jianwen Li, Geguang Pu, Lucas M. Tabajara, Moshe Y. Vardi, Giuseppe De Giacomo, Antonio Di Stasio, Giuseppe Perelli, Shengping Xiao, Yingying Shi, Marco Favorito. This work is supported by the ERC Advanced Grant WhiteMech (No. 834228).

References

[1] Benjamin Aminof, Giuseppe De Giacomo, Aniello Murano, and Sasha Rubin. Planning under LTL environment specifications. In *ICAPS*, pages 31–39, 2019.

[2] Alberto Camacho, Meghyn Bienvenu, and Sheila A. McIlraith. Finite LTL synthesis with environment assumptions and quality measures. In *KR*, pages 454–463, 2018.

[3] Giuseppe De Giacomo, Antonio Di Stasio, Giuseppe Perelli, and Shufang Zhu. Synthesis with mandatory stop actions. In *KR*, pages 237–246, 2021.

[4] Giuseppe De Giacomo, Antonio Di Stasio, Luca M. Tabajara, Moshe Y. Vardi, and Shufang Zhu. Finite-trace and generalized-reactivity specifications in temporal synthesis. In *IJCAI*, 2021.

[5] Giuseppe De Giacomo, Antonio Di Stasio, Moshe Y. Vardi, and Shufang Zhu. Two-stage technique for LTL_f synthesis under LTL assumptions. In *KR*, 2020.

[6] Giuseppe De Giacomo, Marco Favorito, Jianwen Li, Moshe Y. Vardi, Shengping Xiao, and Shufang Zhu. LTL_f synthesis as AND-OR graph search: Knowledge compilation at work. In *IJCAI*, pages 2591–2598, 2022.

[7] Giuseppe De Giacomo and Moshe Y. Vardi. Synthesis for LTL and LDL on Finite Traces. In *IJCAI*, 2015.

[8] Bernd Finkbeiner. Synthesis of reactive systems. In *Dependable Software Systems Engineering*, pages 72–98. 2016.

[9] Giuseppe De Giacomo and Moshe Y. Vardi. Linear Temporal Logic and Linear Dynamic Logic on Finite Traces. In *IJCAI*, 2013.

[10] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. Pddl – the planning domain definition language – version 1.2. Technical report, TR-98-003, Yale Center for Computational Vision and Contro, 1998.

[11] Philipp J. Meyer, Salomon Sickert, and Michael Luttenberger. Strix: Explicit reactive synthesis strikes back! In Hana Chockler and Georg Weissenbacher, editors, *CAV*, pages 578–586, 2018.

[12] Amir Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57, 1977.

[13] Amir Pnueli and Roni Rosner. On the Synthesis of a Reactive Module. In *POPL*, 1989.

[14] Jussi Rintanen. Complexity of planning with partial observability. In *ICAPS*, pages 345–354, 2004.

[15] Shengping Xiao, Jianwen Li, Shufang Zhu, Yingying Shi, Geguang Pu, and Moshe Y. Vardi. On-the-fly synthesis for LTL over finite traces. In *AAAI*, pages 6530–6537, 2021.

[16] Shufang Zhu, Giuseppe De Giacomo, Geguang Pu, and Moshe Y. Vardi. LTL_f synthesis with fairness and stability assumptions. In *AAAI*, pages 3088–3095, 2020.

[17] Shufang Zhu, Lucas M. Tabajara, Jianwen Li, Geguang Pu, and Moshe Y. Vardi. Symbolic LTL_f Synthesis. In *IJCAI*, pages 1362–1369, 2017.