Past-present temporal programs over finite traces

Pedro Cabalar[1 [0000-0001-7440-0953]], Martín Diéguez[2 [0000-0003-3440-4348]], François Laferrière[3 [0009-0006-8147-572X]], and Torsten Schaub[3 [0000-0002-7456-041X]]

[1] University of Corunna, Spain
[2] University of Angers, France
[3] University of Potsdam, Germany

Reasoning about dynamic scenarios is a central problem in the areas of Knowledge Representation [5] (KR) and Artificial Intelligence (AI). Several formal approaches and systems have emerged to introduce non-monotonic reasoning features in scenarios where the formalisation of time is fundamental [3, 4, 12, 18, 23]. In *Answer Set Programming* [6] (ASP), former approaches to temporal reasoning use first-order encodings [19] where the time is represented by means of a variable whose value comes from a finite domain. The main advantage of those approaches is that the computation of answer sets can be achieved via incremental solving [17]. Their downside is that they require an explicit representation of time points.

Temporal Equilibrium Logic [2] (TEL) was proposed as a temporal extension of *Equilibrium Logic* [21] with connectives from *Linear Time Temporal Logic* [22] (LTL). Due to the computational complexity of its satisfiability problem (ExpSpace), finding tractable fragments of TEL with good computational properties have also been a topic in the literature. Within this context, *splittable temporal logic programs* [1] have been proved to be a syntactic fragment of TEL that allows for a reduction to LTL via the use of Loop Formulas [15].

When considering incremental solving, logics on finite traces such as LTL_f [11] have been shown to be more suitable. Accordingly, *Temporal Equilibrium Logic on Finite traces* (TEL_f) [8] was created and became the foundations of the temporal ASP solver *telingo* [7].

We present a new syntactic fragment of TEL_f, named *past-present* temporal logic programs. Inspired by Gabbay's seminal paper [16], where the declarative character of past temporal operators is emphasized, this language consists of a set of logic programming rules whose formulas in the head are disjunctions of atoms that reference the present, while in its body we allow for any arbitrary temporal formula without the use of future operators. Such restriction ensures that the past remains independent of the future, which is the case in most dynamic domains, and makes this fragment advantageous for incremental solving.

As a contribution, we study the Lin-Zhao theorem [20] within the context of past-present temporal logic programs. More precisely, we show that when the program is *tight* [13], extending Clark's completion [10, 14] to the temporal case suffices to capture the answer sets of a finite past-present program as the LTL_f-models of a corresponding temporal formula. We also show that, when the program is not tight, the use of loop formulas is necessary. To this purpose, we extend the definition of loop formulas to the case of past-present programs and we prove the Lin-Zhao theorem in our setting. The full version of this paper can be found in [9].

References

1. Aguado, F., Cabalar, P., Pérez, G., Vidal, C.: Loop formulas for splitable temporal logic programs. In: Delgrande, J., Faber, W. (eds.) Proceedings of the Eleventh International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'11). Lecture Notes in Artificial Intelligence, vol. 6645, pp. 80–92. Springer-Verlag (2011)
2. Aguado, F., Cabalar, P., Diéguez, M., Pérez, G., Schaub, T., Schuhmann, A., Vidal, C.: Linear-time temporal answer set programming. Theory Pract. Log. Program. **23**(1), 2–56 (2023)
3. Baral, C., Zhao, J.: Non-monotonic temporal logics for goal specification. In: Veloso, M.M. (ed.) IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007. pp. 236–242 (2007)
4. Baral, C., Zhao, J.: Non-monotonic temporal logics that facilitate elaboration tolerant revision of goals. In: Fox, D., Gomes, C.P. (eds.) Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008. pp. 406–411. AAAI Press (2008)
5. Brachman, R.J., Levesque, H.J.: Knowledge Representation and Reasoning. Elsevier (2004), `http://www.elsevier.com/wps/find/bookdescription.cws\_home/702602/description`
6. Brewka, G., Eiter, T., Truszczyński, M.: Answer set programming at a glance. Communications of the ACM **54**(12), 92–103 (2011)
7. Cabalar, P., Kaminski, R., Morkisch, P., Schaub, T.: telingo = ASP + time. In: Balduccini, M., Lierler, Y., Woltran, S. (eds.) Proceedings of the Fifteenth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'19). Lecture Notes in Artificial Intelligence, vol. 11481, pp. 256–269. Springer-Verlag (2019)
8. Cabalar, P., Kaminski, R., Schaub, T., Schuhmann, A.: Temporal answer set programming on finite traces. Theory and Practice of Logic Programming **18**(3-4), 406–420 (2018)
9. Cabalar, P., Diéguez, M., Laferrière, F., Schaub, T.: Past-present temporal programs over finite traces (2023), `https://arxiv.org/pdf/2307.12620.pdf`
10. Clark, K.: Negation as failure. In: Gallaire, H., Minker, J. (eds.) Logic and Data Bases, pp. 293–322. Plenum Press (1978)
11. De Giacomo, G., Vardi, M.: Linear temporal logic and linear dynamic logic on finite traces. In: Rossi, F. (ed.) Proceedings of the Twenty-third International Joint Conference on Artificial Intelligence (IJCAI'13). pp. 854–860. IJCAI/AAAI Press (2013)
12. Emerson, E.: Temporal and modal logic. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, pp. 995–1072. MIT Press (1990)
13. Erdem, E., Lifschitz, V.: Tight logic programs. Theory and Practice of Logic Programming **3**(4-5), 499–518 (2003)

14. Fages, F.: Consistency of Clark's completion and the existence of stable models. Journal of Methods of Logic in Computer Science **1**, 51–60 (1994)
15. Ferraris, P., Lee, J., Lifschitz, V.: A generalization of the Lin-Zhao theorem. Annals of Mathematics and Artificial Intelligence **47**(1-2), 79–101 (2006)
16. Gabbay, D.: The declarative past and imperative future: Executable temporal logic for interactive systems. In: Banieqbal, B., Barringer, H., Pnueli, A. (eds.) Proceedings of the Conference on Temporal Logic in Specification. Lecture Notes in Computer Science, vol. 398, pp. 409–448. Springer-Verlag (1987)
17. Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Thiele, S.: Engineering an incremental ASP solver. In: Garcia de la Banda, M., Pontelli, E. (eds.) Proceedings of the Twenty-fourth International Conference on Logic Programming (ICLP'08). Lecture Notes in Computer Science, vol. 5366, pp. 190–205. Springer-Verlag (2008)
18. González, G., Baral, C., Cooper, P.A.: Modeling Multimedia Displays Using Action Based Temporal Logic, pp. 141–155. Springer US, Boston, MA (2002)
19. Lifschitz, V.: Answer set planning. In: de Schreye, D. (ed.) Proceedings of the International Conference on Logic Programming (ICLP'99). pp. 23–37. MIT Press (1999)
20. Lin, F., Zhao, J.: On tight logic programs and yet another translation from normal logic programs to propositional logic. In: Gottlob, G., Walsh, T. (eds.) Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03). pp. 853–858. Morgan Kaufmann Publishers (2003)
21. Pearce, D.: A new logical characterisation of stable models and answer sets. In: Dix, J., Pereira, L., Przymusinski, T. (eds.) Proceedings of the Sixth International Workshop on Non-Monotonic Extensions of Logic Programming (NMELP'96). Lecture Notes in Computer Science, vol. 1216, pp. 57–70. Springer-Verlag (1997)
22. Pnueli, A.: The temporal logic of programs. In: Proceedings of the Eight-teenth Symposium on Foundations of Computer Science (FOCS'77). pp. 46–57. IEEE Computer Society Press (1977)
23. Sandewall, E.: Features and fluents: the representation of knowledge about dynamical systems, vol. 1. Oxford University Press, New York, NY, USA (1994)