

Dynamic and Temporal Answer Set Programming on Linear Finite Traces*

Pedro Cabalar

University of Corunna, Spain

Torsten Schaub

University of Potsdam, Germany

The logical foundations of Answer Set Programming (ASP; (Lifschitz, 1999)) rest upon the logic of Here-and-There (HT; (Heyting, 1930)), or more precisely its equilibrium models (Pearce, 1997) that correspond to stable models semantics (Gelfond and Lifschitz, 1988). For defining extensions to ASP from firm logical principles, it has thus become good practice to first elaborate upon equilibrium models for an extended HT setting in order to consider afterwards the respective language fragments of that setting that are well suited in the context of logic programming.

This avenue was also followed in (Cabalar and Vega, 2007), which gave rise to the temporal extension of HT called *Temporal Here-and-There* (THT) and its non-monotonic counterpart *Temporal Equilibrium Logic* (TEL (Aguado et al., 2013)). More precisely, TEL builds upon an extension of the logic of HT with Linear Temporal Logic (LTL; (Pnueli, 1977)) sharing the syntax of the latter. This results in an expressive non-monotonic modal logic, which extends traditional temporal logic programming approaches (Cabalar, Diéguez, and Vidal, 2015) to the general syntax of LTL and possesses a computational complexity beyond LTL (Bozzelli and Pearce, 2015). As in LTL, a model in TEL is an *infinite* sequence of states, called a *trace*. These infinite traces are convenient for describing formal properties of reactive systems and, as in LTL, can be processed by model checking techniques based on automata construction (Cabalar and Demri, 2011; Cabalar and Diéguez, 2011). However, the usual temporal reasoning problems treated in the ASP literature are simulation, explanation and planning under the assumption of a finite narrative. In that sense, temporal approaches based on finite traces (Baier and McIlraith, 2006; De Giacomo and Vardi, 2013) seem more aligned with the way in which ASP domain representation and solving technology deals with temporal problems.

To capture this feature, we recently proposed in (Cabalar et al., 2018) an alternative combination of the logics of HT and LTL whose semantics rests upon *finite* traces. On the one hand, this amounts to a restriction of THT and TEL to finite traces. On the other hand, this is similar to the restriction of LTL to LTL_f advocated by (De Giacomo and Vardi, 2013); see also (Baier and McIlraith, 2006). Our new ap-

proach, dubbed TEL_f , has the following advantages. First, it is readily implementable via ASP technology. Second, it can be reduced to a normal form which is close to logic programs and much simpler than the one obtained for TEL. Finally, its temporal models are finite and offer a one-to-one correspondence to plans. Interestingly, TEL_f also sheds light on concepts and methodology used in incremental ASP solving when understanding incremental parameters as time points.

Another distinctive feature of TEL_f is the inclusion of future as well as past temporal operators. As an example, the temporal rule:

$$\Box(\text{shoot} \wedge \blacklozenge\text{shoot} \wedge \blacksquare\text{unloaded} \rightarrow \diamond\text{fail})$$

combines future operators (\Box , \diamond) with past operators (\blacklozenge , \blacksquare) and expresses that “it is always the case that if we shoot at least twice some gun that has never been loaded then the gun will eventually fail.” We associate this combination of future and past operators with the following benefits. When using the causal reading of program rules, it is generally more natural to draw upon the past in rule bodies and to refer to the future in rule heads. A similar argument was put forward by Gabbay (1987) in his proposal of “declarative past and imperative future.” This format also yields a simpler normal form and lends itself to a systematic modeling methodology which favors the definition of states in terms of the past rather than mixing in future operators. For instance, in reasoning about actions, the idea is to derive action effects for the current state and check their preconditions in the previous one, rather than to represent this as a transition from the current to the next state. This methodology aligns state constraints, effect axioms, etc. to capture the present state. As well, computationally, past operators are much easier to handle than their future counterparts when it comes to incremental reasoning, since they refer to already computed knowledge.

TEL_f is implemented in the `telingo` system (Cabalar et al., 2019), extending the ASP system `clingo` to compute the temporal stable models of (non-ground) temporal logic programs. To this end, it extends the full-fledged input language of `clingo` with temporal operators and computes temporal models incrementally by multi-shot solving (Gebser et al., 2019) using a modular translation into

*This is a revised version of the one submitted to *Actions@KR18*.

ASP. `telingo` is freely available at [github](https://github.com/potassco/telingo)¹. The interested reader might have a good time playing with the examples given in the `examples` folder at the same site.

Similar to the extension of LTL_f to its (linear) dynamic logic counterpart LDL_f (De Giacomo and Vardi, 2013), we introduced in (Bossler et al., 2018) a dynamic extension of HT that draws up upon this linear version of dynamic logic. As an example of expression using this logic, the formula:

$$[(\neg help)^*](\neg help \rightarrow sos)$$

represents the policy “keep sending an SOS while no help is perceived.” We elaborate upon its restriction to finite traces in (Cabalar, Diéguez, and Schaub, 2019). We refer to the resulting logic as (*Linear*) *Dynamic logic of Here-and-There* (DHT for short). As usual, the equilibrium models of DHT are used to define temporal stable models and induce the non-monotonic counterpart of DHT, referred to as (*Linear*) *Dynamic Equilibrium Logic* (DEL). In doing so, we actually parallel earlier work extending HT with LTL, ultimately leading to THT and TEL.

In fact, we show that THT (and its equilibrium counterpart TEL) can be embedded into our new logic DHT (and DEL, respectively) — just as LTL can be put in LDL. Moreover, we prove that the satisfiability problem in DEL is EXPSpace-complete; it thus coincides with that of TEL but goes beyond that of LDL and LTL, both being PSPACE-complete. In fact, the membership part of this result is obtained by means of an automata-based method for computing DEL models. Finally, we show that the monotonic base logic of DEL, namely DHT, allows us to decide strong equivalence in DEL; this reinforces the adequacy of the relation between both logics.

In the context of the version of DEL for finite traces, DEL_f , we developed a translation of any (converse-free) arbitrary DEL_f theory into a propositional theory (under the semantics of HT) and in turn into a logic program. This translation has turned out to be non-trivial: it is based on unfolding path expressions, something potentially equivalent to the execution of a sequential program. Termination is guaranteed by some preprocessing steps for normalizing path expressions.

These recent results open several interesting topics for future study. As an open topic, it would be interesting to adapt existing model checking techniques (based on automata construction) for temporal logics to solve the problem of existence of temporal stable models. This was done for infinite traces in (Cabalar and Diéguez, 2011; Cabalar and Demri, 2011), but no similar method has been implemented for finite traces on TEL_f or DEL_f yet. The importance of having an efficient implementation of such a method is that it would allow deciding non-existence of a plan in a given planning problem, something not possible by current incremental solving techniques. Another interesting topic is the optimization of grounding in temporal ASP specifications as those handled by `telingo`. The current grounding of `telingo` is inherited from incremental solving in `clingo` and does not exploit the semantics of temporal expressions

that are available now in the input language. Finally, we envisage to extend the `telingo` system with features of DEL in order to obtain a powerful system for representing and reasoning about dynamic domains, not only providing an effective implementation of TEL and DEL but, furthermore, a platform for action and control languages, like $\mathcal{A}, \mathcal{B}, \mathcal{C}$ (Gelfond and Lifschitz, 1998; Giunchiglia et al., 2004) or GOLOG (Levesque et al., 1997).

References

- Aguado, F.; Cabalar, P.; Diéguez, M.; Pérez, G.; and Vidal, C. 2013. Temporal equilibrium logic: a survey. *Journal of Applied Non-Classical Logics* 23(1-2):2–24.
- Baier, J., and McIlraith, S. 2006. Planning with first-order temporally extended goals using heuristic search. In Gil, Y., and Mooney, R., eds., *Proceedings of the Twenty-first National Conference on Artificial Intelligence (AAAI’06)*, 788–795. AAAI Press.
- Balduccini, M.; Lierler, Y.; and Woltran, S., eds. 2019. *Proceedings of the Fifteenth International Conference on Logic Programming and Nonmonotonic Reasoning (LP-NMR’19)*, volume 11481 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag.
- Bossler, A.; Cabalar, P.; Diéguez, M.; and Schaub, T. 2018. Introducing temporal stable models for linear dynamic logic. In Thielscher, M.; Toni, F.; and Wolter, F., eds., *Proceedings of the Sixteenth International Conference on Principles of Knowledge Representation and Reasoning (KR’18)*, 12–21. AAAI Press.
- Bozzelli, L., and Pearce, D. 2015. On the complexity of temporal equilibrium logic. In *Proceedings of the Thirtieth Annual Symposium on Logic in Computer Science (LICS’15)*, 645–656. IEEE Computer Society Press.
- Cabalar, P., and Demri, S. 2011. Automata-based computation of temporal equilibrium models. In Vidal, G., ed., *Proceedings of the Twenty-first International Symposium on Logic-Based Program Synthesis and Transformation (LOPSTR’11)*, volume 7225 of *Lecture Notes in Computer Science*, 57–72. Springer-Verlag.
- Cabalar, P., and Diéguez, M. 2011. STELP — a tool for temporal answer set programming. In Delgrande, J., and Faber, W., eds., *Proceedings of the Eleventh International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR’11)*, volume 6645 of *Lecture Notes in Artificial Intelligence*, 370–375. Springer-Verlag.
- Cabalar, P., and Vega, G. P. 2007. Temporal equilibrium logic: A first approach. In Moreno-Díaz, R.; Pichler, F.; and Quesada-Arencibia, A., eds., *Proceedings of the Eleventh International Conference on Computer Aided Systems Theory (EUROCAST’17)*, volume 4739 of *Lecture Notes in Computer Science*, 241–248. Springer-Verlag.
- Cabalar, P.; Kaminski, R.; Schaub, T.; and Schuhmann, A. 2018. Temporal answer set programming on finite traces. *Theory and Practice of Logic Programming* 18(3-4):406–420.

¹<https://github.com/potassco/telingo>

- Cabalar, P.; Kaminski, R.; Morkisch, P.; and Schaub, T. 2019. *telingo = ASP + time*. In Balduccini et al. Balduccini, Lierler, and Woltran (2019), 256–269.
- Cabalar, P.; Diéguez, M.; and Vidal, C. 2015. An infinitary encoding of temporal equilibrium logic. *Theory and Practice of Logic Programming* 15(4-5):666–680.
- Cabalar, P.; Diéguez, M.; and Schaub, T. 2019. Towards dynamic answer set programming over finite traces. In Balduccini et al. Balduccini, Lierler, and Woltran (2019), 148–162.
- De Giacomo, G., and Vardi, M. 2013. 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)*, 854–860. IJCAI/AAAI Press.
- Gabbay, D. 1987. The declarative past and imperative future: Executable temporal logic for interactive systems. In Banieqbal, B.; Barringer, H.; and Pnueli, A., eds., *Proceedings of the Conference on Temporal Logic in Specification*, volume 398 of *Lecture Notes in Computer Science*, 409–448. Springer-Verlag.
- Gebser, M.; Kaminski, R.; Kaufmann, B.; and Schaub, T. 2019. Multi-shot ASP solving with clingo. *Theory and Practice of Logic Programming* 19(1):27–82.
- Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In Kowalski, R., and Bowen, K., eds., *Proceedings of the Fifth International Conference and Symposium of Logic Programming (ICLP'88)*, 1070–1080. MIT Press.
- Gelfond, M., and Lifschitz, V. 1998. Action languages. *Electronic Transactions on Artificial Intelligence* 3(6):193–210.
- Giunchiglia, E.; Lee, J.; Lifschitz, V.; McCain, N.; and Turner, H. 2004. Nonmonotonic causal theories. *Artificial Intelligence* 153(1-2):49–104.
- Heyting, A. 1930. Die formalen Regeln der intuitionistischen Logik. In *Sitzungsberichte der Preussischen Akademie der Wissenschaften*. Deutsche Akademie der Wissenschaften zu Berlin. 42–56. Reprint in *Logik-Texte: Kommentierte Auswahl zur Geschichte der Modernen Logik*, Akademie-Verlag, 1986.
- Levesque, H.; Reiter, R.; Lespérance, Y.; Lin, F.; and Scherl, R. 1997. GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming* 31(1-3):59–83.
- Lifschitz, V. 1999. Answer set planning. In de Schreye, D., ed., *Proceedings of the International Conference on Logic Programming (ICLP'99)*, 23–37. MIT Press.
- Pearce, D. 1997. A new logical characterisation of stable models and answer sets. In Dix, J.; Pereira, L.; and Przytusinski, T., eds., *Proceedings of the Sixth International Workshop on Non-Monotonic Extensions of Logic Programming (NMELP'96)*, volume 1216 of *Lecture Notes in Computer Science*, 57–70. Springer-Verlag.
- Pnueli, A. 1977. The temporal logic of programs. In *Proceedings of the Eighteenth Symposium on Foundations of Computer Science (FOCS'77)*, 46–57. IEEE Computer Society Press.