Metric Dynamic Equilibrium Logic*

Arvid Becker[1], Pedro Cabalar[2], Martín Diéguez[3], Luis Fariñas[4], Anna Schuhmann[1], and Torsten Schaub[1]

[1] University of Potsdam, Germany
[2] University of Corunna, Spain
[3] Université d'Angers, France
[4] Université de Toulouse, France

Abstract. In temporal extensions of Answer Set Programming (ASP) based on linear-time, the behavior of dynamic systems is captured by sequences of states. While this representation reflects their relative order, it abstracts away the specific times associated with each state. In many applications, however, time constraints are important, for instance, when planning and scheduling go hand in hand. We address this by developing a metric extension of linear-time Dynamic Equilibrium Logic, in which dynamic operators are constrained by intervals over integers. The resulting Metric Dynamic Equilibrium Logic provides the foundation of an ASP-based approach for specifying qualitative and quantitative dynamic constraints. As such, it constitutes the most general among a whole spectrum of temporal extensions of Equilibrium Logic.

Keywords: Linear Temporal Logic · Linear Dynamic Logic · Metric Temporal logic · Logic of Here-and-There · Equilibrium Logic.

Reasoning about action and change, or more generally reasoning about dynamic systems, often requires both qualitative and quantitative dynamic constraints. Whereas with qualitative dynamic constraints the order of events can be specified, quantitative dynamic constraints are crucial in case effects of actions need to meet deadlines.

An important first step to address qualitative dynamic constraints in an ASP setting was the combination of Linear Temporal Logic (LTL [16]), with the base logic of Answer Set Programming (ASP [14]), namely, the logic of Here-and-There (HT [12]) and its non-monotonic extension, called Equilibrium Logic [15]. This combination gave birth to a temporal extension of Equilibrium Logic called Temporal Equilibrium Logic (TEL [8, 2, 7, 1]) which serves as the semantics of the temporal ASP system *telingo* [6, 3] extending the ASP system *clingo* [10].

* This work was supported by MICINN, Spain, grant PID2020-116201GB-I00, Xunta de Galicia, Spain (GPC ED431B 2019/03), Fundación BBVA (LIANDA research project), Spain, Région Pays de la Loire, France (EL4HC and étoiles montantes CTASP), DFG grant SCHA 550/15, Germany, and European Union COST action CA-17124.

In temporal logic, we can use the formula $\square(use \rightarrow \Diamond clean)$ to express the qualitative dynamic constraint that a machine has to be eventually cleaned after being used.

In preceding work, we explored extending the LTL syntax in two different directions. On the one hand, we studied the use of path expressions from Dynamic Logic [11], adopting the syntax and principles of Linear Dynamic Logic (LDL) [9]. This lead to the extensions called *Dynamic HT* (DHT) and *Dynamic Equilibrium Logic* (DEL) that, despite allowing a richer syntax and expressiveness, used the same semantic structures as LTL, since a temporal stable model is just a (finite or infinite) trace.

A commonality of dynamic and temporal logics is that they abstract from specific time points when capturing temporal relationships. Consequently, nothing can be said about the delay between using and cleaning the machine. To address this lack of expressivity we studied in a somehow orthogonal way the incorporation of metric information in Temporal Here-and-There (THT), by extending modal operators with time intervals as in Metric Temporal Logic (MTL) [13], leading to *Metric HT* (MHT) and *Metric Equilibrium Logic* (MEL) [5]. We continued to maintain the same linear-time semantics, embodied by sequences of states, when elaborating upon a first "light-weight" metric temporal extension of HT [4]. The "light-weightiness" is due to treating time as a state counter by identifying the next timepoint with the next state. For instance, this allows us to refine our example by stating that, if the machine is used, it has to be cleaned within the next 3 states, viz. $\square(use \rightarrow \Diamond_{(1..3)} clean)$. Although this permits the restriction of temporal operators to subsequences of states, no fine-grained timing constraints are expressible.

In [5], we filled this gap in the context of temporal logic by associating each state with its *time*, as done in Metric Temporal Logic (MTL [13]). This resulted in a metric temporal extension of HT, referred to as MEL. It allows us to measure time differences between events. For instance, in our example, we may thus express that whenever the machine is used, it has to be cleaned within 60 to 120 time units, by writing $\square(use \rightarrow \Diamond_{(60..120)} clean)$. Unlike the non-metric version, this stipulates that once *use* is true in a state, *clean* must be true in some future state whose associated time is at least 60 and at most 120 time units after the time of *use*. The choice of time domain is crucial, and might even lead to undecidability in the continuous case (that is, using real numbers). We rather adopt a discrete approach that offers a sequence of snapshots of a dynamic system.

In our work, we combine the aforementioned temporal, dynamic, and (time-based) metric extensions of the logic of Here-and-There and its non-monotonic extension Equilibrium logic within a single logical setting by extending the dynamic variants with time-based metrics. This results in the Metric Dynamic logic of Here-and-There (MDHT) and its non-monotonic extension of Metric Dynamic Equilibrium Logic (MDEL). As in the classical case, we may formulate temporal, metric, as well as Boolean operators in terms of metric dynamic ones. This already hints at the great expressive power of MDHTand MDEL.

References

1. Aguado, F., Cabalar, P., Diéguez, M., Pérez, G., Schaub, T., Schuhmann, A., Vidal, C.: Linear-time temporal answer set programming. Theory and Practice of Logic Programming pp. 1–55 (2021). https://doi.org/10.1017/S1471068421000557
2. Aguado, F., Cabalar, P., Diéguez, M., Pérez, G., Vidal, C.: Temporal equilibrium logic: a survey. Journal of Applied Non-Classical Logics **23**(1-2), 2–24 (2013). https://doi.org/10.1080/11663081.2013.798985
3. Cabalar, P., Diéguez, M., Laferriere, F., Schaub, T.: Implementing dynamic answer set programming over finite traces. In: De Giacomo, G., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J. (eds.) Proceedings of the Twenty-fourth European Conference on Artificial Intelligence (ECAI'20). pp. 656–663. IOS Press (2020). https://doi.org/10.3233/FAIA200151
4. Cabalar, P., Diéguez, M., Schaub, T., Schuhmann, A.: Towards metric temporal answer set programming. Theory and Practice of Logic Programming **20**(5), 783–798 (2020)
5. Cabalar, P., Diéguez, M., Schaub, T., Schuhmann, A.: Metric temporal answer set programming over timed traces. In: Inclezan, G.G.D., Maratea, M. (eds.) Proceedings of the Sixteenth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'22). Lecture Notes in Artificial Intelligence, Springer-Verlag (2022)
6. 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). https://doi.org/10.1007/978-3-030-20528-7“˙19
7. 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). https://doi.org/10.1017/S1471068418000297
8. Cabalar, P., Vega, G.P.: Temporal equilibrium logic: A first approach. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) Proceedings of the Eleventh International Conference on Computer Aided Systems Theory (EUROCAST'17). Lecture Notes in Computer Science, vol. 4739, pp. 241–248. Springer-Verlag (2007)
9. 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)
10. Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Wanko, P.: Theory solving made easy with clingo 5. In: Carro, M., King, A. (eds.) Technical Communications of the Thirty-second International Conference on Logic Programming (ICLP'16). OpenAccess Series in Informatics (OASIcs), vol. 52, pp. 2:1–2:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2016)
11. Harel, D., Tiuryn, J., Kozen, D.: Dynamic Logic. MIT Press (2000). https://doi.org/10.1145/568438.568456
12. Heyting, A.: Die formalen Regeln der intuitionistischen Logik. In: Sitzungsberichte der Preussischen Akademie der Wissenschaften, pp. 42–56. Deutsche Akademie der Wissenschaften zu Berlin (1930)
13. Koymans, R.: Specifying real-time properties with metric temporal logic. Real-Time Systems **2**(4), 255–299 (1990)

14. 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)
15. 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). https://doi.org/10.1007/BFb0023801
16. 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). https://doi.org/10.1109/SFCS.1977.32