Two-sorted Dynamic Here-and-There

Arvid Becker[1], Pedro Cabalar[2], Martín Diéguez[3], and Torsten Schaub[1]

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

1 Introduction

Representing and reasoning about dynamic systems is a key problem in Artificial Intelligence and beyond. Among the various formal systems to model dynamic domains are action languages [5, 8], whose appeal lies in their elegant syntactic and semantic simplicity: they usually consist of static and dynamic laws inducing an unique transition system. A key element of the language is the distinction between action and fluents, leading to a transition system that displays actions as labeled transitions.

Another formalism distinguishing between actions and fluents is Dynamic Logic (DL, [7]). DL is tailored to reason about actions and effects as its language entails modalities that are parametrised by regular expressions, that can be regarded as an abstraction of sequences of actions.

Several approaches have successfully used temporal logics, especially Linear Temporal Logic (LTL), to model domain-dependent information in planning ([2]). This approach led to very efficient implementations ([3]). Opposed to DL there is no distinction between action and fluents, leading to traces where actions and fluents occur together.

In the realm of combining the trace based approach of LTL and the suitability of DL operators to reason about actions and effects Linear Dynamic Logic (LDL) was invented ([4]). Although highly expressive, LDL does not differentiate between actions and fluents. Reintroducing it foremost supports clarity. Separating actions and fluents makes the model more understandable and interpretable. It provides a clear distinction between what can be done (actions) and what can be observed or evaluated (fluents). Our aim is to combine LTL and *two-sorted* DL on a common nonmonotonic extension.

2 Approach

Differing from the original approach to (one-sorted) Dynamic Here-and-There (DHT) we consider a two-sorted alphabet $(\mathcal{A}, \mathcal{P})$, consisting of a set $\mathcal{A}$ of action variables and a disjoint set $\mathcal{P}$ of state variables (also referred to as fluents). A *two-sorted dynamic formula* φ and *path expressions* ρ are then mutually defined by the following pair of grammar rules:

$$\varphi ::= p \in \mathcal{P} \mid \bot \mid [\rho]\,\varphi \mid \langle\rho\rangle\,\varphi \qquad \rho ::= a \in \mathcal{A} \mid \varphi? \mid \rho + \rho \mid \rho \,;\, \rho \mid \rho^* \mid \rho^- \ .$$

The models of *two-sorted DHT* are described in terms of labeled traces. A labeled *HT-trace* of length λ over alphabet $(\mathcal{A}, \mathcal{P})$ is a pair $(\langle \mathbf{H}, \mathbf{T} \rangle, \alpha)$ consisting of

- an *HT-trace* $\langle \mathbf{H}, \mathbf{T} \rangle = (\langle H_i, T_i \rangle)_{i \in [0..\lambda)}$ and
- a surjective function $\alpha : \mathcal{A} \to I_\lambda$, where $I_\lambda = \{(i, i+1) \mid i, i+1 \in [0..\lambda)\}$.

Imposing surjectivity on the labeling function guarantees that at least one action occurs in each label. A particular case of interest are HT-traces labeled with singelton sets of actions, since they correspond to sequential plans. Such restriction leads to a version of [6] and [1].

Definition 1 (two-sorted DHT satisfaction). *A labeled HT-trace* $\mathbf{M} = (\langle \mathbf{H}, \mathbf{T} \rangle, \alpha)$ *of length* λ satisfies *a two-sorted dynamic formula* φ *at time point* $k \in [0..\lambda)$*, written* $\mathbf{M}, k \models_\ell \varphi$*, if*

1. $\mathbf{M}, k \models \top$ *and* $\mathbf{M}, k \not\models \bot$
2. $\mathbf{M}, k \models p$ *if* $p \in H_k$ *for any atom* $p \in \mathcal{P}$
3. $\mathbf{M}, k \models \langle \rho \rangle \varphi$ *if* $\mathbf{M}, i \models \varphi$ *for some* i *with* $(k, i) \in \|\rho\|^{\mathbf{M}}$
4. $\mathbf{M}, k \models [\rho] \varphi$ *if* $\mathbf{M}', i \models \varphi$ *for all* i *with* $(k, i) \in \|\rho\|^{\mathbf{M}'}$
   for both $\mathbf{M}' = \mathbf{M}$ *and* $\mathbf{M}' = \langle \mathbf{T}, \mathbf{T} \rangle$

where, for any labeled HT-trace $(\mathbf{M}, \alpha)$ $\|\rho\|^{\mathbf{M}} \subseteq \mathbb{N}^2$ *is a relation on pairs of time points inductively defined as follows.*

5. $\|a\|^{\mathbf{M}} \stackrel{def}{=} \alpha(a)$
6. $\|\varphi?\|^{\mathbf{M}} \stackrel{def}{=} \{(k, k) \mid \mathbf{M}, k \models \varphi\}$
7. $\|\rho_1 + \rho_2\|^{\mathbf{M}} \stackrel{def}{=} \|\rho_1\|^{\mathbf{M}} \cup \|\rho_2\|^{\mathbf{M}}$
8. $\|\rho_1 \,;\, \rho_2\|^{\mathbf{M}} \stackrel{def}{=} \{(k, i) \mid (k, j) \in \|\rho_1\|^{\mathbf{M}} \text{and } (j, i) \in \|\rho_2\|^{\mathbf{M}} \text{for some } j\}$
9. $\|\rho^*\|^{\mathbf{M}} \stackrel{def}{=} \bigcup_{n \geq 0} \|\rho^n\|^{\mathbf{M}}$
10. $\|\rho^-\|^{\mathbf{M}} \stackrel{def}{=} \{(k, i) \mid (i, k) \in \|\rho\|^{\mathbf{M}}\}$

It is important to note that our approach is a first step towards a respective ASP extension. As *two-sorted* DHT is still monotonic one still has to define *two-sorted* Dynamic Equilibrium models, leading to the nonmonotonic *two-sorted* Dynamic Equilibrium Logic (DEL). This will be interesting as, opposed to *one-sorted* DEL, actions are not part of the minimisation anymore.

As an example of our language consider the Yale Shooting scenario with the set of actions $\mathcal{A} = \{load, shoot\}$ and the set of fluents $\mathcal{P} = \{loaded, alive\}$. Assuming singleton sets of action labels, the following formulas could be used to express the dynamic laws of the system:

$$P \vee \neg P \tag{1}$$

$$\Box([load]\, loaded) \tag{2}$$

$$\Box([shoot]\, \neg loaded) \tag{3}$$

$$\Box([loaded?; shoot]\, \neg alive) \tag{4}$$

By labeling the transtion to a fluent set by its specified cause, the respective labeled traces precisely show the corresponding effects of the actions.

References

[1] F. Aguado, G. Pérez, and C. Vidal. "Integrating Temporal Extensions of Answer Set Programming". In: *Proceedings of the Twelfth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'13)*. Ed. by P. Cabalar and T. Son. Vol. 8148. Lecture Notes in Artificial Intelligence. Springer-Verlag, 2013, pp. 23–35. URL: https://doi.org/10.1007/978-3-642-40564-8_3.

[2] F. Bacchus and F. Kabanza. "Using temporal logics to express search control knowledge for planning". In: *Artificial Intelligence* 116.1-2 (2000), pp. 123–191.

[3] J. Baier et al. "Beyond Classical Planning: Procedural Control Knowledge and Preferences in State-of-the-Art Planners". In: *Proceedings of the Twenty-third National Conference on Artificial Intelligence (AAAI'08)*. Ed. by D. Fox and C. Gomes. AAAI Press, 2008, pp. 1509–1512.

[4] G. De Giacomo and M. Vardi. "Linear Temporal Logic and Linear Dynamic Logic on Finite Traces". In: *Proceedings of the Twenty-third International Joint Conference on Artificial Intelligence (IJCAI'13)*. Ed. by F. Rossi. IJCAI/AAAI Press, 2013, pp. 854–860.

[5] M. Gelfond and V. Lifschitz. "Action languages". In: *Electronic Transactions on Artificial Intelligence* 3.6 (1998), pp. 193–210.

[6] L. Giordano, A. Martelli, and D. Theseider Dupré. "Reasoning about actions with Temporal Answer Sets". In: *Theory and Practice of Logic Programming* 13.2 (2013), pp. 201–225.

[7] D. Harel, J. Tiuryn, and D. Kozen. *Dynamic Logic*. MIT Press, 2000.

[8] E. Sandewall. *Features and fluents: the representation of knowledge about dynamical systems*. Vol. 1. New York, NY, USA: Oxford University Press, 1994.