Towards Temporal Answer Set Programming with Preferences (Extended Abstract)

Javier Romero, Torsten Schaub and Hannah Steinbach

University of Potsdam, Germany

We present our current work combining Temporal Answer Set Programming (Aguado et al. 2023) with the Asprin framework for preferences (Brewka, Delgrande, et al. 2023). Answer Set Programming (ASP; Lifschitz 2008) provides a declarative basis for knowledge representation and reasoning. Its temporal extension, based on Temporal Equilibrium Logic, supports reasoning about time, while Asprin offers a flexible language for expressing preferences. Their integration yields a general approach to optimization in temporal ASP. In this note, we outline the formalization of the approach, and describe a prototype implementation that combines meta-programming with the *asprin* system.

The language of ASP includes *minimize statements* that declare preference relations over sets of atoms (Simons, Niemelä, and Soininen 2002).[1] For example,

$$\#\mathit{minimize}\{1 : a\}$$

declares the preference relation $\succeq$ such that $X \succeq Y$ iff either $a \notin X$ or $a \in Y$. Beyond minimize statements, several other preference formalisms have been proposed in the literature, including subset preferences, answer set optimization (Brewka, Niemelä, and Truszczyński 2003), poset preferences (Rosa, Giunchiglia, and Maratea 2010), and CP-nets (Boutilier et al. 2004). The Asprin framework provides a general approach to represent and combine these and other preferences in ASP.

In Asprin, each preference formalism is defined as a preference type, which maps sets of preference elements to preference relations. For example, in *asprin*'s library of preferences, minimize statements are represented by preference type $\mathit{less}(\mathit{weight})$. This type maps the set of preference elements $\{1 : a\}$ to the relation $\succeq$ above, which can be declared by the following preference statement of name p:

$$\#\mathit{preference}(p, \mathit{less}(\mathit{weight}))\{1 : a\}.$$

Multiple preference statements can be combined using composite types such as *pareto* or *lexico*. Together, they form a preference specification, which declares a preference relation analogous to that induced by minimize statements.

Linear Temporal Logic (LTL; Kamp 1968) is a well-established formalism for reasoning about time. Temporal Equilibrium Logic (TEL) is a nonmonotonic version of LTL that preserves its syntax but interprets it under a temporal variant of the semantics of ASP. This variant is defined in terms of traces, that

[1] A preference relation $\succeq$ is a transitive and reflexive relation.

are (possibly infinite) sequences of states, each of which is a set of atoms. A temporal answer set of a temporal theory is a trace that satisfies the formulas of the theory and a minimality condition analogous to the one of ASP. A fragment of TEL, called temporal logic programs, allows rules with temporal operators and has been implemented (for finite traces) in the solver *telingo* (Cabalar et al. 2019). As an example, this is a temporal logic program

$$a \vee \neg a \leftarrow \qquad\qquad \widehat{\circ}\,\Box(a \leftarrow \bullet a)$$

that guesses a at the initial state and enforces its persistence in later states. Its temporal answer sets are either sequences of empty states or sequences of $\{a\}$'s.

The combination of TEL with Asprin preferences is straightforward. In ASP, a logic program has a set of answer sets, a minimize statement defines a preference relation over sets of atoms, and an answer set is optimal if there is no other answer set preferred to it. In our approach to temporal ASP with preferences, a temporal theory has a set of temporal answer sets, a preference specification declares a preference relation over traces, and a temporal answer set is optimal if there is no other temporal answer set preferred to it. For finite traces, we also consider k-optimal temporal answer sets: those of length $\leq k$ such that there is no other temporal answer set of length $\leq k$ preferred to them.

The formal definition keeps temporal theories as in TEL, and extends preference specifications to accommodate temporal formulas. Preference types map sets of preference elements to preference relations over traces, and preference specifications declare those relations. For example, the following statement q declares that traces where a never holds are preferred to the others:

$$\#\mathit{preference}(q, \mathit{less}(\mathit{weight}))\{1 : \Diamond a\}.$$

Together with the previous temporal program, it selects as optimal the temporal answer sets whose states are all empty.

Our implementation[2] combines meta-programming with the *asprin* system. The input is a temporal logic program with a preference specification, where temporal operators are represented by (possibly nested) function symbols. Given a trace length k, it computes k-optimal temporal answer sets. It starts by grounding the temporal logic program and reifying it into a set of facts, using the ASP solver *clingo* (Kaminski et al. 2023). A meta-encoding interprets these facts and generates temporal answer sets of length $\leq k$. This meta-encoding also defines when a temporal answer set satisfies a temporal formula from the preference specification. In this way, the reified program, the meta-encoding, and the preference specification can be combined to compute k-optimal temporal answer sets using *asprin*. This implementation takes advantage of all preference types in *asprin*'s library, which can now be used for temporal reasoning.

[2] Available at `https://github.com/krr-up/t-asprin`.

References

Aguado, F., P. Cabalar, M. Diéguez, G. Pérez, T. Schaub, A. Schuhmann, and C. Vidal (2023). "Linear-Time Temporal Answer Set Programming". In: *Theory and Practice of Logic Programming* 23.1, pp. 2–56. DOI: 10.1017/S1471068421000557.

Boutilier, C., R. Brafman, C. Domshlak, H. Hoos, and D. Poole (2004). "CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements". In: *Journal of Artificial Intelligence Research* 21, pp. 135–191. DOI: 10.1613/jair.1234.

Brewka, G., J. Delgrande, J. Romero, and T. Schaub (2023). "A general framework for preferences in answer set programming". In: *Artificial Intelligence* 325, p. 104023.

Brewka, G., I. Niemelä, and M. Truszczyński (2003). "Answer Set Optimization". In: *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*. Ed. by G. Gottlob and T. Walsh. Morgan Kaufmann Publishers, pp. 867–872.

Cabalar, P., R. Kaminski, P. Morkisch, and T. Schaub (2019). "telingo = ASP + Time". In: *Proceedings of the Fifteenth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'19)*. Ed. by M. Balduccini, Y. Lierler, and S. Woltran. Vol. 11481. Lecture Notes in Artificial Intelligence. Springer-Verlag, pp. 256–269. DOI: 10.1007/978-3-030-20528-7_19.

Kaminski, R., J. Romero, T. Schaub, and P. Wanko (2023). "How to Build Your Own ASP-based System?!" In: *Theory and Practice of Logic Programming* 23.1, pp. 299–361. DOI: 10.1017/S1471068421000508.

Kamp, J. (1968). "Tense Logic and the Theory of Linear Order". PhD thesis. University of California at Los Angeles.

Lifschitz, V. (2008). "What Is Answer Set Programming?" In: *Proceedings of the Twenty-third National Conference on Artificial Intelligence (AAAI'08)*. Ed. by D. Fox and C. Gomes. AAAI Press, pp. 1594–1597.

Rosa, E. Di, E. Giunchiglia, and M. Maratea (2010). "Solving satisfiability problems with preferences". In: *Constraints* 15.4, pp. 485–515.

Simons, P., I. Niemelä, and T. Soininen (2002). "Extending and implementing the stable model semantics". In: *Artificial Intelligence* 138.1-2, pp. 181–234.