On Strong Equivalence Notions in Logic Programming and Abstract Argumentation

Giovanni Buraglio[0009−0004−9592−4739], Wolfgang Dvořák[0000−0002−2269−8193], and Stefan Woltran[0000−0003−1594−8972]

TU Wien, Institute of Logic and Computation, Wien, Austria
{giovanni.buraglio,wolfgang.dvorak,stefan.woltran}@tuwien.ac.at

Abstract. We summarize our recent work on strong equivalence notions in Logic Programming and Abstract Argumentation [2]. While (classes of) logic programs and argumentation frameworks are known to be semantically equivalent in static settings, this alignment breaks in dynamic contexts due to differing notions of update. As a result, strong equivalence does not always carry over from one formalism to the other. Motivated by this discrepancy we investigate a new notion of strong equivalence for certain classes of logic programs which is preserved under translation to and from argumentation frameworks, thus restoring compatibility across these formalisms.

1 Strong Equivalence Based on Rule Refinement

In the field of Knowledge Representation and Reasoning, the concept of equivalence between knowledge bases has been the subject of extensive studies. A primary motivation behind this research is the potential to exploit the equivalence between two knowledge bases to achieve a compact representation of the same information. While this advantageous behavior can be taken for granted in monotonic formalisms, it is usually not the case in non-monotonic ones. Thus, the notion of strong equivalence has been introduced to capture equivalence in a dynamic environment, under any possible update [3, 7, 10, 11, 15, 16].

In this paper, we consider two families of non-monotonic formalisms, namely logic programming and abstract argumentation. Logic programming is a declarative programming paradigm where a reasoning problem is specified by means of a so-called logic program (LP), i.e. a set of inference rules made of atoms, possibly preceded by a negation-as-failure operator. Abstract argumentation [1, 13] is a sub-field of symbolic Artificial Intelligence that offers formal approaches to represent and reason over situations where conflicting information is present. An argumentative scenario is specified by means of an abstract argumentation framework, which is a directed graph where nodes represent arguments and edges an attack relation [5]. Previous work has shown a one-to-one mapping between various argumentation formalisms and (classes of) normal LPs under the stable model semantics [4, 5]. In particular, any problem can be either specified via a program P or an argumentation framework F in such a way that the answer sets of P are identical to the stable extensions of F.

However, this semantic correspondence does not carry over to dynamic contexts, where strong equivalence is required. That is, there are logic programs that are strongly equivalent but induce argumentation frameworks that do not share this property, due to the incongruous notions of update (or expansion) in the two realms. We show this in the following example:

Example 1. Consider the AF F (left) and the equivalent logic program P:

F

$$\begin{tikzcd}
x \arrow[r, bend left] & y \arrow[l, bend left] & a \arrow[l] \arrow[ll, bend right]
\end{tikzcd}$$

$$P = \{x \leftarrow \texttt{not}\, y, \texttt{not}\, a.,\;\; y \leftarrow \texttt{not}\, x, \texttt{not}\, a.,\;\; a \leftarrow .\}$$

Subsequently, new information comes into play, captured via:

F'

$$\begin{tikzcd}
a & d \arrow[l]
\end{tikzcd}$$

$$P' = \{a \leftarrow \texttt{not}\, d.,\;\; d \leftarrow .\}$$

In accordance with equivalence results between LPs and AFs, the two ways of modeling are consistent with each other when taken individually. However, this does not happen when they are combined. Incorporating the second pair of knowledge bases (F' and P') into the first one (resp. F and P) yields different result: in the case of F and F', their union returns the expected result in the form of two possible extensions $\{d, x\}$ and $\{d, y\}$. On the other hand, the union of the two logic programs P and P' yields an unexpected prediction: the fact '$a \leftarrow$' is not overwritten by '$d \leftarrow$', leaving $\{a, d\}$ as the only possible solution.

The discrepancy illustrated above reveals that updating an existing logic program by simply adding rules may produce unexpected outcomes. Facts (e.g. '$a \leftarrow$' in P) cannot be overwritten by incoming information, while their corresponding arguments (e.g. $a \in F$) can. This behavior of logic programs is fundamentally in contrast with the way in which non-monotonicity is encoded in abstract argumentation: an argument can always be attacked by new ones.

In our work, we carefully analyze the relationship between logic programming and abstract argumentation, with a particular interest in dynamic contexts. Motivated by the mismatch above, we introduce a novel notion of update, called Rule Refinement, for the class of LPs that correspond to AFs. It resolves the issue by mimicking precisely the existing notion of updates for AFs. In particular, updating a program P with another program P' consists of two possible operations: in the case where two rules $r \in P$ and $r' \in P'$ share the same head-atom, the body of r is merged with that of r'; otherwise, the update is a simple set-union. Indeed, from the LP perspective, adding attackers on the corresponding AF means adding negated atoms to the rule whose head identifies the attacked argument. Notice that this idea of updating a program by extending its rule-bodies has already been considered in various settings, e.g. involving update sequences [6], abduction [9, 8] and inconsistency repairs [14].

In future research, we aim to extend our characterization of strong equivalence under Rule Refinement for the entire class of normal logic programs, and connect this to more expressive type of argumentation frameworks. Further, we aim to investigate the possibility to characterize Rule Refinement strong equivalence via an equivalence notion in an intermediate logic, analogously to the approach employed for the standard notion and the logic of Here-and-There [10, 12].

References

1. Baroni, P., Gabbay, D., Giacomin, M., van der Torre, L. (eds.): Handbook of Formal Argumentation. College Publications (2018)
2. Buraglio, G., Dvořák, W., Woltran, S.: On strong equivalence notions in logic programming and abstract argumentation. NMR 2025
3. Cabalar, P.: A three-valued characterization for strong equivalence of logic programs. In: Dechter, R., Kearns, M.J., Sutton, R.S. (eds.) Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence. pp. 106–111. AAAI Press / The MIT Press (2002), http://www.aaai.org/Library/AAAI/2002/aaai02-017.php
4. Caminada, M., Sá, S., Alcântara, J., Dvořák, W.: On the equivalence between logic programming semantics and argumentation semantics. Int. J. Approx. Reasoning **58**, 87–111 (2015). https://doi.org/10.1016/j.ijar.2014.12.004
5. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**(2), 321–358 (1995)
6. Eiter, T., Fink, M., Sabbatini, G., Tompits, H.: On properties of update sequences based on causal rejection. Theory Pract. Log. Program. **2**(6), 711–767 (2002). https://doi.org/10.1017/S1471068401001247
7. Eiter, T., Fink, M., Woltran, S.: Semantical characterizations and complexity of equivalences in answer set programming. ACM Trans. Comput. Log. **8**(3) (2007), http://doi.acm.org/10.1145/1243996.1244000
8. Inoue, K.: A simple characterization of extended abduction. In: Lloyd, J.W., Dahl, V., Furbach, U., Kerber, M., Lau, K., Palamidessi, C., Pereira, L.M., Sagiv, Y., Stuckey, P.J. (eds.) Proceedings of the First International Conference on Computational Logic - CL 2000. Lecture Notes in Computer Science, vol. 1861, pp. 718–732. Springer (2000). https://doi.org/10.1007/3-540-44957-4_48
9. Kowalski, R.A., Toni, F.: Abstract argumentation. Artif. Intell. Law **4**(3-4), 275–296 (1996). https://doi.org/10.1007/BF00118494
10. Lifschitz, V., Pearce, D., Valverde, A.: Strongly equivalent logic programs. ACM Trans. Comput. Log. **2**(4), 526–541 (2001). https://doi.org/10.1145/383779.383783
11. Oikarinen, E., Woltran, S.: Characterizing strong equivalence for argumentation frameworks. Artif. Intell. **175**(14-15), 1985–2009 (2011)
12. Pearce, D.: A new logical characterisation of stable models and answer sets. In: Dix, J., Pereira, L.M., Przymusinski, T.C. (eds.) Non-Monotonic Extensions of Logic Programming, NMELP '96, Bad Honnef, Germany, September 5-6, 1996, Selected Papers. Lecture Notes in Computer Science, vol. 1216, pp. 57–70. Springer (1996). https://doi.org/10.1007/BFB0023801
13. Rahwan, I., Simari, G.R.: Argumentation in Artificial Intelligence. Springer (2009)
14. Thevapalan, A., Kern-Isberner, G.: On establishing robust consistency in answer set programs. Theory Pract. Log. Program. **23**(5), 1094–1127 (2023). https://doi.org/10.1017/S1471068422000357
15. Truszczynski, M.: Strong and uniform equivalence of nonmonotonic theories - an algebraic approach. Ann. Math. Artif. Intell. **48**(3-4), 245–265 (2006). https://doi.org/10.1007/S10472-007-9049-2
16. Turner, H.: Strong equivalence for logic programs and default theories (made easy). In: Eiter, T., Faber, W., Truszczynski, M. (eds.) Logic Programming and Nonmonotonic Reasoning, 6th International Conference, LPNMR 2001. Lecture Notes in Computer Science, vol. 2173, pp. 81–92. Springer (2001). https://doi.org/10.1007/3-540-45402-0_6