SAT Modulo Well-Founded Semantics

Extended Abstract of the SAT'26 Publication

Thomas Eiter, Tobias Nießen, and Davide Soldà

TU Wien, Austria
{thomas.eiter,tobias.niessen,davide.solda}@tuwien.ac.at

Abstract. The well-founded semantics (WFS) yields a unique three-valued model for logic programs that serves as an efficient core for skeptical reasoning, but lacks built-in mechanisms for choice and case-based reasoning, limiting its expressiveness for problems such as decision making and planning. Propositional SAT solvers excel at combinatorial problems but, unlike WFS, do not naturally support reasoning under incomplete information or encoding transitive closure properties. We present an integration of a choice operator into WFS that preserves the suitability of the semantics for scalable, partial-information reasoning, as well as an efficient solving procedure.

Keywords: Well-Founded Semantics · SAT · Least Fixpoint Computation

The well-founded semantics (WFS) [12] provides a three-valued semantics for logic programs with negation that for ground (propositional) programs is computable in polynomial time, but its skeptical nature offers only limited support for decision making, as WFS does not explicitly branch reasoning into alternative models. No broadly accepted solution has emerged for overcoming this limitation through disjunctive rule heads [1,5,7,18,21]. We propose an integration of a *choice* construct [17] into WFS, which allows capturing nondeterministic branching in a semantically principled manner. Notably, such controllable nondeterminism is conceptually different from undefinedness, which can be interpreted as epistemic uncertainty.

We introduce *extended logic programs with choice* (ELP$^{\mathrm{C}}$) by augmenting the syntax of extended logic programs (ELP) with an explicit choice operator. We define the semantics of such programs by extending WFSX [2], i.e., well-founded semantics with explicit negation. We show how the combination of the choice construct and explicit negation enables reasoning about actions in a partial information setting, where distinguishing choice and undefinedness reflects the classical separation of optional actions and incomplete states [9,20]. We develop support for *integrity constraints* [3, Section 7.1] for ELP$^{\mathrm{C}}$. Interestingly, this combination of nondeterministic choices and integrity constraints allows deriving facts from undefinedness.

Example 1 (Meta reasoning about undefinedness). Consider the two programs

$$\pi_1 = \{\, \mathtt{a} \leftarrow \mathrm{not}\,\mathtt{a}, \mathtt{b}., \mathtt{b}.,\ 0\{\,\mathtt{a\_und}\,\}1.\,\}\quad \pi_2 = \{\, \mathtt{a} \leftarrow \mathrm{not}\,\mathtt{a}, \mathtt{b}.,\ \neg\mathtt{b}., 0\{\,\mathtt{a\_und}\,\}1.\,\}$$

and the following integrity constraints:

$$c_1:\ \ \mathrm{not}\,\mathtt{a\_und} \Lleftarrow \mathtt{a}. \quad c_2:\ \ \mathrm{not}\,\mathtt{a\_und} \Lleftarrow \mathrm{not}\,\mathtt{a}. \quad c_3:\ \mathtt{a} \lor \mathrm{not}\,\mathtt{a} \Lleftarrow \mathrm{not}\,\mathtt{a\_und}.$$

The latter represent constraints ($\Leftarrow$ implications) that are evaluated against the full model rather than being part of the model construction. During the evaluation of π_1, there are two possible valid choices for $0\{\,\mathtt{a\_und}\,\}1.$, which result in the respective well-founded models $W_1 = \{\,\text{not}\,\neg\mathtt{a},\ \mathtt{b},\ \text{not}\,\neg\mathtt{b},\ \mathtt{a\_und},\ \text{not}\,\neg\mathtt{a\_und}\,\}$ and $W_2 = \{\,\text{not}\,\neg\mathtt{a}, \mathtt{b}, \text{not}\,\neg\mathtt{b}, \text{not}\,\mathtt{a\_und}, \neg\mathtt{a\_und}\,\}$. Because a is undefined in either well-founded model of π_1, the integrity constraints c_1 and c_2 are satisfied by both. However, W_2 violates c_3 since a is undefined, i.e., neither true nor false, while a_und is false in W_2, so the only well-founded model of π_1 satisfying the integrity constraints is W_1. Similary, the two well-founded models of π_2 are $W_3 = \{\,\text{not}\,\mathtt{a}, \text{not}\,\neg\mathtt{a}, \text{not}\,\mathtt{b}, \neg\mathtt{b}, \mathtt{a\_und}, \text{not}\,\neg\mathtt{a\_und}\,\}$ and $W_4 = \{\,\text{not}\,\mathtt{a}, \text{not}\,\neg\mathtt{a}, \text{not}\,\mathtt{b}, \neg\mathtt{b}, \text{not}\,\mathtt{a\_und}, \neg\mathtt{a\_und}\,\}$. W_3 contains a_und but also not a and therefore violates integrity constraint c_2, so the only well-founded model of π_2 satisfying all integrity constraints is W_4.

We further present a program evaluation procedure for ELP^{C} that integrates choices into a standard WFSX evaluation algorithm via calls to a SAT solver. We develop two optimization techniques for this procedure: first, *clause learning* is a mechanism inspired by the SLX procedure [3], which is able to reconstruct and generalize the choices that led to contradictions and violations of integrity constraints. Second, *program decomposition* builds on program splitting [15] and stratified subprograms, structuring the input program into an alternation of stratified and non-stratified components, which can be exploited operationally.

We obtain in this way a *SAT-modulo-fixpoint* mechanism in which recursive definitions coexist with Boolean connectives in a uniform and semantically grounded manner. Viewed from this angle, our work contributes to the long-standing challenge of integrating inductive definitions and transitive closure into SMT solving, a problem that motivated several extensions such as μZ [14] and approaches based on Constrained Horn Clauses [4,13], which however are two-valued and have different fixpoint semantics. In [16], an implementation of SAT(ID) was reported, where SAT was extended with inductive definitions, in the style of FO(ID) [8].

We implement the proposed semantics using a standard SAT solver [6] for making choices with an efficient encoding of cardinality constraints [19]. Further, the development of a clause learning procedure for propagating information about contradictory fixpoints back to the SAT solver and of program decomposition techniques makes our semantics capable of handling complex logic programs. We evaluate our framework using a combinatorial variant of the graph winning problem [11], in which the solver must mark graph nodes as *winning* or *losing* based on inductive constraints, while being allowing to edit the graph through edge addition or removal. The "undefined" truth value is assigned to nodes for which it is impossible to determine whether they are winning or losing. In this experimental setting, we find that our SAT-based approach overall significantly outperforms an ASP encoding of the same semantics, solved by a state-of-the-art ASP engine [10].

Disclosure of Interests. This project has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 101034440). Eiter and Soldà are partially supported by the MOSAIC project (EU H2020-MSCA-RISE-2020 Project 101007627) and by WWTF project ICT22-023.

References

1. Alcântara, J.F.L., Damásio, C.V., Pereira, L.M.: A well-founded semantics with disjunction. In: Gabbrielli, M., Gupta, G. (eds.) Logic Programming, 21st International Conference, ICLP 2005, Sitges, Spain, October 2-5, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3668, pp. 341–355. Springer, Springer (2005). https://doi.org/10.1007/11562931_26
2. Alferes, J.J.: Semantics of logic programs with explicit negation. Ph.D. thesis, Universidade Nova de Lisboa, Lisbon, Portugal (1993), http://hdl.handle.net/10362/1056
3. Alferes, J.J., Damásio, C.V., Pereira, L.M.: A logic programming system for nonmonotonic reasoning. Journal of Automated Reasoning **14**, 93–147 (1995). https://doi.org/10.1007/BF00883931
4. Amrollahi, D., Hojjat, H., Rümmer, P.: An encoding for CLP problems in SMT-LIB. In: Kutsia, T., Ventura, D., Monniaux, D., Morales, J.F. (eds.) Proceedings 18th International Workshop on Logical and Semantic Frameworks, with Applications and 10th Workshop on Horn Clauses for Verification and Synthesis, LSFA/HCVS 2023. vol. 402, pp. 118–130. Open Publishing Association (Apr 2023). https://doi.org/10.4204/eptcs.402.12
5. Baral, C., Lobo, J., Minker, J.: Generalized disjunctive well-founded semantics for logic programs. Annals of Mathematics and Artificial Intelligence **5**(2), 89–131 (1992). https://doi.org/10.1007/BF01543473
6. Biere, A., Faller, T., Fazekas, K., Fleury, M., Froleyks, N., Pollitt, F.: CaDiCaL 2.0. In: Gurfinkel, A., Ganesh, V. (eds.) Computer Aided Verification - 36th International Conference, CAV 2024, Montreal, QC, Canada, July 24-27, 2024, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14681, pp. 133–152. Springer, Springer (2024). https://doi.org/10.1007/978-3-031-65627-9_7
7. Cabalar, P., Odintsov, S.P., Pearce, D.: Logical foundations of well-founded semantics. In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006. vol. 6, pp. 25–35. AAAI Press (2006), https://aaai.org/papers/KR06-006-logical-foundations-of-well-founded-semantics/
8. Denecker, M., Ternovska, E.: A logic of nonmonotone inductive definitions. ACM transactions on computational logic (TOCL) **9**(2), 1–52 (2008). https://doi.org/10.1145/1342991.1342998
9. Eiter, T., Faber, W., Leone, N., Pfeifer, G., Polleres, A.: A logic programming approach to knowledge-state planning, II: the dlv$^{\mathrm{k}}$ system. Artificial Intelligence **144**(1-2), 157–211 (2003). https://doi.org/10.1016/S0004-3702(02)00367-3
10. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Multi-shot ASP solving with clingo. Theory and Practice of Logic Programming **19**(1), 27–82 (2019). https://doi.org/10.1017/S1471068418000054
11. Gelder, A.V., Ross, K.A., Schlipf, J.S.: Unfounded sets and well-founded semantics for general logic programs. In: Edmondson-Yurkanan, C., Yannakakis, M. (eds.) Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, March 21-23, 1988, Austin, Texas, USA. pp. 221–230. ACM (1988). https://doi.org/10.1145/308386.308444
12. Gelder, A.V., Ross, K.A., Schlipf, J.S.: The well-founded semantics for general logic programs. Journal of the ACM (JACM) **38**(3), 620–650 (1991). https://doi.org/10.1145/116825.116838

13. Gurfinkel, A.: Program verification with constrained horn clauses (invited paper). In: Shoham, S., Vizel, Y. (eds.) Computer Aided Verification - 34th International Conference, CAV 2022, Haifa, Israel, August 7-10, 2022, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13371, pp. 19–29. Springer (2022). https://doi.org/10.1007/978-3-031-13185-1_2
14. Hoder, K., Bjørner, N.S., de Moura, L.M.: μZ- an efficient engine for fixed points with constraints. In: Gopalakrishnan, G., Qadeer, S. (eds.) Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6806, pp. 457–462. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22110-1_36
15. Lifschitz, V., Turner, H.: Splitting a logic program. In: Hentenryck, P.V. (ed.) Proceedings of the Eleventh International Conference on Logic Programming. vol. 94, pp. 23–37. MIT Press (1994)
16. Mariën, M., Wittocx, J., Denecker, M.: Integrating inductive definitions in sat. In: International Conference on Logic for Programming Artificial Intelligence and Reasoning. pp. 378–392. Springer (2007). https://doi.org/10.1007/978-3-540-75560-9_28
17. Niemelä, I., Simons, P., Soininen, T.: Stable model semantics of weight constraint rules. In: Gelfond, M., Leone, N., Pfeifer, G. (eds.) Logic Programming and Nonmonotonic Reasoning, 5th International Conference, LPNMR'99, El Paso, Texas, USA, December 2-4, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1730, pp. 317–331. Springer, Springer (1999). https://doi.org/10.1007/3-540-46767-X_23
18. Ross, K.A.: The well founded semantics for disjunctive logic programs. In: Kim, W., Nicolas, J., Nishio, S. (eds.) Deductive and Object-Oriented Databases, Proceedings of the First International Conference on Deductive and Object-Oriented Databases (DOOD'89), Kyoto Research Park, Kyoto, Japan, 4-6 December, 1989. pp. 385–402. North-Holland/Elsevier Science Publishers (1989)
19. Sinz, C.: Towards an optimal CNF encoding of boolean cardinality constraints. In: van Beek, P. (ed.) Principles and Practice of Constraint Programming - CP 2005, 11th International Conference, CP 2005, Sitges, Spain, October 1-5, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3709, pp. 827–831. Springer, Berlin, Heidelberg (2005). https://doi.org/10.1007/11564751_73
20. Soldà, D., Eiter, T.: deon-B: A language for well-founded deontic planning. In: Casini, G., Dundua, B., Kutsia, T. (eds.) Logics in Artificial Intelligence - 19th European Conference, JELIA 2025, Kutaisi, Georgia, September 1-4, 2025, Proceedings, Part I. Lecture Notes in Computer Science, vol. 16093, pp. 187–204. Springer, Springer (2025). https://doi.org/10.1007/978-3-032-04587-4_12
21. Wang, K., Zhou, L.: Comparisons and computation of well-founded semantics for disjunctive logic programs. ACM Transactions on Computational Logic (TOCL) **6**(2), 295–327 (2005). https://doi.org/10.1145/1055686.1055690