Witnesses for Answer Sets of Logic Programs (Extended Abstract)*

Yisong Wang[1 [0000−0003−2126−7006]], Thomas Eiter[2 [0000−0001−6003−6345]], Yuanlin Zhang[3], and Fangzhen Lin[4 [0000−0002−3141−8675]]

[1] Guizhou University, Guiyang, Guizhou, China yswang@gzu.edu.cn
[2] Technische Universität Wien, Vienna, Austria eiter@kr.tuwien.ac.at
[3] Texas Tech University, Lubbock, Texas, USA y.zhang@ttu.edu
[4] Hong Kong University of Science and Technology, HK flin@cs.ust.hk

Abstract. In this paper, we propose a notion of reduct for logic programs and show that each atom in an answer set has a resolution proof from the reduct with respect to the answer set. Such a resolution proof provides an explanation of "why a set of atoms is an answer set". We then further consider (minimal) sets of rules that will be sufficient to provide resolution proofs for sets of atoms. Such sets of rules will be called witnesses and are in the focus of this paper. We study complexity issues of computing various witnesses and provide algorithms for computing them. In particular, we show that the problem is intractable in general. Experiments on many well-known ASP and SAT benchmarks show that computing a minimal witness for an atom of an answer set is often feasible. In most cases, a resolution proof for an atom is easy to construct from its witness.

Keywords: Witness · Explanation · Minimal Models · Disjunctive Logic Programs.

1 Motivation

Explainability and explanation play an important role in increasing transparency in automated-decision systems [8, 4]. Answer set programming (ASP) has been deployed to decision making [2] and provides due to its rule-based language and declarative semantics a promising base for achieving this capability. However, to explain atoms in answer sets of an ASP program in a well-justified manner [3, 5, 1] is challenging and has, to our best knowledge, not been widely addressed.

To address this challenge, we consider logical (resolution) proofs as solid explanations and propose a new reduct $\mathrm{MR}(\Pi, M)$ of a logic program Π *w.r.t.* an answer set M that keeps all rules whose bodies are satisfied and removes all literals on atoms false in M. For example, $\Pi = \{r_1 : a \vee b \leftarrow \mathit{not}\ c;\ r_2 : a \leftarrow b;\ r_3 : b \leftarrow a\}$ has the unique answer set $M = \{a, b\}$. Then $\mathrm{MR}(\Pi, M) = \{a \vee b;\ a \leftarrow b;\ b \leftarrow a\}$, from which both a and b can be logically derived viewing '$\leftarrow$' as material implication.

* This is an extended abstract of an article in ACM Trans. Computational Logic [9].

2 Basic Notions of Witness and Results

Assuming a propositional language $\mathcal{L}$ over a signature $\mathcal{A}$, a logic program Π is a finite set of rules r of the form

$$p_1 \vee \cdots \vee p_k \leftarrow p_{k+1}, \cdots, p_m, not\ p_{m+1}, \cdots, not\ p_n \tag{1}$$

where each p_i is from $\mathcal{A}$. Recall that a set $M \subseteq \mathcal{A}$ is an *answer set* of Π if M is a ($\subseteq$)-minimal model of the GL-reduct $\Pi^M = \{p_1 \vee \cdots \vee p_k \leftarrow p_{k+1}, \cdots, p_m \mid r$ of form (1) $\in \Pi, \{p_{m+1}, \ldots, p_n\} \cap M = \emptyset\}$ [6, 7].

Definition 1. *The reduct of a logic program Π w.r.t. $M \subseteq \mathcal{A}$, denoted* $\mathrm{MR}(\Pi, M)$*, contains for each r of form (1) in Π such that $\{p_{m+1}, \ldots, p_n\} \cap M = \emptyset$ the rule*

$$q_1 \vee \cdots \vee q_s \leftarrow p_{k+1}, \ldots, p_m \quad (\textit{i.e., clause } q_1 \vee \cdots \vee q_s \vee \neg p_{k+1} \vee \cdots \vee \neg p_m)$$

with $\{q_1, \ldots, q_s\} = \{p_1, \ldots, p_k\} \cap M$, $\{p_{k+1}, \ldots, p_m\} \subseteq M$.

Theorem 1. *Let Π be a logic program and $M \subseteq \mathcal{A}$. Then M is an answer set of Π iff M is the least (i.e., unique minimal) model of* $\mathrm{MR}(\Pi, M)$*.*

Our notions of witnesses are as follows.

Definition 2. *Let $M \subseteq \mathcal{A}$, and $B, S \subseteq M$ be disjoint. A logic program Π is a (minimal)* witness *of B under S w.r.t. M, if $\mathrm{MR}(\Pi, M) \cup S$ (minimally) logically entails B, i.e., 'minimal' means that for none of its proper subset this holds.*

Definition 3. *Let Π be a logic program and $M \neq \emptyset$ be an answer set of Π. Furthermore, let $B, S \subseteq M$ be disjoint. Then, an $\alpha^\star$-*witness *of B under Π and S w.r.t. M is a DAG $G = (\{(S_i, \Pi_i) \mid 1 \leq i \leq n\}, E)$ where $\{S_i \mid 1 \leq i \leq n\}$ is a partitioning of B and, for every $i, 1 \leq i \leq n$,*

(i) $\Pi_i \subseteq \Pi$ is a witness of S_i under $S \cup X_i$ w.r.t. M, and
(ii) Π_i is not a witness of S_j under $S \cup X_i$ w.r.t. M, for every $1 \leq j \neq i \leq n$,

$$X_k = \bigcup\{S' \mid (S', \Pi') \in D_G((S_k, \Pi_k))\}, \quad 1 \leq k \leq n,$$

where $D_G(v)$ is the set of nodes of G from which v is reachable.

If G induces a total order $(S_1, \Pi_1) < (S_2, \Pi_2) < \cdots < (S_n, \Pi_n)$, we call it an α-witness *of B under Π and S w.r.t. M and write $G = [(S_1, \Pi_1), \ldots, (S_n, \Pi_n)]$. We call G* minimal*, if every Π_i is minimal and G is* compact*, if in addition to minimality $\Pi_i \cap \Pi_j = \emptyset$ for all $1 \leq i < j \leq n$. If $B = M$ and $S = \emptyset$, we call G an (minimal, compact) $\alpha^\star$-witness resp. α-witness of M w.r.t. Π.*

If each S_i is a singleton, the (minimal, compact) $\alpha^\star$- resp. α-witness is a (minimal, compact) $\beta^\star$- resp. *β-witness.* Intuitively, a (minimal) α witness splits B into subparts S_i that can be modularly derived with (non-redundant) rules relative to asserted atoms S and the previously derived atoms; β means each part S_i is an atom and * that modules are merely partially ordered, while compactness forbids reusing rules across modules. E.g., the answer set M of Π from above has two compact β-witnesses $G_1 = [(a, \{r_1, r_2\}), (b, \{r_3\})]$, $G_2 = [(b, \{r_1, r_3\}), (a, \{r_2\})]$.

Minimal and compact witnesses of all sorts always exist, except compact $\beta^\star$- and β-witnesses. It is intractable to tell whether such witnesses exist.

Theorem 2. *Deciding whether an answer set M of a logic program Π has a compact β-witness (resp. compact β-witness) is Σ_2^p-complete.*

Extensive experiments of computing minimal β-witnesses on well-known benchmarks of ASP and SAT solving show that many of them are in fact compact and that in most cases, the local witness Π_i consists of only a single rule.

References

1. Arias, J., Carro, M., Chen, Z., Gupta, G.: Justifications for goal-directed constraint answer set programming. In: Ricca, F., Russo, A., Greco, S., Leone, N., Artikis, A., Friedrich, G., Fodor, P., Kimmig, A., Lisi, F.A., Maratea, M., Mileo, A., Riguzzi, F. (eds.) Proceedings 36th International Conference on Logic Programming (Technical Communications), ICLP Technical Communications 2020, (Technical Communications) UNICAL, Rende (CS), Italy, 18-24th September 2020. EPTCS, vol. 325, pp. 59–72 (2020). https://doi.org/10.4204/EPTCS.325.12, https://doi.org/10.4204/EPTCS.325.12
2. Brewka, G., Eiter, T., Truszczynski, M.: Answer set programming at a glance. Commun. ACM **54**(12), 92–103 (2011)
3. Cabalar, P., Fandinno, J.: Justifications for programs with disjunctive and causal-choice rules. Theory and Practice of Logic Programming **16**(5-6), 587–603 (2016). https://doi.org/10.1017/S1471068416000454, http://dx.doi.org/10.1017/S1471068416000454
4. Confalonieri, R., Weyde, T., Besold, T.R., Moscoso del Prado Martín, F.: Using ontologies to enhance human understandability of global post-hoc explanations of black-box models. Artificial Intelligence **296**, 103471 (2021)
5. Fandinno, J., Schulz, C.: Answering the "why" in answer set programming – a survey of explanation approaches. Theory and Practice of Logic Programming **19**(2), 114–203 (2019). https://doi.org/10.1017/S1471068418000534
6. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R.A., Bowen, K.A. (eds.) Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, USA, August 15-19, 1988 (2 Volumes). pp. 1070–1080. MIT Press, Seattle, Washington, USA (1988)
7. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing **9**, 365–385 (1991)
8. Srinivasan, R., Chander, A.: Explanation perspectives from the cognitive sciences - A survey. In: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020. pp. 4812–4818. ijcai.org, Yokohama, Japan (2020). https://doi.org/10.24963/ijcai.2020/670, https://doi.org/10.24963/ijcai.2020/670, scheduled for July 2020, Yokohama, Japan, postponed due to the Corona pandemic.
9. Wang, Y., Eiter, T., Zhang, Y., Lin, F.: Witnesses for answer sets of logic programs. ACM Transactions on Computational Logic (TOCL) Just accepted, 45+21 (2022). https://doi.org/10.1145/3568955