

Towards Observing the Effect of Abstraction on Understandability of Explanations in Answer Set Programming

Zeynep G. Saribatur¹[0000–0001–8690–5043], Johannes Langer²[0009–0009–0603–3034], Anna M. Thaler²[0009–0001–9027–5811], and Ute Schmid²[0000–0002–1301–0326]

¹ Institute of Logic and Computation, TU Wien
`zeynep.saribatur@tuwien.ac.at`

² University of Bamberg
`{johannes.langer,anna.thaler,ute.schmid}@uni-bamberg.de`

Abstract. In order for AI systems to provide explanations of their decision-making that are concise and understandable, they need to have the ability of getting rid of irrelevant details and presenting a higher-level view. Answer Set Programming (ASP) is one of the core formalisms of Symbolic AI, based on logic programming with stable model semantics, widely used in various applications. Explaining the solutions (i.e., answer sets) of an answer set program continues to be a widely studied topic, with various systems available. This technical communication reports on a preliminary study for observing the effect of abstraction on the understandability of ASP explanations, by considering the recent abstraction notions which preserve the dependencies as much as possible while abstracting over answer set programs. We describe our experiment design which will be our base for further extensions of the study. Our preliminary results show the challenge of capturing the effect of abstraction on understandability, requiring further investigations in this direction.³

Keywords: Abstraction · Explanations · Answer set programming

1 Introduction

Abstraction through simplifying and generalizing are abilities that humans unwittingly use when understanding and reasoning about the world [3,13]. Especially as we aim for AI systems that are transparent and understandable to humans, such systems need to acquire abstraction abilities that allow them

³ This version of the contribution has been accepted for publication, after peer review (when applicable) but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/978-3-032-02813-6_18. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use <https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms>

to present explanations of their complex decision-making and representations overviewing their complex structures via showing the key elements, so-called “model-of-self” [14], making it easier for humans to understand. In the field of Explainable AI (XAI), some recent works involving abstraction include forgetting [22] or projecting away details in explanations [6], simplifying solutions [23], abstracting rule-based representations by predicate invention [20] or domain clustering [8], explaining neural networks through causal abstractions [11] or decision trees [7], showing the need and the potential of abstraction.

Answer set programming (ASP) [4], which is based on logic programming with stable model semantics, is one of the core formalisms in Symbolic AI and thanks to efficient solvers widely used in many areas of Computer Science and AI, from combinatorial search problems over system modeling to knowledge-intensive applications [9]. Obtaining explanations for ASP solutions has also been a topic that is widely studied with available systems. The challenge of achieving concise explanations to aid in human understanding still remains [10]. Obtaining abstractions over the explanations would be a way to achieve this.

Abstraction is first introduced for ASP as an “over-approximation” of programs that ensures all answer sets are preserved while reducing the vocabulary [16,17]. Later, more restrictive properties have been investigated ensuring that all dependencies are preserved, by considering strong/uniform equivalence like relations [19,18]. These abstractions of programs were shown to get rid of or abstract over details that are not the key points for reasoning.

We report on an ongoing work that investigates, through cognitive experiments, whether the concise explanations obtained over these abstract answer set programs aid in human understandability. In this paper, we illustrate our hypothesis and describe the experiment design, with preliminary results in fact showing no effect. We view these results as the first step towards understanding the problem, discuss potential shortcomings and ideas on further extensions of the experiment with added complexity to be able capture the effect of abstraction on understandability.

2 Background

ASP An answer set program P over a set \mathcal{U} of propositional atoms is a set of rules r of the form $\alpha_0 \leftarrow \alpha_1, \dots, \alpha_m, \text{not } \alpha_{m+1}, \dots, \text{not } \alpha_n$, $0 \leq m \leq n$, where each $\alpha_i \in \mathcal{U}$ is a propositional literal and *not* is default negation. We also write r as $H(r) \leftarrow B^+(r), \text{not } B^-(r)$, where $H(r) = \{A_1, \dots, A_l\}$ is the *head*, $B^+(r) = \{A_{l+1}, \dots, A_m\}$ is the *positive body* and $B^-(r) = \{A_{m+1}, \dots, A_n\}$ is the *negative body* of r . A rule r is a *constraint* if α_0 is falsity (\perp , then omitted), a *fact* if $n = 0$ and *positive* if $B^-(r) = \emptyset$. Semantically P induces a set $AS(P)$ of stable models (or answer sets), which are sets I of atoms of P that are minimal models of the *reduct* [12] given by $P^I = \{H(r) \leftarrow B^+(r) \mid r \in P, B^-(r) \cap I = \emptyset\}$.

Abstraction Recent works study the theory of removal [19] and abstraction [18] of irrelevant details in answer set programs. The aim is to define a simplification resp. an abstraction of an answer set program that preserves the dependencies

while removing resp. abstracting over some details. As it was shown that simplification by removal [19] is a special case of abstraction defined in [18], we only provide details of the latter, where a mapping is defined to cluster atoms in \mathcal{U} .

Definition 1 ([18]) *Given sets of atoms $\mathcal{U}, \mathcal{U}'$ with $|\mathcal{U}| \geq |\mathcal{U}'|$, a program P over \mathcal{U} and a mapping $m : \mathcal{U} \mapsto \mathcal{U}'$, Q over \mathcal{U}' is a uniform m -abstraction of P if, for any set F of facts over \mathcal{U} , we have*

$$m(AS(P \cup F)) = AS(Q \cup m(F)). \quad (1)$$

Informally, the aim is to map atoms from the language \mathcal{U} of program P to atoms in a smaller language \mathcal{U}' in such a way that the answer sets of P and the resulting abstraction Q correspond, independently of the facts, i.e., the instance data, added to the program. As not every program might have a uniform m -abstraction, the necessary and sufficient conditions for abstractability, and model-based representations of abstractions were provided. Furthermore it was shown that, whenever possible, the abstract program can be achieved simply by syntactic clustering of the atoms (or syntactic removal in [19]).

3 Abstractions over Explanations

In this section, we illustrate the potential use of abstraction notion from Definition 1 for obtaining abstract explanations containing the key details of reasoning.

Example 1 *Let us consider the following program P .*

$$\begin{aligned} reachPotsdam &\leftarrow takePlane. \\ reachPotsdam &\leftarrow takeTrain. \\ attendKI &\leftarrow talkReady, register, reachPotsdam. \end{aligned}$$

If we reach Potsdam by either taking the plane or the train, register to the conference and have our talk ready, then we can attend the conference.

Now consider having different scenarios in which we prepare our talk to present at KI, we either take the plane, the train, or start walking towards Potsdam, and we register or not. Since the talk is ready in all scenarios, that detail is not decisive in attending the conference, thus can be removed. Since both, taking the plane and taking the train (but not walking), allow us to reach Potsdam, the details of which transportation is taken can be abstracted over. Removing and abstracting over these details yield the program Q below.

$$\begin{aligned} reachPotsdam &\leftarrow takePlaneOrTrain. \\ attendKI &\leftarrow register, reachPotsdam. \end{aligned}$$

To theoretically capture the cases as in Example 1, where there is a set of scenarios to consider when abstracting, Definition 1 need to be relaxed to consider a set of sets of facts describing the potential instances, which we leave for future work and focus instead on how they might aid with explanations.

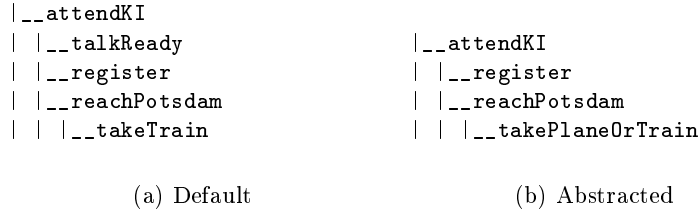


Fig. 1: Explanations obtained by xclingo

The idea is to obtain explanations over the abstracted program. In order to obtain explanations, we make use of the tool **xclingo** [5] which provides explanation graphs of a given answer set. Below we illustrate its use.

Example 2 (Ex. 1 ctd) *For the scenario $\{talkReady, register, takeTrain\}$, we have $AS(P) = \{talkReady, register, takeTrain, reachPotsdam, attendKI\}$. The explanation for the atom `attendKI` to appear in $AS(P)$ is as in Figure 1a. With an abstraction mapping that removes `talkReady` and clusters `takeTrain` and `takePlane` into `takePlaneOrTrain`, and the abstract program Q over the abstracted vocabulary, we obtain the explanation for `attendKI` appearing in $AS(Q)$ as in Figure 1b.*

4 Empirical Study

We hypothesize that the explanations obtained in the abstract programs would help in understandability, due to only containing the relevant details compared to the default explanations of the original programs. In this section, we describe the cognitive experiment in form of an online survey which we designed to test our hypothesis and report on preliminary results.

Task Participants are presented with a classification task of tabular data, where each instance is a set of attributes with a binary class label. Each instance is also visualized by a corresponding image (see Figure 2).

Classification. We present three concept learning tasks to participants, where each task is defined through an answer set program, assigning the instance to the target class when certain conditions are met.⁴

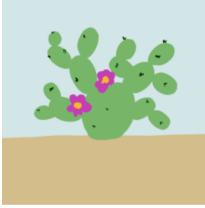
During evaluation, participants are asked to decide for each instance whether it belongs to the target class or not.

Domains. To avoid proactive interference [24] between previously learned rules, we select three different domains, each of which being different biological specimen: flowers, mushrooms, and cacti. Each domain has six domain-specific decision attributes, and the same target attribute “dangerous”.

⁴ Details on the answer set programs and the stimuli description can be found here https://www.dbai.tuwien.ac.at/user/saribat/pub/ki25_supp.pdf

Cactus Attributes:

Spines	Shape	Arms	Stem	Flower	Height	Dangerous
few	flattened-padded	many	thick	yes	short	yes



Explanation:

This cactus is dangerous because all of the following attributes apply:

- adaptive because
 - has arms, since **many** arms
 - no leaf-like shape, since **flattened-padded** shape
 - **short** height
- **few** spines
- **flowers**
- **thick** stem

Fig 2: Example of a learning phase stimulus as presented in the study. The table lists all attributes and values, with a corresponding image. The provided explanation is the default without abstraction.

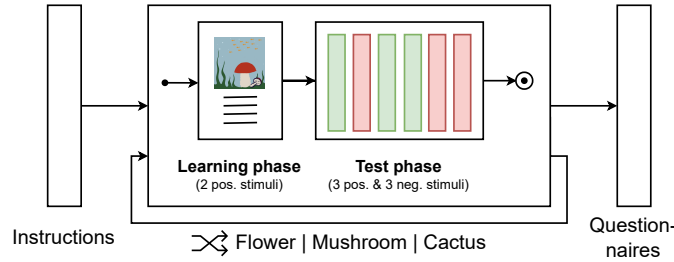


Fig 3: Study Procedure Overview. The order of the three domains and the test phase, i.e. the order of the positive (illustrative in green) and negative (resp. in red) stimuli were random to balance out possible position effects.

Study Design and Procedure The empirical online study is based on a complete 2x2 between-subject factorial design, where participants were randomly assigned to one of the four groups formed by a combination of 2 factors: 'cluster' and 'removal', and the data between these groups were compared to calculate the effect of each factor on our dependent variable.⁵ The main part of the experiment consists of a learning and a test phase (see Figure 3).

Learning Phase At the beginning of each domain, participants receive 2 positive (dangerous) examples together with an explanation why the stimulus was classified dangerous. These explanations differ depending on the assigned group of the experiment. The **control** group has an explanation with no abstraction (see Figure 2). The three abstraction groups, **cluster**, **removal**, and **cluster_removal**, receive an abstracted explanation based on their assigned group. The **cluster**, resp. **removal**, group receives an explanation based on abstraction by clustering

⁵ Informed consent about data protection, anonymity and the right to leave the study at any time was given. At the end of the study, participants had the opportunity to leave further comments and notes.

(e.g. 'water or mud'), resp. removal of irrelevant atoms. The `cluster_removal` group sees explanations with cluster abstraction and removal (e.g., Figure 1b).

Test Phase After the learning phase, unseen stimuli (3 positive, 3 negative) are presented without explanations. Participants need to decide whether the stimulus is dangerous or not and give a rating about their confidence with a slider bar (range 0-100), or choose the option "I don't know". The attribute values were evenly distributed across all stimuli to avoid accumulations of a single attribute value.

Subjective Assessments After the last domain, participants gave ratings on the perceived usefulness of the explanations (5 items, based on a subset of [2]), their active memorization efforts (1 item) and their familiarity with the domain terminologies (1 item). All these items were rated on a Likert-type scale ranging from 1 ('strongly disagree') to 7 ('strongly agree'). Prior knowledge in computer science, logic, mathematics and programming as well as the three domains (7 items) is rated on a scale from 0 ('none') to 4 ('expert').

Participants We recruited 71 participants from two universities whose study program was either Psychology or Computer Science (age mean = 23.1, sd = 4.9; 49 female, 38 male). There is no significant difference in the distribution of gender, age, or self-reported knowledge in computer science among the abstraction groups. Participants were compensated in course credits for their participation.

Analysis We report our statistical tests on the following hypotheses.

Hypothesis 1: Having abstract explanations during the learning phase increases the classification accuracy compared to the `control` group.

Per participant mean accuracy is normally distributed for three of the four abstraction groups, with a Shapiro-Wilk test [21] showing barely significant ($p = .03$) for the `cluster_removal` condition. Homoscedasticity holds. A two-factor ANOVA considering each abstraction as its own factor shows no significant effect for either factor and the interaction (`cluster` $F = 1.30, p = .26$, `removal` $F = .74, p = .39$, `cluster_removal` $F = .24, p = .63$). A one-factor ANOVA considering abstraction groups as a single factor shows no significant effect ($F = .74, p = .53$).

Hypothesis 2: Having abstract explanations during the learning phase increases the self-reported confidence in the classification compared to the `control` group.

Per participant mean accuracy is normally distributed for each abstraction group. Homoscedasticity holds. A two-factor ANOVA as in *Hypothesis 1* shows not significant for either factor and the interaction (`cluster` $F = .29, p = .59$, `removal` $F = .08, p = .78$, `cluster_removal` $F = .20, p = .66$). A one-factor ANOVA considering abstraction groups as a single factor shows no significant effect ($F = .19, p = .91$).

5 Discussion

Motivated by how predicate invention aids in comprehensibility [20,15], we expected a positive effect of abstraction. When examining the results, we detected some unforeseen limitations which might have caused observing no such effect. One major issue, which was also reported among the participants' comments, is the choice of the target variable "dangerous". Due to its semantic meaning, participants may have been inclined to choose dangerousness to avoid repercussions. Another potential limitation might be that tasks requires to be of a certain "intermediate" complexity for explanations to be helpful [1]. The default explanation might have been already easy to follow, requiring no help with abstraction.

We aim to extend our study by changing the target variable to a less triggering term, and increasing the complexity of the provided explanations, e.g., considering clustering of more than two atoms, adding further irrelevant atoms to be removed. The challenge will be to have the study complex enough to show the effect of abstraction but not too complex to not see any effect at all.

Acknowledgments. This work has been supported by the Austrian Science Fund (FWF) project T-1315 (Saribatur).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ai, L., Muggleton, S.H., Hocquette, C., Gromowski, M., Schmid, U.: Beneficial and harmful explanatory machine learning. *Machine Learning* **110**, 695–721 (2021)
2. Bahel, V., Sriram, H., Conati, C.: Personalizing explanations of ai-driven hints to users' cognitive abilities: an empirical evaluation. *CoRR* **abs/2403.04035** (2024). <https://doi.org/10.48550/ARXIV.2403.04035>, <https://doi.org/10.48550/arXiv.2403.04035>
3. Blaha, L.M., Abrams, M., Bibyk, S.A., Bonial, C., Hartzler, B.M., Hsu, C.D., Khemlani, S., King, J., St. Amant, R., Trafton, J.G., Wong, R.: Understanding Is a Process. *Frontiers in Systems Neuroscience* **16** (2022). <https://doi.org/10.3389/fnsys.2022.800280>
4. Brewka, G., Eiter, T., Truszczyński, M.: Answer set programming at a glance. *Commun. ACM* **54**(12), 92–103 (2011)
5. Cabalar, P., Muñoz, B.: Explanation graphs for stable models of labelled logic programs. In: Arias, J., Batsakis, S., Faber, W., Gupta, G., Pacenza, F., Papadakis, E., Robaldo, L., Rückschloß, K., Salazar, E., Saribatur, Z.G., Tachmazidis, I., Weitkämper, F., Wyner, A.Z. (eds.) *Proceedings of the International Conference on Logic Programming 2023 Workshops co-located with the 39th International Conference on Logic Programming (ICLP 2023)*, London, United Kingdom, July 9th and 10th, 2023. *CEUR Workshop Proceedings*, vol. 3437. CEUR-WS.org (2023), <https://ceur-ws.org/Vol-3437/paper3ASP0CP.pdf>
6. Chakraborti, T., Sreedharan, S., Kambhampati, S.: The emerging landscape of explainable automated planning & decision making. In: *Proc. IJCAI*. pp. 4803–4811 (2020), <https://doi.org/10.24963/ijcai.2020/669>

7. Confalonieri, R., Weyde, T., Besold, T.R., del Prado Martín, F.M.: Using ontologies to enhance human understandability of global post-hoc explanations of black-box models. *Artificial Intelligence* **296**, 103471 (2021)
8. Eiter, T., Saribatur, Z.G., Schüller, P.: Abstraction for zooming-in to unsolvability reasons of grid-cell problems. In: *Proc. XAI@IJCAI* (2019)
9. Erdem, E., Gelfond, M., Leone, N.: Applications of answer set programming. *AI Magazine* **37**(3), 53–68 (2016), <http://www.aaai.org/ojs/index.php/aimagazine/article/view/2678>
10. Fandinno, J., Schulz, C.: Answering the “why” in answer set programming - A survey of explanation approaches. *TPLP* **19**(2), 114–203 (2019)
11. Geiger, A., Lu, H., Icard, T., Potts, C.: Causal abstractions of neural networks. *Advances in Neural Information Processing Systems* **34**, 9574–9586 (2021)
12. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. *New Generation Computing* **9**(3), 365–385 (1991)
13. Ho, M.K., Abel, D., Correa, C.G., Littman, M.L., Cohen, J.D., Griffiths, T.L.: People construct simplified mental representations to plan. *Nature* **606**(7912), 129–136 (2022)
14. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. *AIJ* **267**, 1–38 (2019). <https://doi.org/10.1016/j.artint.2018.07.007>
15. Muggleton, S.H., Schmid, U., Zeller, C., Tamaddoni-Nezhad, A., Besold, T.: Ultra-strong machine learning: comprehensibility of programs learned with ILP. *ML* **107**(7), 1119–1140 (2018)
16. Saribatur, Z.G., Eiter, T.: Omission-based abstraction for answer set programs. *Theory Pract. Log. Program.* **21**(2), 145–195 (2021). <https://doi.org/10.1017/S1471068420000095>, <https://doi.org/10.1017/S1471068420000095>
17. Saribatur, Z.G., Eiter, T., Schüller, P.: Abstraction for non-ground answer set programs. *Artif. Intell.* **300**, 103563 (2021). <https://doi.org/10.1016/j.artint.2021.103563>, <https://doi.org/10.1016/j.artint.2021.103563>
18. Saribatur, Z.G., Knorr, M., Gonçalves, R., Leite, J.: On abstracting over the irrelevant in answer set programming. In: Marquis, P., Ortiz, M., Pagnucco, M. (eds.) *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024, Hanoi, Vietnam. November 2–8, 2024* (2024). <https://doi.org/10.24963/KR.2024/61>, <https://doi.org/10.24963/kr.2024/61>
19. Saribatur, Z.G., Woltran, S.: Foundations for Projecting Away the Irrelevant in ASP Programs. In: *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning*. pp. 614–624. IJCAI Organization (2023). <https://doi.org/10.24963/kr.2023/60>
20. Schmid, U., Zeller, C., Besold, T., Tamaddoni-Nezhad, A., Muggleton, S.: How Does Predicate Invention Affect Human Comprehensibility? In: *Inductive Logic Programming. ILP 2016*. pp. 52–67. Springer (2016). https://doi.org/10.1007/978-3-319-63342-8_5
21. Shapiro, S.S., Wilk, M.B.: An analysis of variance test for normality (complete samples). *Biometrika* **52**(3/4), 591–611 (1965), <http://www.jstor.org/stable/2333709>
22. Siebers, M., Schmid, U.: Please delete that! why should I? *KI* **33**(1), 35–44 (2019)
23. Poesia Reis e Silva, G., Goodman, N.: Left to the reader: Abstracting solutions in mathematical reasoning. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. vol. 44 (2022)
24. Underwood, B.J.: Interference and forgetting. *Psychological Review* **64**(1), 49 (1957)