Conflict Generalisation in ASP

Learning Correct and Effective Non-Ground Constraints (Extended Abstract*)

Richard Taupe[1,2](✉), Antonius Weinzierl[3], and Gerhard Friedrich[2]

[1] Siemens AG Österreich
richard.taupe@siemens.com
[2] Alpen-Adria-Universität Klagenfurt
gerhard.friedrich@aau.at
[3] TU Wien (Vienna University of Technology), Austria
weinzierl@kr.tuwien.ac.at

Conflict-Driven Nogood Learning (CDNL) [6, 10, 14] is a major success factor for high-performance state-of-the-art ASP systems. When a conflict occurs, new propositional nogoods are learned that prevent the same conflict from re-occurring, which improves search performance. We present an extension of CDNL that learns non-ground constraints. The idea is that whole parts of the search tree can be pruned when these learned constraints are added to the original program. We aim to derive non-ground constraints from the problem encoding that are valid for all possible inputs and which can be employed to speed up solving new instances.

CDNL and Explanation-Based Learning (EBL) [2, 7, 8, 12] are our starting point. EBL is a well-known logic-based machine learning technique which learns first-order rules that are entailed by the background knowledge (in our case, the problem encoding). We combine CDNL with EBL to learn non-ground nogoods while solving prior problem instances. Since the number of generalised nogoods can be overwhelming, choosing those that will actually pay off is particularly challenging. Our basic idea is to generalise those non-ground conflicts that occur most often, i.e., we generalise propositional nogoods learned from frequently violated nogoods. The underlying assumption is that nogoods learned from frequent conflicts will also be able to prevent many conflicts.

Previous work by Lühne et al. [4, 11] used different means to generalise learned nogoods for future re-use: In their approach, propositional constraints are extracted while solving, then generalized by minimization and abstraction, and finally validated by means of proof techniques.

We use a realistic hardware configuration example (the House Reconfiguration Problem (HRP) [3, 13]) and a graph colouring problem for demonstration and experimentation purposes. In simplified terms, the HRP is about assigning things to cabinets and cabinets to rooms s.t. every room contains things belonging to only one person. This is expressed by the following rule and constraint:

$$\mathrm{personTOroom}(P,R) \leftarrow \mathrm{personTOthing}(P,T),\ \mathrm{cabinetTOthing}(C,T),\ \mathrm{roomTOcabinet}(R,C).$$
$$\leftarrow \mathrm{personTOroom}(\mathit{P1},R),\ \mathrm{personTOroom}(\mathit{P2},R),\ \mathit{P1} < \mathit{P2}.$$

* This is an extended abstract of research already published in [15].

Employing our method, the following redundant constraint can be learned when the above constraint is violated:

$$\leftarrow \text{cabinetTOthing}(C, \mathit{T1}),\ \text{personTOthing}(\mathit{P1}, \mathit{T1}),\\ \text{cabinetTOthing}(C, \mathit{T2}),\ \text{personTOthing}(\mathit{P2}, \mathit{T2}),\ \mathit{P1} < \mathit{P2}.$$

This makes the knowledge explicit that no two things belonging to different persons may be placed in the same cabinet.

Experimental results with ALPHA [9, 16], DLV 2.0 [1], and CLINGO [5] show that both lazy-grounding and ground-and-solve systems benefit from our approach. By adding learned constraints to the problem encodings, more instances can be solved and/or the problems can be solved faster. Learning itself has been implemented within ALPHA[4] and requires low computational resources.

It seems natural to assume that many domains feature redundant constraints that may not be obvious to a human modeller. We therefore see our approach mainly as a tool to support the design of efficient answer-set programs.

Acknowledgments. This work has been conducted in the scope of the research project *DynaCon (FFG-PNr.: 861263)*, which is funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under the program "ICT of the Future" between 2017 and 2020, and in the scope of the research project Productive4.0, which is funded by EU-ECSEL under grant agreement no737459.

References

1. Alviano, M., Calimeri, F., Dodaro, C., Fuscà, D., Leone, N., Perri, S., Ricca, F., Veltri, P., Zangari, J.: The ASP system DLV2. In: LPNMR. LNCS, vol. 10377, pp. 215–221. Springer (2017)
2. DeJong, G., Mooney, R.J.: Explanation-based learning: An alternative view. Mach. Learn. **1**(2), 145–176 (1986)
3. Friedrich, G., Ryabokon, A., Falkner, A.A., Haselböck, A., Schenner, G., Schreiner, H.: (Re)configuration using answer set programming. In: Configuration Workshop. CEUR-WS.org (2011)
4. Gebser, M., Kaminski, R., Kaufmann, B., Lühne, P., Romero, J., Schaub, T.: Answer set solving with generalized learned constraints. In: ICLP (Technical Communications). OASICS, vol. 52, pp. 9:1–9:15 (2016)
5. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Clingo = ASP + control: Preliminary report. CoRR **abs/1405.3694** (2014)
6. Gebser, M., Kaufmann, B., Schaub, T.: Conflict-driven answer set solving: From theory to practice. Artif. Intell. **187**, 52–89 (2012)
7. Van Harmelen, F., Bundy, A.: Explanation-based generalisation = partial evaluation. Artif. Intell. **36**(3), 401–412 (1988)
8. Hirsh, H.: Explanation-based generalization in a logic-programming environment. In: IJCAI. pp. 221–227. Morgan Kaufmann (1987)
9. Leutgeb, L., Weinzierl, A.: Techniques for efficient lazy-grounding ASP solving. In: DECLARE. LNCS, vol. 10997, pp. 132–148. Springer (2017)
10. Lintao Zhang, Madigan, C.F., Moskewicz, M.H., Malik, S.: Efficient conflict driven learning in a boolean satisfiability solver. In: ICCAD. pp. 279–285. IEEE (2001)

[4] https://github.com/alpha-asp/Alpha

11. Lühne, P.: Generalizing Learned Knowledge in Answer Set Solving. Master's thesis, Hasso Plattner Institute (2015), https://www.luehne.de/theses/generalizing-learned-knowledge-in-answer-set-solving.pdf
12. Mitchell, T.M., Keller, R.M., Kedar-Cabelli, S.T.: Explanation-based generalization: A unifying view. Mach. Learn. **1**(1), 47–80 (1986)
13. Ryabokon, A.: Knowledge-based (Re)configuration of Complex Products and Services. Ph.D. thesis, Alpen-Adria-Universität Klagenfurt (2015), http://netlibrary.aau.at/urn:nbn:at:at-ubk:1-26431
14. Silva, J.P.M., Lynce, I., Malik, S.: Conflict-driven clause learning SAT solvers. In: Handbook of Satisfiability, pp. 131–153. IOS Press (2009)
15. Taupe, R., Weinzierl, A., Friedrich, G.: Conflict generalisation in ASP: Learning correct and effective non-ground constraints. TPLP **20**(5), 799–814 (2020)
16. Weinzierl, A.: Blending lazy-grounding and CDNL search for answer-set solving. In: LPNMR. LNCS, vol. 10377, pp. 191–204. Springer (2017)