Rushing and Strolling among Answer Sets – Navigation Made Easy (Extended Abstract)

Johannes Klaus Fichte[1] , **Sarah Alice Gaggl**[2] , **Dominik Rusovac**[2]
[1]TU Wien, Research Unit Database and Artificial Intelligence, Austria
[2]TU Dresden, Logic Programming and Argumentation Group, Germany
firstname.lastname@{tuwien.ac.at,tu-dresden.de}

Relevance to KR.

Our work presents a novel approach together with a prototypical implementation to make sense of incomprehensible solution spaces of answer set programs. As ASP has been in the interest of the KR community for quite some time, we believe that our framework paves the way for further work with interesting applications within the field of knowledge representation and reasoning. In particular, weighted faceted answer set navigation applies to configuration problems and planning problems, where it might be of interest to know to what amount certain components (partial solutions) restrict the solution space. Quantifying the effect of a navigation step involves quantitative reasoning and model counting, which may attract members of the quantitative logic programming and model counting community, respectively. Even more so, our approach can be used to explore tremendously large solution spaces for which enumerating all answer sets is infeasible. Dealing with a large number of answer sets and incomprehensible amounts of solutions is a challenge for ASP technology, in general, which is why we think that our work is interesting for regular KR'22 attendants.

1 Introduction

Answer set programming (ASP) is a popular declarative programming paradigm, which is widely used for knowledge representation and problem solving (Brewka, Eiter, and Truszczyński 2011). Oftentimes an answer set program results in a large number of answer sets, which means that the solution space of a program can easily become infeasible to comprehend. Thus, trying to gradually identify specific answer sets can be a quite tedious task. We propose a framework, called *weighted faceted navigation*, that goes beyond simple search for one solution, and enables users to systematically navigate through the solution space in order to explore desirable solutions. Besides characterizing the weight of a facet, we demonstrate that weighted faceted navigation is hard. Finally, we provide an interactive prototypical implementation of our approach on top of the `clingo` solver (Gebser et al. 2018).

Background. We follow standard definitions of propositional answer set programs and assume familiarity with disjunctive logic programs and stable model semantics (Gelfond and Lifschitz 1991) (Gelfond and Lifschitz 1988). A *disjunctive logic program* Π is a finite set of rules of the form $\alpha_0 \mid \ldots \mid \alpha_k \leftarrow \alpha_{k+1}, \ldots, \alpha_m, {\sim}\alpha_{m+1}, \ldots, {\sim}\alpha_n$ over atoms α_i with $0 \leq k \leq m \leq n$. We denote answer sets of a program Π by $\mathcal{AS}(\Pi)$. Further, we denote *brave consequences* by $\mathcal{BC}(\Pi) \coloneqq \bigcup \mathcal{AS}(\Pi)$ and *cautious consequences* by $\mathcal{CC}(\Pi) \coloneqq \bigcap AS(\Pi)$. We denote the *facets* of Π by $\mathcal{F}(\Pi) \coloneqq \mathcal{F}^+(\Pi) \cup \mathcal{F}^-(\Pi)$ where $\mathcal{F}^+(\Pi) \coloneqq \mathcal{BC}(\Pi) \setminus \mathcal{CC}(\Pi)$ denotes *inclusive facets* and $\mathcal{F}^-(\Pi) \coloneqq \{\overline{\alpha} \mid \alpha \in \mathcal{F}^+(\Pi)\}$ denotes *exclusive facets* of Π. For further details on faceted answer set navigation, we refer to (Alrabbaa, Rudolph, and Schweizer 2018).

2 Weighted Faceted Navigation

A finite sequence $\delta \coloneqq \langle f_0, \ldots, f_n \rangle \in \Delta^\Pi$ of navigation steps towards facets $f_i \in F(\Pi)$ is called a *route* over Π. The empty route is denoted by ϵ, and δ is a *subroute* of δ', denoted by $\delta \sqsubseteq \delta'$, whenever if $f_i \in \delta$, then $f_i \in \delta'$. A navigation step, denoted by $\Pi^{\langle f \rangle}$, is performed by activating a facet $f \in \{\alpha, \overline{\alpha}\}$, which means that we add an integrity constraint to Π such that $\Pi^{\langle f \rangle} = \Pi \cup \{\leftarrow {\sim}\alpha.\}$, if $f = \alpha$, and otherwise, if $f = \overline{\alpha}$, $\Pi^{\langle f \rangle} = \Pi \cup \{\leftarrow \alpha.\}$. Thereby we land in a subset $AS(\Pi^{\langle f \rangle}) \subset AS(\Pi)$ of solutions (sub-space) where each solution includes/excludes the specific inclusive/exclusive facet. To avoid stepping into empty sub-spaces we distinguish between so called *safe* routes denoted by $\Delta^\Pi_s \coloneqq \{\delta \in \Delta^\Pi \mid AS(\Pi^\delta) \neq \emptyset\}$ and *unsafe* routes $\Delta^\Pi \setminus \Delta^\Pi_s$. Faceted navigation is possible as long as we are on a safe route. In case we take an unsafe route δ, we can proceed navigating toward some facet f by taking a possible *redirection* of δ with respect to f from $R(\delta, f) \coloneqq \{\delta' \sqsubseteq \delta \mid f \in \delta', AS(\Pi^{\delta'}) \neq \emptyset\} \cup \{\epsilon\}$. The so called *weight* $\omega_\#(f, \Pi^\delta, \delta')$ of a facet $f \in F(\Pi)$ on route δ and with respect to redirection $\delta' \in R(\delta, f)$ as defined by

$$\omega_\#(f, \Pi^\delta, \delta') \coloneqq \begin{cases} \#(\Pi^\delta) - \#(\Pi^{\delta'}), & \text{if } \langle \delta, f \rangle \not\in \Delta_s^{\Pi^\delta} \\ & \text{and } \delta' \neq \epsilon; \\ \#(\Pi^\delta) - \#(\Pi^{\langle \delta, f \rangle}), & \text{otherwise.} \end{cases}$$

serves the purpose of quantifying the effect of activating f. It is a parameter that indicates to what a amount the activation of f restricts the solution space with respect to the

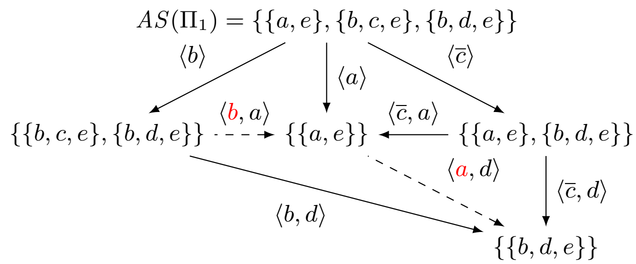

Figure 1: Navigating on several routes through answer sets of program Π_1. Dashed lines indicate redirections where red colored facets were retracted.

associated *weighting function* $\# : \{\Pi^\delta \mid \delta \in \Delta^\Pi\} \to \mathbb{N}$ for which $\#(\Pi^\delta) > 0$, if $|AS(\Pi)| \geq 2$. While the *absolute weight* $\omega_{\#AS}$ (counting answer sets) turns out to be the natural choice, counting answer sets is hard (Fichte et al. 2017). Thus, we investigated the *relative supp weight* $\omega_{\#S}$ (counting supported models) and the *relative facet-counting weight* $\omega_{\#F}$ (counting facets), which provide cheaper methods for quantifying the effect of a navigation step. In the original paper we prove, in a systematic comparison, which weights can be employed under the search space navigation operations and illustrate the computational complexity for computing the weights.

Example 1. *Consider* $\Pi_1 = \{a \,|\, b; c \,|\, d; d \leftarrow b; e\}$. *We have* $AS(\Pi_1) = \{\{a, e\}, \{b, c, e\}, \{b, d, e\}\}$ *and* $F(\Pi_1) = \{a, b, c, d, \overline{a}, \overline{b}, \overline{c}, \overline{d}\}$. *As illustrated by Figure 1, facets* b *and* $\overline{c}$ *have the same absolute weight of* 1*, that is, we zoom-in by one solution, but their facet-counting weights differ. While* b *has facet-counting weight* 4*, the facet counting weight of* $\overline{c}$ *is* 2*. The reason is that counting facets indicates to what amount solutions converge, that is, since the solutions in* $AS(\Pi_1^{\langle b \rangle}) = \{\{b, c, e\}, \{b, d, e\}\}$ *with facets* $\{c, d, \overline{c}, \overline{d}\}$ *are more similar to each other than the solutions in sub-space* $AS(\Pi_1^{\langle \overline{c} \rangle}) = \{\{a, e\}, \{b, d, e\}\}$ *with facets* $\{a, b, d, \overline{a}, \overline{b}, \overline{d}\}$*, inclusive facet* b *has a higher facet-counting weight than exclusive facet* $\overline{c}$*.*

Weighted faceted navigation enables users to explore solutions to a problem by consciously zooming in or out of sub-spaces of solutions at a certain configurable pace. In the full version, we furthermore introduce so called *navigation modes*, which can be understood as search strategies, in order to for instance find a unique desirable solution as quick as possible. In particular, our approach applies to configuration problems and planning problems, where it might be of interest to know to what amount certain components restrict the solution space. However, weighted faceted navigation is in general useful for making sense of solution spaces, as it provides insights on the properties that certain facets of a problem have.

Experiments. To study the feasibility of our framework, we implemented `fasb` (Fichte, Gaggl, and Rusovac 2021b), an interactive prototype of our framework on top of the `clingo` solver (Gebser et al. 2011; Gebser et al. 2018). We conducted experiments (Fichte, Gaggl, and Rusovac 2021a) in order to (i) demonstrate the feasibility of our approach by navigating through the solution space of a PC configuration encoding with more than a billion solutions; and (ii) demonstrate that the feasibility of our approach depends on the complexity of the given program, by analyzing runtimes of navigation over two encodings of different complexity but with an identical solution space.

3 Conclusion

Establishing fundamental notions such as routes, navigation modes and weights of facets for faceted answer set navigation, we provide a formal and dynamic framework for navigating through the solution space of an answer set program in a systematic way. Our framework is intended as an additional layer on top of a solver, which adds functionality for exploring the search space with quantitative considerations expressed by weights of facets. Our prototypical implementation demonstrates the feasibility of our framework for an incomprehensible solution space.

References

Alrabbaa, C.; Rudolph, S.; and Schweizer, L. 2018. Faceted answer-set navigation. In Benzmüller, C.; Ricca, F.; Parent, X.; and Roman, D., eds., *Proc. of the 2nd Int. Joint Conf. on Rules and Reasoning (RuleML+RR'18)*, 211–225. Springer.

Brewka, G.; Eiter, T.; and Truszczyński, M. 2011. Answer set programming at a glance. *Communications of the ACM* 54(12):92–103.

Fichte, J. K.; Hecher, M.; Morak, M.; and Woltran, S. 2017. Answer set solving with bounded treewidth revisited. In Balduccini, M., and Janhunen, T., eds., *Proc. of the 14th Int. Conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR 2017)*, volume 10377 of *LNCS*, 132–145. Springer.

Fichte, J. K.; Gaggl, S. A.; and Rusovac, D. 2021a. Rushing and Strolling among Answer Sets - Navigation Made Easy (Experiments). https://doi.org/10.5281/zenodo.5780050.

Fichte, J. K.; Gaggl, S. A.; and Rusovac, D. 2021b. Rushing and Strolling among Answer Sets - Navigation Made Easy (Faceted Answer Set Browser fasb). https://doi.org/10.5281/zenodo.5767981.

Gebser, M.; Kaminski, R.; König, A.; and Schaub, T. 2011. Advances in gringo series 3. In Delgrande, J. P., and Faber, W., eds., *Proc. of the Int. Conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR'11)*, 345–351. Springer.

Gebser, M.; Kaminski, R.; Kaufmann, B.; Lühne, P.; Obermeier, P.; Ostrowski, M.; Romero, J.; Schaub, T.; Schellhorn, S.; and Wanko, P. 2018. The potsdam answer set solving collection 5.0. *Künstliche Intell.* 32(2-3):181–182.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In Kowalski, R. A., and Bowen, K. A., eds., *Proc. of the 5th Int. Conf. and Symposium on Logic Programming (ICLP/SLP'88)*, volume 2, 1070–1080. MIT Press.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Comput.* 9(3/4):365–386.