Translating monotone aggregates from Ferraris into Gelfond-Zhang semantics: Work in progress

Pedro Cabalar[1], Jorge Fandinno[2], Javier Romero[3], Nicolas Rühling[3], Torsten Schaub[3], and Philipp Wanko[3]

[1] University of A Coruña, Spain
[2] University of Nebraska at Omaha, NE, USA
[3] University of Potsdam, Germany

1 Introduction

The Logic of Here-and-There with constraints (HT_c, [4, 2, 3]) extends the formal foundations of Answer Set Programming (ASP) with a framework for constraint values and variables which allows to assign default values to constraint variables or to leave them undefined. Recent work incorporated so-called *conditional constraints* [3], thereby providing a semantics for (conditional) aggregates with constraint variables. Two alternative semantics are studied that are respectively based on the principle of Gelfond-Zhang (GZ; [6]) and Ferraris (F; [5]) semantics for aggregates without constraint variables. Its results have subsequently been applied in the implementation of *fclingo*[4], a hybrid solver for ASP modulo conditional linear constraints with founded variables. However, unlike *clingo* which uses an aggregate semantics based on the F-semantics, the results from [3] currently only make it possible to use GZ-aggregates in *fclingo*. While the F-semantics guarantees *definedness* of aggregates, the GZ-semantics is based on the *Vicious Circle Principle* and therefore prohibits derivations where the body of a rule depends on some object in its head. Consider for example the rule $a \leftarrow \texttt{count}\{a\} \geq 0$. While under the F-semantics (and thus in *clingo*) this gives us one stable model $\{a\}$, the GZ-semantics prohibits a derivation and we get no stable models. We would like to achieve the same well known aggregate behavior from *clingo* in *fclingo*.

For this, we envision a two-step translation where first any non-convex aggregates are translated into monotone ones by using results from [1]. Then, monotone aggregates under the F-semantics are translated into a set of rules under the GZ-semantics such that stable models are preserved under projection to the original variables. This abstract conjectures a theorem for the second part of the translation.

2 Translating monotone aggregates

Due to lack of space, we provide only a few essential definitions from HT_c needed for further reading and refer the reader to [3] for more details. A *conditional*

[4] `https://github.com/potassco/fclingo`, a full system description is planned for the future

term is of form $(\tau|\tau'\colon\varphi)$ with τ and τ' terms and φ a formula representing the condition. All three of τ, τ' and φ are assumed to be condition-free, ie. they do not contain any conditional terms themselves. Intuitively, the conditional term evaluates to τ when the condition holds and τ' otherwise. However, there are differences in the evaluation depending on the chosen semantics which can be based either on the *vicious circle* (*vc*) or *definedness principle* (*df*), corresponding to the GZ- and F-semantics, respectively. More precisely, for both semantics we have that for $\langle h,t\rangle \models_\kappa \varphi$ the conditional term evaluates to τ and for $\langle t,t\rangle \not\models_\kappa \varphi$, the conditional term evaluates to τ'. A different behavior arises for the case where $\langle t,t\rangle \models_\kappa \varphi$ but $\langle h,t\rangle \not\models_\kappa \varphi$ (the condition is assumed to be true but cannot be proven). Then, under *df* the conditional term evaluates to τ' while under *vc* it evaluates to undefined.

We assume that there exists a *selection function* κ which assigns to each conditional term either *vc* or *df* and we denote the set of κ-stable models of a theory Γ by $SM_\kappa(\Gamma)$. Further, we define the set of variables of a theory Γ by $vars(\Gamma)$ and the set of κ-stable models projected onto a set of variables as $SM_\kappa(\Gamma)|_X = \{t|_X \mid t \in SM_\kappa(\Gamma)\}$. For now, aggregates have the form of *(conditional) linear constraints* (or constraint atoms for short) which are a comparison of the form $\alpha \circ \beta$ such that $\circ \in \{\leq, <, =, \neq\}$ and α and β are sums of (possibly conditional) terms.

For the conjecture, we are working with a syntactic definition of monotone constraint atoms. We write $c[s/s']$ to represent the syntactic replacement in c of subexpression s by s'.

Definition 1 (Statically monotone linear constraint). *Let c be a constraint atom, $s = (\tau|\tau'\colon\varphi)$ a conditional term occurring in c and κ be a selection function.*

We say that s is statically κ-monotone *wrt c if every $h \subseteq h' \subseteq t$ satisfy that $\langle h,t\rangle \models_\kappa c[s/\tau']$ implies $\langle h',t\rangle \models_\kappa c[s/\tau]$. We say that c is* statically κ-monotone *if all conditional terms occurring in it are statically κ-monotone wrt c.*

Example 1. Consider the constraint atom $(1|0\colon x=\mathbf{t}) + (1|0\colon y=\mathbf{t}) \geq 1$. The first term $(1|0\colon x=\mathbf{t})$ is statically *df*-monotone wrt the constraint as for any ht-interpretation $\langle h,t\rangle$ which satisfies $0 + (1|0\colon y=\mathbf{t}) \geq 1$ it follows that any $\langle h',t\rangle$ such that $h \subset h' \subseteq t$ satisfies $1 + (1|0\colon y=\mathbf{t}) \geq 1$. By the same arguments, the second term and therefore the whole constraint atom is statically *df*-monotone.

We define the following translation for logic programs

Definition 2 (Translation). *For a logic program Γ, we translate every rule r as follows*

1. $\neg\neg r$
2. $p_i \vee \neg p_i \leftarrow \varphi_i$ *for every conditional term* $s_i = (\tau_i|\tau_i'\colon\varphi_i)$ *in* r
3. r'

where the p_i are fresh, propositional atoms for each φ_i and r' is the result of replacing each $s_i = (\tau_i|\tau_i'\colon\varphi_i)$ in r by $s_i' = (\tau_i|\tau_i'\colon p_i)$. We denote the result of the translation by $\Phi(\Gamma)$.

We conjecture the following theorem which would allow us to use the behavior of F-aggregates inside the GZ-semantics of *fclingo* for monotone aggregates

Conjecture 1. Let Γ be a logic program with statically df-monotone constraint atoms in the body. Then,

$$SM_{df}(\Gamma) = SM_{vc}(\Phi(\Gamma))|_{vars(\Gamma)}$$

Next, we illustrate the translation and the conjecture with a small example.

Example 2. Consider the following program

$$\Gamma = \begin{cases} x = \mathbf{t}. \\ y = \mathbf{t} \leftarrow (1|0\colon x = \mathbf{t}) + (1|0\colon y = \mathbf{t}) \geq 1 \end{cases}$$

As shown before, the constraint atom $(1|0\colon x = \mathbf{t}) + (1|0\colon y = \mathbf{t}) \geq 1$ is statically df-monotone. The program has a single df-stable model $\{x \mapsto \mathbf{t}, y \mapsto \mathbf{t}\}$, but no vc-stable model as there is a smaller ht-model $\langle\{x \mapsto \mathbf{t}\}, \{x \mapsto \mathbf{t}, y \mapsto \mathbf{t}\}\rangle$ which satisfies the second rule under the vc-semantics by making the constraint atom undefined (The second term $(1|0\colon y = \mathbf{t})$ evaluates to undefined which makes the whole constraint atom undefined). Translating Γ gives

$$\Phi(\Gamma) = \begin{cases} x = \mathbf{t}. \\ \bot \leftarrow (1|0\colon x = \mathbf{t}) + (1|0\colon y = \mathbf{t}) \geq 1, \neg(y = \mathbf{t}). \\ p_x \vee \neg p_x \leftarrow x = \mathbf{t}. \\ p_y \vee \neg p_y \leftarrow y = \mathbf{t}. \\ y = \mathbf{t} \leftarrow (1|0\colon p_x = \mathbf{t}) + (1|0\colon p_y = \mathbf{t}) \geq 1. \end{cases}$$

where p_x and p_y are fresh, propositional constants.

The translation now has vc-stable model $\{x \mapsto \mathbf{t}, y \mapsto \mathbf{t}, p_x \mapsto \mathbf{t}\}$ which when projected to the original variables of Γ gives the original df-stable model $\{x \mapsto \mathbf{t}, y \mapsto \mathbf{t}\}$.

References

1. Alviano, M., Faber, W., Gebser, M.: From non-convex aggregates to monotone aggregates in ASP. In: Kambhampati [7], pp. 4100–4104
2. Cabalar, P., Fandinno, J., Schaub, T., Wanko, P.: An ASP semantics for constraints involving conditional aggregates. In: De Giacomo, G., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J. (eds.) Proceedings of the Twenty-fourth European Conference on Artificial Intelligence (ECAI'20). pp. 664–671. IOS Press (2020)
3. Cabalar, P., Fandinno, J., Schaub, T., Wanko, P.: A uniform treatment of aggregates and constraints in hybrid ASP. In: Calvanese, D., Erdem, E., Thielscher, M. (eds.) Proceedings of the Seventeenth International Conference on Principles of Knowledge Representation and Reasoning (KR'21). pp. 193–202. AAAI Press (2020)
4. Cabalar, P., Kaminski, R., Ostrowski, M., Schaub, T.: An ASP semantics for default reasoning with constraints. In: Kambhampati [7], pp. 1015–1021. https://doi.org/10.5555/3060621.3060762

5. Ferraris, P.: Logic programs with propositional connectives and aggregates. ACM Transactions on Computational Logic **12**(4), 25:1–25:40 (2011)
6. Gelfond, M., Zhang, Y.: Vicious circle principle and logic programs with aggregates. Theory and Practice of Logic Programming **14**(4-5), 587–601 (2014)
7. Kambhampati, R. (ed.): Proceedings of the Twenty-fifth International Joint Conference on Artificial Intelligence (IJCAI'16). IJCAI/AAAI Press (2016)

This article was processed using the comments style on November 21, 2024.
There remain 0 comments to be processed.