The Complexity Boundary of Answer Set Programming with Generalized Atoms under the FLP Semantics

Mario Alviano and Wolfgang Faber

Department of Mathematics
University of Calabria
87030 Rende (CS), Italy
{alviano,faber}@mat.unical.it

Abstract. In recent years, Answer Set Programming (ASP), logic programming under the stable model or answer set semantics, has seen several extensions by generalizing the notion of an atom in these programs: be it aggregate atoms, HEX atoms, generalized quantifiers, or abstract constraints, the idea is to have more complicated satisfaction patterns in the lattice of Herbrand interpretations than traditional, simple atoms. In this paper we refer to any of these constructs as generalized atoms. It is known that programs with generalized atoms that have monotonic or antimonotonic satisfaction patterns do not increase complexity with respect to programs with simple atoms (if satisfaction of the generalized atoms can be decided in polynomial time) under most semantics. It is also known that generalized atoms that are nonmonotonic (being neither monotonic nor antimonotonic) can, but need not, increase the complexity by one level in the polynomial hierarchy if non-disjunctive programs under the FLP semantics are considered. In this paper we provide the precise boundary of this complexity gap: programs with convex generalized atom never increase complexity, while allowing a single non-convex generalized atom (under reasonable conditions) always does. We also discuss several implications of this result in practice.

1 Introduction

Various extensions of the basic Answer Set Programming language have been proposed by allowing more general atoms in rule bodies, for example aggregate atoms, HEX atoms, dl-atoms, generalized quantifiers, or abstract constraints. The FLP semantics defined in [5] provides a semantics to all of these extensions, as it treats all body elements in the same way (and it coincides with the traditional ASP semantics when no generalized atoms are present). The complexity analyses reported in [5] show that in programs with single simple atom rule heads, the main complexity tasks do not increase when the generalized atoms present are monotonic or antimonotonic ($coNP$-complete for cautious reasoning), but there is an increase in complexity otherwise (Π_2^P-complete for cautious reasoning). These complexity results hold under the assumptions of dealing with propositional programs and that determining the satisfaction of a generalized atom in an interpretation can be done in polynomial time. Also throughout this paper, we will work under these assumptions.

However, there are several examples of generalized atoms that are nonmonotonic (neither monotonic nor antimonotonic), for which reasoning is still in $coNP$. Examples

P. Cabalar and T.C. Son (Eds.): LPNMR 2013, LNAI 8148, pp. 67–72, 2013.

for such easy nonmonotonic generalized atoms are count aggregates with an equality guard, cardinality constraints with upper and lower bounds, or weight constraints with non-negative weights and upper and lower guards. All of these have the property of being convex, which can be thought of as a conjunction of monotonic and antimonotonic. Convex generalized atoms have been studied in [7], and it is implicit in there, and in general not hard to see that there is no increase in complexity in the presence of atoms of this kind.

In this paper, we show that convex generalized atoms are indeed the only ones for which cautious reasoning under the FLP semantics remains in $coNP$. Our main result is that when a language allows any kind of non-convex generalized atom, Π_2^P-hardness of cautious reasoning can be established. We just require two basic properties of generalized atoms: they should be closed under renaming of atoms, and only a subset of all available (simple) atoms should be relevant for the satisfaction of a single generalized atom (this subset is the domain of the generalized atom). All types of generalized atoms that we are aware of meet these assumptions. Essentially, the first requirement means that it is possible to rename the simple atoms in the representation of a generalized atom while retaining its semantic properties, while the second means that modifying truth values of simple atoms that are irrelevant to the general atom does not alter its semantic behavior.

Our result has several implications that are discussed in more detail in section 4. The main ones concern implementation and rewriting issues, but also simpler identification of the complexity of ASP extensions. In the following, we will present a simple language for our study in section 2; essentially, we view a rule body as a single "structure" that takes the role of a generalized atom (sufficiently detailed and expressive, since the FLP semantics treats rule bodies monolithically anyway and because convexity is closed under conjunction). In section 3 we present our main theorem and its proof, and in section 4 we wrap up.

2 Syntax and Semantics

Let $\mathcal{U}$ be a fixed, countable set of propositional atoms. An interpretation I is a subset of $\mathcal{U}$. A structure S on $\mathcal{U}$ is a mapping of interpretations into Boolean truth values. Each structure S has an associated domain $D_S \subset \mathcal{U}$, indicating those atoms that are relevant to the structure. A general rule r is of the following form:

$$H(r) \leftarrow B(r) \tag{1}$$

where $H(r)$ is a propositional atom in $\mathcal{U}$ referred as the head of r, and $B(r)$ is a structure on $\mathcal{U}$ called the body of r. No particular assumption is made on the syntax of $B(r)$, in the case of normal propositional logic programs these structures are conjunctions of literals. We assume that structures are closed under propositional variants, that is, for any structure S, also $S\sigma$ is a structure for any bijection $\sigma : \mathcal{U} \to \mathcal{U}$, the associated domain is $D_{S\sigma} = \{\sigma(a) \mid a \in D_S\}$.

A general program P is a set of general rules. By datalog^S we refer to the class of programs that may contain only the following rule bodies: structures corresponding to conjunctions of atoms, S, or any of its variants $S\sigma$.

Let $I \subseteq \mathcal{U}$ be an interpretation. I is a model for a structure S, denoted $I \models S$, if S maps I to true. Otherwise, if S maps I to false, I is not a model of S, denoted $I \not\models S$. We require that atoms outside the domain of S are irrelevant for modelhood, that is, for any interpretation I and $X \subseteq \mathcal{U} \setminus D_S$ it holds that $I \models S$ if and only if $I \cup X \models S$. Moreover, for any bijection $\sigma : \mathcal{U} \to \mathcal{U}$, let $I\sigma = \{\sigma(a) \mid a \in I\}$, and we require that $I\sigma \models S\sigma$ if and only if $I \models S$. I is a model of a rule r of the form (1), denoted $I \models r$, if $H(r) \in I$ whenever $I \models B(r)$. I is a model of a program P, denoted $I \models P$, if $I \models r$ for every rule $r \in P$.

The FLP reduct P^I of a program P with respect to I is defined as the set $\{r \mid r \in P \wedge I \models B(r)\}$. I is a stable model of P if $I \models P^I$ and for each $J \subset I$ it holds that $J \not\models P^I$. A propositional atom a is a cautious consequence of a program P, denoted $P \models_c a$, if a belongs to all stable models of P.

Structures can be characterized in terms of *monotonicity* as follows: Let S be a structure. S is monotonic if for all pairs X, Y of interpretations such that $X \subset Y$, $X \models S$ implies $Y \models S$. S is antimonotonic if for all pairs Y, Z of interpretations such that $Y \subset Z$, $Z \models S$ implies $Y \models S$. S is convex if for all triples X, Y, Z of interpretations such that $X \subset Y \subset Z$, $X \models S$ and $Z \models S$ implies $Y \models S$.

3 Main Complexity Result

It is known that cautious reasoning over answer set programs with generalized atoms under FLP semantics is Π_2^P-complete in general. It is also known that the complexity drops to $coNP$ if structures in body rules are constrained to be convex. This appears to be "folklore" knowledge and can be argued to follow from results in [7]. An easy way to see membership in $coNP$ is that all convex structures can be decomposed into a conjunction of a monotonic and an antimonotonic structure, for which membership in $coNP$ has been shown in [5].

We will therefore focus on showing that convex structures define the precise boundary between the first and the second level of the polynomial hierarchy. In fact, we prove that any extension of datalog by at least one non-convex structure and its variants raises the complexity of cautious reasoning to the second level of the polynomial hierarchy.

The hardness proof is similar to the reduction from 2QBF to disjunctive logic programs as presented in [2]. This reduction was adapted to nondisjunctive programs with nonmonotonic aggregates in [5], and a similar adaption to weight constraints was presented independently in [6]. The fundamental tool in these adaptations in terms of structures is the availability of structures S_1, S_2 that allow for encoding "need to have either atom x^T or x^F, or both of them, but the latter only upon forcing the truth of both atoms." S_1, S_2 have domains $D_{S_1} = D_{S_2} = \{x^T, x^F\}$ and the following satisfaction patterns:

$$\begin{array}{llll} \emptyset \models S_1 & \{x^T\} \models S_1 & \{x^F\} \not\models S_1 & \{x^T, x^F\} \models S_1 \\ \emptyset \models S_2 & \{x^T\} \not\models S_2 & \{x^F\} \models S_2 & \{x^T, x^F\} \models S_2 \end{array}$$

A program that meets the specification is $P = \{x^T \leftarrow S_1, x^F \leftarrow S_2\}$. Indeed, $\emptyset$ is not an answer set of P as $P^\emptyset = P$ and $\emptyset \not\models P$ (so also any extension of P can never have an answer set containing neither x^T nor x^F). Both $\{x^T\}$ and $\{x^F\}$ are answer sets of P, because the reducts cancel one appropriate rule. $\{x^T, x^F\}$ is not an answer set of

P because of minimality ($P^{\{x^T,x^F\}} = P$ and $\{x^T, x^F\} \models P$, but also $\{x^T\} \models P$ and $\{x^F\} \models P$), but can become an answer set in an extension of P that forces the truth of both x^T and x^F.

A crucial observation is that S_1 and S_2 are not just nonmonotonic, but also non-convex. The main idea of our new proof is that any non-convex structure S that is closed under propositional variants can take over the role of both S_1 and S_2. For such an S, we will create appropriate variants $S\sigma^T$ and $S\sigma^F$ that use indexed copies of x^T and x^F in order to obtain the required multitudes of elements:

$$\begin{array}{lll}
\{a_1,\ldots,a_p\} \models S & \{x_1^T,\ldots,x_p^T\} \models S\sigma^T & \{x_1^F,\ldots,x_p^F\} \models S\sigma^F \\
\{a_1,\ldots,a_p,\ldots,a_q\} \not\models S & \{x_1^T,\ldots,x_q^T\} \not\models S\sigma^T & \{x_1^F,\ldots,x_q^F\} \not\models S\sigma^F \\
\{a_1,\ldots,a_p,\ldots,a_q,\ldots a_r\} \models S & \{x_1^T,\ldots,x_r^T\} \models S\sigma^T & \{x_1^F,\ldots,x_r^F\} \models S\sigma^F
\end{array}$$

We can then create a program P' acting like P by using x_q^T, x_q^F, $S\sigma^F$ and $S\sigma^T$ in place of x^T, x^F, S_1 and S_2, respectively. In addition, we need some auxiliary rules for the following purposes: to force $x_1^T,\ldots,x_p^T,x_1^F,\ldots,x_p^F$ to hold always; to require the same truth value for $x_{p+1}^T,\ldots,x_q^T$ and similar for $x_{p+1}^F,\ldots,x_q^F$; to force truth of $x_{p+1}^T,\ldots,x_r^T$ whenever any of $x_{q+1}^T,\ldots,x_r^T$ is true and to force truth of $x_{p+1}^F,\ldots,x_r^F$ whenever any of $x_{q+1}^F,\ldots,x_r^F$ is true. The resulting program can then give rise to answer sets containing either x_q^T or x_q^F, or both x_q^T, x_q^F when they are forced in an extension of the program. In particular, the answer sets of P' are the following: $\{x_1^T,\ldots,x_q^T, x_1^F,\ldots,x_p^F\}$, corresponding to $\{x^T\}$; and $\{x_1^T,\ldots,x_p^T,x_1^F,\ldots,x_q^F\}$, corresponding to $\{x^F\}$. Model $\{x_1^T,\ldots,x_r^T,x_1^F,\ldots,x_r^F\}$ instead is not an answer set of P' because of minimality, but it can be turned into an answer set by extending the program suitably. In the proof, we need to make the assumption that all symbols x_i^T and x_j^F are outside the domain D_S, which is not problematic if $\mathcal{U}$ is sufficiently large.

Theorem 1. *Let S be any non-convex structure on a set $\{a_1,\ldots,a_s\}$. Cautious reasoning over datalog^S is Π_2^P-hard.*

Proof. Deciding validity of a QBF $\Psi = \forall x_1 \cdots \forall x_m \exists y_1 \cdots \exists y_n\ E$, where E is in 3CNF, is a well-known Π_2^P-hard problem. Formula Ψ is equivalent to $\neg\Psi'$, where $\Psi' = \exists x_1 \cdots \exists x_m \forall y_1 \cdots \forall y_n\ E'$, and E' is a 3DNF equivalent to $\neg E$ and obtained by applying De Morgan's laws. To prove the claim we construct a datalog^S program P_Ψ such that $P_\Psi \models_c w$ (w a fresh atom) if and only if Ψ is valid, i.e., iff Ψ' is invalid.

Since S is a non-convex structure by assumption, there are interpretations A, B, C such that $A \subset B \subset C$, $A \models S$ and $C \models S$ but $B \not\models S$. Without loss of generality, let $A = \{a_1,\ldots,a_p\}$, $B = \{a_1,\ldots,a_q\}$ and $C = \{a_1,\ldots,a_r\}$, for $0 \leq p < q < r \leq s$. Let $E' = (l_{1,1} \wedge l_{1,2} \wedge l_{1,3}) \vee \cdots \vee (l_{k,1} \wedge l_{k,2} \wedge l_{k,3})$, for some $k \geq 1$.

Program $P_{\Psi'}$ is reported in Fig. 1, where $\sigma_i^T(a_j) = x_{i,j}^T$ and $\sigma_i^F(a_j) = x_{i,j}^F$ for all $i = 1,\ldots,m$ and $j = 1,\ldots,q$; $\theta_i^T(a_j) = y_{i,j}^T$ and $\theta_i^F(a_j) = y_{i,j}^F$ for all $i = 1,\ldots,n$ and $j = 1,\ldots,q$; $\mu(x_i) = x_{i,r}^T$ and $\mu(\neg\, x_i) = x_{i,r}^F$ for all $i = 1,\ldots,m$; $\mu(y_i) = y_{i,r}^T$ and $\mu(\neg\, y_i) = y_{i,r}^F$ for all $i = 1,\ldots,n$.

Rules (2)–(9) represent one copy of the program P' discussed earlier for each of the x_i and y_j ($i = 1,\ldots,m$; $j = 1,\ldots,n$), and so force each answer set of P_Ψ to contain at least one of $x_{i,q}^T$, $x_{i,q}^F$, and $y_{j,q}^T$, $y_{j,q}^F$, respectively, encoding an assignment

$$
\begin{array}{llll}
x^T_{i,j} \leftarrow & x^F_{i,j} \leftarrow & i \in \{1,\ldots,m\},\ j \in \{1,\ldots,p\} & (2)\\
x^T_{i,j} \leftarrow x^T_{i,k} & x^F_{i,j} \leftarrow x^F_{i,k} & i \in \{1,\ldots,m\},\ j,k \in \{p+1,\ldots,q\} & (3)\\
x^T_{i,j} \leftarrow x^T_{i,k} & x^F_{i,j} \leftarrow x^F_{i,k} & i \in \{1,\ldots,m\},\ j \in \{p+1,\ldots,r\},\ k \in \{q+1,\ldots,r\} & (4)\\
x^T_{i,q} \leftarrow S\sigma^F_i & x^F_{i,q} \leftarrow S\sigma^T_i & i \in \{1,\ldots,m\} & (5)\\
y^T_{i,j} \leftarrow & y^F_{i,j} \leftarrow & i \in \{1,\ldots,n\},\ j \in \{1,\ldots,p\} & (6)\\
y^T_{i,j} \leftarrow y^T_{i,k} & y^F_{i,j} \leftarrow y^F_{i,k} & i \in \{1,\ldots,n\},\ j,k \in \{p+1,\ldots,q\} & (7)\\
y^T_{i,j} \leftarrow y^T_{i,k} & y^F_{i,j} \leftarrow y^F_{i,k} & i \in \{1,\ldots,n\},\ j \in \{p+1,\ldots,r\},\ k \in \{q+1,\ldots,r\} & (8)\\
y^T_{i,q} \leftarrow S\theta^F_i & y^F_{i,q} \leftarrow S\theta^T_i & i \in \{1,\ldots,n\} & (9)\\
y^T_{i,j} \leftarrow sat & y^F_{i,j} \leftarrow sat & i \in \{1,\ldots,n\},\ j \in \{p+1,\ldots,r\} & (10)\\
\multicolumn{2}{l}{sat \leftarrow \mu(l_{i,1}),\mu(l_{i,2}),\ \mu(l_{i,3})} & i \in \{1,\ldots,k\} & (11)\\
a_j \leftarrow & & j \in \{1,\ldots,p\} & (12)\\
a_j \leftarrow a_k & & j,k \in \{p+1,\ldots,q\} & (13)\\
a_j \leftarrow sat & & j \in \{p+1,\ldots,q\} & (14)\\
w \leftarrow S & & & (15)
\end{array}
$$

Fig. 1. Program $P_{\Psi'}$

of the propositional variables in Ψ'. Rules (10) are used to simulate universality of the y variables, as described later. Having an assignment, rules (11) derive sat if the assignment satisfies some disjunct of E' (and hence also E' itself). Finally, rules (12)–(15) derive w if sat is false.

We first show that Ψ not valid implies $P_\Psi \not\models_c w$. If Ψ is not valid, Ψ' is valid. Hence, there is an assignment ν for $x_1,\ldots,x_m$ such that no extension to $y_1,\ldots,y_n$ satisfies E, i.e., all these extensions satisfy E'. Consider the following model of P_Ψ:

$$
\begin{aligned}
M = {} & \{x^T_{i,j} \mid \nu(x_i) = 1,\ i = 1,\ldots,m,\ j = p+1,\ldots,q\}\\
& \cup \{x^F_{i,j} \mid \nu(x_i) = 0,\ i = 1,\ldots,m,\ j = p+1,\ldots,q\}\\
& \cup \{x^T_{i,j}, x^F_{i,j} \mid i = 1,\ldots,m,\ j = 1,\ldots,p\}\\
& \cup \{y^T_{i,j}, y^F_{i,j} \mid i = 1,\ldots,n,\ j = 1,\ldots,r\}\\
& \cup \{a_j \mid j = 1,\ldots,q\} \cup \{sat\}
\end{aligned}
$$

We claim that M is a stable model of P_Ψ. Consider $I \subseteq M$ such that $I \models P^M_\Psi$. I contains all x atoms in M due to rules (2)–(5). I also contains an assignment for the y variables because of rules (6)–(9). Since any assignment for the ys satisfies at least a disjunct of E', from rules (11) we derive $sat \in I$. Hence, rules (10) force all y atoms to belong to I, and thus $I = M$ holds, which proves that M is a stable model of P_Ψ.

Now we show that $P_\Psi \not\models_c w$ implies that Ψ is not valid. To this end, let M be a stable model of P_Ψ such that $w \notin M$. Hence, by rule (15) we have that $M \not\models S$. Since $A \subseteq M$ because of rules (12), in order to have $M \not\models S$, atoms in B have to belong to M. These atoms can be supported only by rules (13)–(14), from which $sat \in M$ follows. From $sat \in M$ and rules (10), we have $y^T_{i,q}, y^F_{i,q} \in M$ for all $i = 1,\ldots,n$. And M contains either $x^T_{i,q}$ or $x^F_{i,q}$ for $i = 1,\ldots,m$ because of rules (2)–(5). Suppose by contradiction that Ψ is valid. Thus, for all assignments of $x_1,\ldots,x_m$, there is an assignment for $y_1,\ldots,y_n$ such that E is true, i.e., E' is false. Let ν be an assignment satisfying E and such that $\nu(x_i) = 1$ if $x^T_{i,q} \in M$ and $\nu(x_i) = 0$ if $x^F_{i,q} \in M$ for all $i = 1,\ldots,m$.

Consider $I = M \setminus \{sat\} \setminus \{y^T_{i,j}, y^F_{i,j} \mid i = 1,\ldots,n,\ j = q+1,\ldots,r\} \setminus \{y^T_{i,j} \mid \nu(y_i) = 0,\ i = 1,\ldots,n,\ j = p+1,\ldots,q\} \setminus \{y^F_{i,j} \mid \nu(y_i) = 1,\ i = 1,\ldots,n,\ j = p+1,\ldots,q\}$. Since ν satisfies E, ν does not satisfy E', i.e., no disjunct of E' is satisfied by ν. Hence, all rules (11) are satisfied, and thus $I \models P^M_\Psi$, contradicting the assumption that M is a stable model of P_Ψ, and so Ψ is not valid. □

4 Discussion

Our results have several consequences. First of all, from our proof it is easy to see that convex generalized atoms also form the complexity boundary for deciding whether a program has an answer set (in this case the boundary is between NP and Σ^P_2) and for checking whether an interpretation is an answer set of a program (from P to $coNP$). It also means that for programs containing only convex structures, techniques as those presented in [1] can be used for computing answer sets, while the presence of any non-convex structure requires more complex techniques such as those presented in [4]. There are several examples for convex structures that are easy to identify syntactically: count aggregates with equality guards, sum aggregates with positive summands and equality guards, dl-atoms that do not involve $\cap\!\!\!-$ and rely on a tractable Description Logic [3]. However many others are in general not convex, for example sum aggregates that involve both positive and negative summands, times aggregates that involve the factor 0, average aggregates, dl-atoms with $\cap\!\!\!-$, and so on. It is still possible to find special cases of such structures that are convex, but that requires deeper analyses.

The results also immediately imply impossibility results for rewritability: unless the polynomial hierarchy collapses to its first level, it is not possible to rewrite a program with non-convex structures into one containing only convex structures (for example, a program not containing any generalized atoms), unless disjunction or similar constructs are allowed in rule heads.

The results obtained in this work apply only to the FLP semantics. Whether the results carry over in any way to other semantics is unclear and left to future work.

References

1. Alviano, M., Calimeri, F., Faber, W., Leone, N., Perri, S.: Unfounded Sets and Well-Founded Semantics of Answer Set Programs with Aggregates. JAIR 42, 487–527 (2011)
2. Eiter, T., Gottlob, G.: On the Computational Cost of Disjunctive Logic Programming: Propositional Case. AMAI 15(3/4), 289–323 (1995)
3. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. Artif. Intell. 172(12-13), 1495–1539 (2008)
4. Faber, W.: Unfounded Sets for Disjunctive Logic Programs with Arbitrary Aggregates. In: Baral, C., Greco, G., Leone, N., Terracina, G. (eds.) LPNMR 2005. LNCS (LNAI), vol. 3662, pp. 40–52. Springer, Heidelberg (2005)
5. Faber, W., Leone, N., Pfeifer, G.: Semantics and complexity of recursive aggregates in answer set programming. AI 175(1), 278–298 (2011), special Issue: John McCarthy's Legacy
6. Ferraris, P.: Answer Sets for Propositional Theories. In: Baral, C., Greco, G., Leone, N., Terracina, G. (eds.) LPNMR 2005. LNCS (LNAI), vol. 3662, pp. 119–131. Springer, Heidelberg (2005)
7. Liu, L., Truszczyński, M.: Properties and applications of programs with monotone and convex constraints. JAIR 27, 299–334 (2006)