Minimal Extended Generalized Answer Sets and their Applications

Mauricio Osorio[1] and Claudia Zepeda[2]

[1] Universidad de las Américas, CENTIA,
Sta. Catarina Mártir, Cholula, Puebla, 72820 México
osoriomauri@gmail.com,
[2] Universidad Politécnica de Puebla,
Tercer Carril del Ejido Serrano, San Mateo Cuanala,
Municipio Juan C. Bonilla, Puebla, 72640 México
czepedac@gmail.com

Abstract When intelligent agents get new knowledge and this knowledge must be added or updated to their knowledge base, then it is important to avoid inconsistencies. In this paper, we propose a semantics for update sequences of programs. The semantics is proposed as an application of an extension of the notion of generalized answer sets. This extension is also introduced in this work.
Keywords: Logic Programming, Answer Set Programming, Updates.

1 Introduction

When intelligent agents get new knowledge and this knowledge must be added or updated to their knowledge base, then it is important to avoid inconsistencies. Currently there are several approaches in non-monotonic reasoning dealing with updates [4,1,6]. As part of the contribution of this paper, we consider sequences of programs to specify updates. We propose a semantics for update sequences of programs. Some challenging problems such as the problems presented in [4,1,6,7] are correctly solved in our approach. Due to lack of space we omit their presentation here. Additionally, we believe that the generalization of the formal properties presented in [6] also hold in this approach.

The formalism used to develop our work is Answer Set Programming (ASP) [5]. ASP is a declarative knowledge representation and logic programming language. It represents a new paradigm for logic programming. It allows us to handle problems with default knowledge and produce non-monotonic reasoning using the concept of negation as failure. Moreover, there are two popular software implementations to compute answer sets: DLV[1] and SMODELS[2]. The efficiency of such programs has increased the list of practical applications in the areas of planning, logical agents and artificial intelligence.

[1] http://www.dbai.tuwien.ac.at/proj/dlv/
[2] http://www.tcs.hut.fi/Software/smodels/

In [7] the authors propose an update semantics based on minimal generalized answer sets [2]. However the update semantics of [7] is only for pairs of programs. In this work we propose a semantics for update sequences of programs. The semantics proposed is given by an extension of the notion of generalized answer sets. This extension is also part of the contribution of this work. The notion of generalized answer sets was used in [2] to define the semantics of Abductive Logic programs. Abductive Logic programs are used to characterize the semantics of Consistency Restoring programs (CR- programs) [2]. In a CR-program there are CR-rules that are added to standard disjunctive logic programs, but they are only applied when the standard rules in the program lead to inconsistency. For further reading on CR-programs and their relation with generalized answer sets, refer to [2].

Our paper is structured as follows. In section 2 we introduce the preliminaries about general syntax of the logic programs, answer sets and generalized answer sets. In section 3 we describe the extension of the notion of generalized answer sets. In section 4 we present an application of minimal extended generalized answer sets: the semantics for update sequences of programs. Finally, in section 5 we present some conclusions and future work.

2 Preliminaries

In this paper, logic programs are understood as propositional theories. We shall use the language of propositional logic in the usual way, using propositional symbols: $p, q, \ldots,$ propositional connectives $\wedge, \vee, \rightarrow, \bot$ and auxiliary symbols: $(,)$. We assume that for any well formed propositional formula f, $\neg f$ is just an abbreviation of $f \rightarrow \bot$. In this work, we consider the two types of negation: true or explicit negation (written as $-$) and negation-as-failure (written as $\neg$). An *atom* is a propositional symbol. A *literal* is either an atom or the negation of an atom. In particular, $f \rightarrow \bot$ is called a *constraint* and it is also denoted as $\leftarrow f$. Sometimes we may use *not* instead of $\neg$ and a, b instead of $a \wedge b$, following the traditional notation of logic programming. A *regular theory* or *logic program* is just a finite set of well formed formulas or rules, it can be called just *theory* or *program* where no ambiguity arises. We shall define as a *rule* any well formed formula of the form: $f \leftarrow g$. The signature of a logic program P, denoted as $\mathcal{L}_P$, is the set of atoms that occur in P.

We want to stress the fact that in our approach, a logic program is interpreted as a propositional theory. We will restrict our discussion to propositional programs. As usual in ASP, we take for granted that programs with predicate symbols are only an abbreviation of the ground program. In this work we follow the answer sets semantics defined in terms of the so called *Gelfond-Lifschitz reduction* [5]. It is worth mentioning that in our approach an atom and its explicitly negated counterpart may never occur in the same answer set.

Finally, we say that B is a subset of A (written as $B \subseteq A$) iff every member of B is a member of A, and we say that B is a proper subset of A (written as $B \subset A$) iff $B \subseteq A$ and $B \neq A$.

2.1 Minimal generalized answer sets

In this section we recall the syntax and semantics of Abductive Logic programs as presented in [2].

Definition 1 (Abductive Logic Program). *[2] An* abductive logic program *is a pair $\langle P, A\rangle$ where P is an arbitrary program and A a set of literals, called* abductives*.*

In [2], the notion of minimal generalized answer sets is used to define the semantics of Abductive Logic programs.

Definition 2 (Generalized Answer Set). *[2] $\langle M, \Delta\rangle$ is a generalized answer set of the abductive program $\langle P, A\rangle$ iff $\Delta \subseteq A$ and M is an answer set of $P \cup \Delta$.*

In [2] is also presented an ordering between generalized answer sets in order to get the minimal generalized answer sets of an Abductive Logic program. We shall see that a minimal generalized answer set is a pair $\langle M, \Delta\rangle$, but for all practical purposes, we are only interested in M.

Definition 3 (Abductive Inclusion Order). *[2] We can establish an ordering between generalized answer sets as follows: Let $\langle M_1, \Delta_1\rangle$ and $\langle M_2, \Delta_2\rangle$ be generalized answer sets of $\langle P, A\rangle$, we define $\langle M_1, \Delta_1\rangle < \langle M_2, \Delta_2\rangle$ if $\Delta_1 \subset \Delta_2$.*

Definition 4 (Minimal Generalized Answer Set). *[2] $\langle M, \Delta\rangle$ is a minimal generalized answer set of $\langle P, A\rangle$, iff $\langle M, \Delta\rangle$ is a generalized answer set of $\langle P, A\rangle$ and it is minimal w.r.t. abductive inclusion order.*

3 Minimal extended generalized answer sets

The semantics for update sequences of programs is given by an extension of the notion of generalized answer sets. This section describes this extension. We start introducing the definition of an Extended Abductive Logic program. We shall see that this definition is similar to Definition 1, but it is added a surjective function that will be used to define an order among the extended generalized answer sets of an Extended Abductive Logic program. We also present the definition of Extended Generalized answer sets of an Extended Abductive Logic program.

Definition 5 (Extended Abductive Logic (EAL) program). *An* extended abductive logic (EAL) *program is a triple $\langle P, A, f\rangle$ such that P is an arbitrary program, A is a set of atoms, and f is some surjective function with domain A and codomain $\{1, \ldots, N\}$, $N > 0$.*

Example 1. We can define an EAL program $\langle P, A, f\rangle$ such that P is the following program:

$$\begin{aligned} a &\leftarrow \neg x_1^1. \\ b &\leftarrow a, \neg x_2^1. \\ -a &\leftarrow \neg x_1^2. \\ -b &\leftarrow \neg x_2^2. \\ c &\leftarrow . \end{aligned}$$

A is the following set of atoms: $\{x_1^1, x_2^1, x_1^2\}$; and $f : A \to \{1, 2\}$ is the function such that $f(x_j^i) = i$.

Definition 6 (Extended Generalized (EG) answer set). *Let $\langle P, A, f\rangle$ be an EAL program, M be a set of literals, and $\Delta \subseteq A$. An* extended generalized (EG) *answer set of $\langle P, A, f\rangle$ is a pair $\langle M, \Delta\rangle$ if M is an answer set of $P \cup \Delta$.*

Example 2. Let us consider the EAL program $\langle P, A, f\rangle$ of Example 1. The Table 1 shows the different EG answer sets of $\langle P, A, f\rangle$ with their respective $\Delta \subseteq A$ (the subsets that do not appear in the table do not have EG answer sets).

Δ	$\langle M, \Delta\rangle$
$\{x_1^1, x_2^1\}$	$\langle\{x_1^1, x_2^1, -a, -b, c\}, \{x_1^1, x_2^1\}\rangle$
$\{x_1^1, x_2^1, x_1^2\}$	$\langle\{x_1^1, x_2^1, x_1^2, -a, c\}, \{x_1^1, x_2^1, x_1^2\}\rangle$
$\{x_1^1, x_2^1, x_2^2\}$	$\langle\{x_1^1, x_2^1, x_2^2, -b, c\}, \{x_1^1, x_2^1, x_2^2\}\rangle$
$\{x_1^1, x_2^1, x_1^2, x_2^2\}$	$\langle\{x_1^1, x_2^1, x_1^2, x_2^2, c\}, \{x_1^1, x_2^1, x_1^2, x_2^2\}\rangle$
$\{x_1^1, x_1^2, x_2^2\}$	$\langle\{x_1^1, x_1^2, x_2^2, c\}, \{x_1^1, x_1^2, x_2^2\}\rangle$
$\{x_2^1, x_1^2, x_2^2\}$	$\langle\{x_2^1, x_1^2, x_2^2, a, c\}, \{x_2^1, x_1^2, x_2^2\}\rangle$
$\{x_1^2, x_2^2\}$	$\langle\{x_1^2, x_2^2, a, b, c\}, \{x_1^2, x_2^2\}\rangle$

Table 1. The EG answer sets of the program $\langle P, A, f\rangle$ of Example 1.

Now we present an order among the EG answer sets of an EAL program.

Definition 7 (Inclusion order among the EG answer sets). *Let $\langle P, A, f\rangle$ be an EAL program where the codomain of f is the set $\{1, \ldots, N\}$, $N > 0$. Let $A_i = \{a \in A | f(a) = i\}$. We establish an inclusion order among the EG answer sets of $\langle P, A, f\rangle$ as follows: Let $\langle M_1, \Delta_1\rangle$ and $\langle M_2, \Delta_2\rangle$ be EG answer sets of $\langle P, A, f\rangle$. We define $\langle M_1, \Delta_1\rangle \leq_{inclu} \langle M_2, \Delta_2\rangle$ iff there is k, $1 \leq k \leq N$ such that $(\Delta_1 \cap A_k) \subset (\Delta_2 \cap A_k)$, and for all j, $k < j \leq N$, $(\Delta_1 \cap A_j) = (\Delta_2 \cap A_j)$.*

Example 3. Let us consider the EAL program $\langle P, A, f\rangle$ of Example 1 and its EG answer sets in Table 1. We recall that $N = 2$ and $f : A \to \{1, 2\}$ is the function where $f(x_j^i) = i$. We can verify that $A_1 = \{x_1^1, x_2^1\}$ and $A_2 = \{x_1^2, x_2^2\}$. So, according to Table 1 and Definition 7, we can verify that
$\langle\{x_2^1, x_1^2, x_2^2, a, c\}, \{x_2^1, x_1^2, x_2^2\}\rangle \leq_{inclu} \langle\{x_1^1, x_2^1, x_1^2, x_2^2, c\}, \{x_1^1, x_2^1, x_1^2, x_2^2\}\rangle$; since there is $k = 1$ such that $(\{x_2^1, x_1^2, x_2^2\} \cap A_1) \subset (\{x_1^1, x_2^1, x_1^2, x_2^2\} \cap A_1)$, i.e, $\{x_2^1\} \subset \{x_1^1, x_2^1\}$, and for all j, $1 < j \leq 2$ we have that
$(\{x_2^1, x_1^2, x_2^2\} \cap A_2) = (\{x_1^1, x_2^1, x_1^2, x_2^2\} \cap A_2)$, i.e, $\{x_1^2, x_2^2\} = \{x_1^2, x_2^2\}$.
In a similar way we can verify that
$\langle\{x_1^1, x_2^1, -a, -b, c\}, \{x_1^1, x_2^1\}\rangle \leq_{inclu} \langle\{x_1^1, x_2^1, x_1^2, -a, c\}, \{x_1^1, x_2^1, x_1^2\}\rangle$, since there is $k = 2$ such that $(\{x_1^1, x_2^1\} \cap A_2) \subset (\{x_1^1, x_2^1, x_1^2\} \cap A_2)$, i.e., $\emptyset \subset \{x_1^2\}$, and there is no j, $2 < j \leq 2$.

Let us notice that we can also define a cardinality order among the EG answer sets of an AL program $\langle P, A, f\rangle$, denoted by $\leq_{card}$. This definition can be obtained from Definition 7 by replacing set inclusion criterion by set cardinality criterion. In the rest of this paper we will use only inclusion order among the EG answer sets and we will write $\leq$ to denote this order.

It is also worth mentioning that in Definition 7 the program P is used to get the EG answer sets, although it is not used to define the order among the EG answer sets. The order among the EG answer sets is defined in terms of the subsets of A (Δ_i and A_j). Moreover, this order can be used to get the minimal extended generalized answer sets of an EAL program $\langle P, A, f\rangle$. We shall see that a minimal extended generalized answer set is a pair $\langle M, \Delta\rangle$, but for all practical purposes, we are only interested in M.

Definition 8 (Minimal Extended Generalized (MEG) answer set). *Let* $\langle P, A, f\rangle$ *be an EAL program.* $\langle M, \Delta\rangle$ *is a* minimal extended generalized (MEG) *answer set of* $\langle P, A, f\rangle$ *if* $\langle M, \Delta\rangle$ *is an EG answer set of* P *and there is no EG answer set* $\langle M', \Delta'\rangle$ *of* $\langle P, A, f\rangle$ *such that* $\langle M', \Delta'\rangle \leq \langle M, \Delta\rangle$.

Example 4. Let us consider the EAL program $\langle P, A, f\rangle$ of Example 1. According to Table 1 and Definition 8, we can verify that $\langle\{x_1^1, x_2^1, -a, -b, c\}, \{x_1^1, x_2^1\}\rangle$ is the only MEG answer set of $\langle P, A, f\rangle$ since there is no EG answer set $\langle M', \Delta'\rangle$ of $\langle P, A, f\rangle$ such that $\langle M', \Delta'\rangle \leq \langle\{x_1^1, x_2^1, -a, -b, c\}, \{x_1^1, x_2^1\}\rangle$.

4 Updates using minimal extended generalized answer sets

In this section we present an application of minimal extended generalized answer sets. We present how the semantics for update sequences of programs is given by the extended generalized answer sets.

Formally, by an update sequence of programs, we understand a sequence $(P_1, \dots, P_n)$ of logic programs where N_{P_i} is the number of rules in each logic program. We say that $\mathbf{P}$ is an update sequence of programs over $\mathcal{L}_{\mathbf{P}}$ iff $\mathcal{L}_{\mathbf{P}}$ represents the set of atoms occurring in $\bigcup_{1\leq i\leq n} P_i$.

Definition 9 (Update program and EAL program of an update sequence). *Given an update sequence of programs* $\boldsymbol{P} = (P_1, \dots, P_n)$ *over* $\mathcal{L}_{\boldsymbol{P}}$*, we define the* update program $\boldsymbol{P}_{\oslash} = P_1 \oslash \cdots \oslash P_n$ *over* $\mathcal{L}^*_{\boldsymbol{P}}$ *(extending* $\mathcal{L}_{\boldsymbol{P}}$ *by new abducible atoms) consisting of the following items:*

1. *all constraints in* $P_1, \dots, P_{n-1}$,
2. *for each* $r_j \in P_i, 1 \leq i \leq n-1$, $1 \leq j \leq N_{P_i}$ *we add the rule* $r_j \leftarrow \neg b_j^i$, *where* b_j^i *is an abducible (a new atom),*
3. *all rules* $r \in P_n$.

An EAL program of $\boldsymbol{P}$ *is a triple* $\langle \boldsymbol{P}_{\oslash}, B, f\rangle$ *such that* B *is the set of abducibles of* $\boldsymbol{P}_{\oslash}$*, i.e.,* $B = \{b_j^i \mid b_j^i \in \boldsymbol{P}_{\oslash}, 1 \leq i \leq n-1, 1 \leq j \leq N_{P_i}\}$*; and* $f : B \to \{1, \dots, n-1\}$ *is the surjective function where* $f(b_j^i) = i$.

Example 5. Let $\mathbf{P} = (P_1, P_2, P_3)$ be an update sequence of programs over $\mathcal{L}_{\mathbf{P}} = \{a, b, c\}$ where,

P_1 :	P_2 :	P_3 :
$a \leftarrow .$	$-a \leftarrow .$	$c \leftarrow .$
$b \leftarrow a.$	$-b \leftarrow .$	

So, the update program $\mathbf{P}_\oslash = P_1 \oslash P_2 \oslash P_3$ over $\mathcal{L}^*_{\mathbf{P}}$ (extending $\mathcal{L}_{\mathbf{P}}$ by new abducible atoms $\{x^1_1, x^1_2, x^2_1, x^2_2\}$) is the following program:

$$
\begin{aligned}
a &\leftarrow \neg x^1_1.\\
b &\leftarrow a, \neg x^1_2.\\
-a &\leftarrow \neg x^2_1.\\
-b &\leftarrow \neg x^2_2.\\
c &\leftarrow .
\end{aligned}
$$

The EAL program of $\mathbf{P}$ is the triple $\langle \mathbf{P}_\oslash, B, f\rangle$, where $B = \{x^1_1, x^1_2, x^2_1, x^2_2\}$; and $f : B \rightarrow \{1, 2\}$ is the function $f(x^i_j) = i$.

Now we will see how the intended update answer sets and update answer sets of a sequence of programs $\mathbf{P}$ can be gotten from the EG answer sets and MEG answer sets of the *EAL program of* $\mathbf{P}$.

Definition 10 (Intended update answer set). *Let $\boldsymbol{P} = (P_1, \ldots, P_{n-1}, P_n)$ be an update sequence of programs over the set of atoms $\mathcal{L}_{\boldsymbol{P}}$. Let $\langle M', \Delta'\rangle$ be an EG answer set of $\langle \boldsymbol{P}_\oslash, B, f\rangle$. Then, M is an* intended update (IU) *answer set of $\boldsymbol{P}$ if only if $M = M' \cap \mathcal{L}_{\boldsymbol{P}}$.*

Example 6. Let us consider the update sequence $\mathbf{P} = (P_1, P_2, P_3)$ of Example 5. We can see that the EAL program of $\mathbf{P}$, $\langle \mathbf{P}_\oslash, B, f\rangle$ coincides with the EAL program of Example 1. So, the EG answer sets of the EAL program of $\mathbf{P}$, $\langle \mathbf{P}_\oslash, B, f\rangle$ coincide with the EG answer sets of the EAL program of Example 1 (see Table 1). Table 2 shows the EG answer sets of the EAL program $\langle \mathbf{P}_\oslash, B, f\rangle$ and their respective intended update answer sets.

Δ	$\langle M, \Delta\rangle$	IU answer set
$\{x^1_1, x^1_2\}$	$\langle\{x^1_1, x^1_2, -a, -b, c\}, \{x^1_1, x^1_2\}\rangle$	$\{-a, -b, c\}$
$\{x^1_1, x^1_2, x^2_1\}$	$\langle\{x^1_1, x^1_2, x^2_1, -a, c\}, \{x^1_1, x^1_2, x^2_1\}\rangle$	$\{-a, c\}$
$\{x^1_1, x^1_2, x^2_2\}$	$\langle\{x^1_1, x^1_2, x^2_2, -b, c\}, \{x^1_1, x^1_2, x^2_2\}\rangle$	$\{-b, c\}$
$\{x^1_1, x^1_2, x^2_1, x^2_2\}$	$\langle\{x^1_1, x^1_2, x^2_1, x^2_2, c\}, \{x^1_1, x^1_2, x^2_1, x^2_2\}\rangle$	$\{c\}$
$\{x^1_1, x^2_1, x^2_2\}$	$\langle\{x^1_1, x^2_1, x^2_2, c\}, \{x^1_1, x^2_1, x^2_2\}\rangle$	$\{c\}$
$\{x^1_2, x^2_1, x^2_2\}$	$\langle\{x^1_2, x^2_1, x^2_2, a, c\}, \{x^1_2, x^2_1, x^2_2\}\rangle$	$\{a, c\}$
$\{x^2_1, x^2_2\}$	$\langle\{x^2_1, x^2_2, a, b, c\}, \{x^2_1, x^2_2\}\rangle$	$\{a, b, c\}$

Table 2. The IU answer sets of the program $\langle \mathbf{P}_\oslash, B, f\rangle$.

Definition 11 (Update answer set). *Let $\boldsymbol{P} = (P_1, \ldots, P_{n-1}, P_n)$ be an update sequence of programs over the set of atoms $\mathcal{L}_{\boldsymbol{P}}$. Let $\langle M', \Delta' \rangle$ be a MEG answer set of the EAL program $\langle \boldsymbol{P}_{\oslash}, B, f \rangle$. Then, M is an* update answer set *of $\boldsymbol{P}$ if only if $M = M' \cap \mathcal{L}_{\boldsymbol{P}}$.*

Example 7. Let us consider the update sequence $\mathbf{P} = (P_1, P_2, P_3)$ of Example 5. In Example 4 we verified that $\langle \{x_1^1, x_2^1, -a, -b, c\}, \{x_1^1, x_2^1\} \rangle$ is the only MEG answer set of $\langle \mathbf{P}_{\oslash}, B, f \rangle$. So, according to Definition 11, we can verify that $\{-b, -a, c\}$ is the only one update answer set of $\mathbf{P}$.

Now, let us consider Example 1 from [6] as another example to illustrate Definition 11.

Example 8. Let us consider the update sequence $\mathbf{P} = (P_1, P_2)$ where

P_1 :	P_2 :
$sleep \leftarrow night, \neg watchTv, \neg other.$	$-tvOn \leftarrow powerFailure.$
$night \leftarrow .$	$-tvOn \leftarrow assignmentDue, working.$
$tvOn \leftarrow \neg tvBroke.$	$assignmentDue \leftarrow .$
$watchTv \leftarrow tvOn.$	$working \leftarrow .$
	$other \leftarrow working.$

We can verify that the only MEG answer set of the EAL program $\langle \mathbf{P}_{\oslash}, B, f \rangle$ is $\langle \{x_3^1, night, other, assignmentDue, working, -tvOn\}, \{x_3^1\} \rangle$. The only update answer set of $\mathbf{P}$ which coincides with the result of Example 1 in [6] is: $\{night, other, assignmentDue, working, -tvOn\}$. Bellow we can see the $\mathbf{P}_{\oslash} = P_1 \oslash P_2$ over $\mathcal{L}^*_{\mathbf{P}}$.

$$
\begin{aligned}
&sleep \leftarrow night, \neg watchTv, \neg other, \neg x_1^1.\\
&night \leftarrow, \neg x_2^1.\\
&tvOn \leftarrow \neg tvBroke, \neg x_3^1.\\
&watchTv \leftarrow tvOn, \neg x_4^1.\\
&-tvOn \leftarrow powerFailure.\\
&-tvOn \leftarrow assignmentDue, working.\\
&assignmentDue \leftarrow .\\
&working \leftarrow .\\
&other \leftarrow working.
\end{aligned}
$$

5 Conclusions and future work

We presented a semantics for update sequences of programs given by an extension of the notion of generalized answer sets called Extended Generalized answer sets. Some challenging problems such as the problems presented in[4,1,6,7] are correctly solved in our approach. Due to lack of space we omit their presentation here. In fact, the work presented in this paper is an initial proposal about the semantics for update sequences of programs. In future work we plan to show how the formal properties for update programs presented in [6] can be generalized easily for update sequences of programs.

We are also studying the possibility of using EG answer sets to represent a simple kind of preferences. In the approach that we are studying, we define a particular EA program $\langle P, B, f \rangle$ where its MEG answer sets correspond to the preferred answer sets. The program P of this particular EA program should include a set of constraints. Each constraint is related to one possible solution of the problem and a new and different atom x_j^i, where i should indicate the satisfaction degree of the corresponding solution. For instance, let us consider an Example from [3] about the decision over possible desserts. In that example is indicated a preference for ice-cream over cake and a preference for coffee over tea. Additionally, it is not possible to have coffee with ice cream. We can see that this problem have three possible solutions: ice-cream with tea, coffee with cake, and tea with cake. Moreover, the preferred solutions are ice-cream with tea and coffee with cake since they have at least one of the most preferred options. So, we could have the EA program $\langle P, A, f \rangle$ where P is the following program:

$$\begin{array}{l} iceCream \lor cake \leftarrow \\ coffee \lor tea \leftarrow \\ \leftarrow coffee, iceCream \\ \leftarrow iceCream, tea, \neg x_1^1 \\ \leftarrow coffee, cake, \neg x_2^1 \\ \leftarrow tea, cake, \neg x_1^2 \end{array}$$

A is the following set of atoms: $\{x_1^1, x_2^1, x_1^2\}$; and $f : A \to \{1, 2\}$ is the function $f(x_j^i) = i$. We can verify that their MEG answer sets correspond to the preferred answer sets $\{iceCream, tea\}$ and $\{coffee, cake\}$.

References

1. J. Alferes and L. Pereira. Logic programming updating: a guided approach, 2002.
2. M. Balduccini and M. Gelfond. Logic Programs with Consistency-Restoring Rules. In P. Doherty, J. McCarthy, and M.-A. Williams, editors, *International Symposium on Logical Formalization of Commonsense Reasoning*, AAAI 2003 Spring Symposium Series, Mar 2003.
3. G. Brewka. Logic Programming with Ordered Disjunction. In *Proceedings of the 18th National Conference on Artificial Intelligence, AAAI-2002.* Morgan Kaufmann, 2002.
4. T. Eiter, M. Fink, G. Sabbatini, and H. Tompits. On properties of update sequences based on causal rejection. *Theory and Practice of Logic Programming*, 2(6):711–767, 2002.
5. M. Gelfond and V. Lifschitz. The Stable Model Semantics for Logic Programming. In R. Kowalski and K. Bowen, editors, *5th Conference on Logic Programming*, pages 1070–1080. MIT Press, 1988.
6. M. Osorio and V. Cuevas. Updates in answer set programming: An approach based on basic structural properties. *Accepted in Theory and Practice of Logic Programming*, 2006.
7. F. Zacarias, M. O. Galindo, J. C. A. Guadarrama, and J. Dix. Updates in Answer Set Programming based on structural properties. In *Proceedings of the 7th International Symposium on Logical Formalizations of Commonsense Reasoning. Dresden University Technical Report (ISSN 1430-211X)*, 2005.