Part II

Contribution

Introduction

We recall that the objective of our work is to investigate and evaluate the capabilities of Answer Set Programming to represent disaster situations to give support in defining evacuation plans. Then we started analyzing how geographic information about the disaster zone can be translated into a format that Answer Sets is capable to understand. We continued studying and applying different Answer Set approaches that are useful to define the evacuation plans, such as, Answer Set Planning, CR-rules, Ordered Disjunction programs, Language $\mathcal{PP}$ for planning preferences, and Minimal Generalized Answer Sets.

In the Contribution part of this work we show how and why the Answer Set approaches mentioned above were applied to define evacuation plans and we also present some lacks of them that this work want to address.

First at all, we have to mention that we start this part proposing a procedure to construct the hazard zone background knowledge from geographic information. This procedure is result of our own experience dealing with this kind of data. It is worth to mention that in this work we tested all of our results with a short part of the real geographic data about Volcano Popocatépetl hazard zone, since the total real data of this zone were obtained almost at the end of this work.

In order to obtain the evacuation plans, we started modeling the evacuation planning problem using Answer Set Planning. As we know, normally in a zone of risk there is a set of pre-defined evacuation routes. However, in a real case is possible that part of the pre-defined evacuation routes become blocked, then generation of alternative evacuation plans is necessary. Then, the model that we propose in this work considers this possible scenario.

At the beginning, we tried to specify this problem using only Answer Set Planning where a set of actions were defined in order to travel by the different possible paths in the risk zone. However after different intents, we realized that using only Answer Set Planning to specify this problem does not result in a natural way. Hence we proposed to extend our model of evacuation plans adding CR-rules. The idea is to obtain the alternative evacuation plans taking advantage of the definition of a CR-rule and use it only if there is no way to obtain an evacuation plan when part of the pre-defined evacuation route is blocked. Additionally, since the semantics of a CR-program is defined in terms of the minimal generalized answer sets, we propose a characterization of them in terms of ordered disjunction programs. This result proves that both abductive programs and programs with CR-rules can be properly represented using ordered disjunction.

Using CR-rules makes it possible to obtain the alternative evacuation plans but the problem now is that these alternative evacuation plans do not consider any other characteristic of the path that they follow. Then, we realized that we needed to apply the concept of preference to obtain an appropriate evacuation plan. For instance, we would like to express that it is preferred that in an alternative evacuation route buses travel only by roads belonging to some evacuation route and it is not important if they travel or not by its assigned evacuation route. Hence, we propose to use language $\mathcal{PP}$ in order to express preferences at different levels over the alternative plans and because it allows us to express preferences over plans where the satisfaction of these preferences depends on time and on their temporal relationships.

While we used $\mathcal{PP}$ to express preferences we realized that there are some preferences that cannot be expressed in a simple and natural way since they result very large. Then, in order to have a natural representation of these kind of preferences we define $\mathcal{PP}^{par}$ language. $\mathcal{PP}^{par}$ is an extension of $\mathcal{PP}$ language where propositional connectives and temporal connectives allow us to represent compactly preferences having a particular property. Since we consider that language $\mathcal{PP}$ could take advantage of the working framework of propositional Linear Temporal Logic LTL to express preferences, we also present a brief overview about the relationship between language $\mathcal{PP}$ and LTL.

We also studied ordered disjunction programs since they can be used to express preferences. We proposed an extension of them to a wider class of logic programs that we called *extended ordered disjunction programs*. Moreover, we shown that in particular extended ordered rules with negated negative literals could be useful to allow a simpler and easier encoding to obtain the preferred plans with respect to a preference expressed in language $\mathcal{PP}$. We also considered them to obtain the maximal answer sets of a program characterizing an Argumentation Framework such that these maximal answer sets correspond to the preferred extensions of it.

Finally, we introduce the notion of *Semantic Contents of a program* as an alternative point of view to obtain different answer set semantics of a program. In particular, we show how to obtain the *standard answer sets*, the *generalized answer sets*, the *minimal generalized answer sets* and a new answer set semantic introduced in this section called *partial answer sets*. In particular we present an example in a planning domain where partial answer sets could be useful.

Chapter 6

Extended ordered disjunction programs

As we have described in Section 4.8, Brewka introduced in [9] the connective $\times$, called *ordered disjunction*, to express default knowledge with knowledge about preferences in a simple and elegant way. While the disjunctive clause $a \vee b$ is satisfied equally by either a or b, to satisfy the ordered disjunctive clause $a \times b$, a will be preferred to b, i.e. a model containing a will have a better *satisfaction degree* than a model that contains b but does not contain a. For example, the natural language statement *"I prefer travel by bus to walk"* can be expressed as $travelBus \times walk$ and a model containing $travelBus$ will be preferred to a model that contains $walk$.

In this chapter we introduce *extended ordered disjunction programs*. The definition presented here extends ordered disjunction programs to a wider class of logic programs[1] [35].

We saw (in Section 4.7) that the semantics of a CR-program is defined in terms of the minimal generalized answer sets of a particular abductive logic program. This abductive logic program is based on the original program and a set of abducibles which

[1] Moreover, while the extension introduced in [35] is in the context of Answer Sets, the extension introduced in [8] for the operator $\times$ is in a different context.

corresponds to a subset of the signature of this original program. In this chapter, we also propose a characterization of minimal generalized answer sets in terms of ordered disjunction programs.

In this chapter we also propose to specify a preference ordering among the answer sets of a program with respect to an ordered list of atoms using a particular set of extended ordered disjunction programs [46]. We propose to use double default negation in each atom of the ordered rule that represents the mentioned list of atoms. We also mention a second application of extended ordered disjunction programs with double default negation to obtain the maximal answer set of a program. Finally, we show how to compute the preferred answer sets for extended ordered programs using PSMODELS.

6.1 Definition of extended ordered disjunction programs

In [7] the head of ordered disjunction rules is defined in terms of ground literals, now in this chapter, the head and the body of extended ordered disjunction rules are defined in terms of well formed propositional formulas. We think that a broader syntax for rules could give us some benefits. For example, the use of nested expressions could simplify the task of writing logic programs and improve their readability since, it could allows us to write more concise rules and in a more natural way. Another example is presented in [22] where the authors discuss that for some knowledge representation problems, an extension to the answer set semantics is needed. They introduce *parametric connectives* and present the solution to some problems using this extension. Another alternative that has been recognized lately as an extension that allows more natural problem solving is the use of implication in the body of rules, since it would provide a more uniform and natural encoding. The need of it arises both while modeling real applications as well as for research purposes. In [31] an application is shown where embedded implication gives a natural representation to a real problem. Hence, we believe that it is worth it to present a generalization of ordered disjunctions to more general theories. We present a simple example that uses an extended ordered disjunction program in order to illustrate how modeling a problem using a broader syntaxis results more natural, direct and intuitive: I prefer *travel by airplane* to either *travel by bus* other *travel by train*, but between *travel by bus* and *travel by train* I don't have any particular preference. Then, using an extended ordered program we could just write $travelByAirplan \times (travelByBus \vee travelByTrain)$.

Definition 6.1. An *extended ordered disjunction rule* is either a well formed propositional formula as defined in Section 4.6, or a formula of the form:

$$f_1 \times \ldots \times f_n \leftarrow g \tag{6.1}$$

where $f_1, \ldots, f_n, g$ are well formed propositional formulas. An *extended ordered disjunction program* is a finite set of extended ordered disjunction rules. □

The formulas $f_1 \ldots f_n$ are usually called the choices of a rule and their intuitive reading is as follows: if the body is true and f_1 is possible, then f_1; if f_1 is not possible, then f_2; $\ldots$; if none of $f_1, \ldots, f_{n-1}$ is possible then f_n.

The particular case where all f_i are literals and g is a conjunction of literals corresponds to the original ordered disjunction programs as presented by Brewka in [7], and as we indicated before we call them *standard ordered disjunction programs.* We recall that Brewka's ordered disjunction programs use the strong negation connective. Here we will consider only one type of negation (default negation) but this does not affect the results given in [7]. If additionally $n = 0$ the rule is a constraint, i.e, $\perp \leftarrow g$.

If $n = 1$ it is an extended rule and if $g = \top$ the rule is a fact and can be written as $f_1 \times \ldots \times f_n$.

An *extended ordered disjunction program* and a *standard ordered disjunction program* can be called just *extended ordered program* and *standard ordered program* respectively where no ambiguity arises.

Example 6.1. A person should travel from his/her home to school in winter. This person prefers to travel by bus and to drink tea inside the bus rather than travel by bicycle. Additionally, he/she prefers to travel by bicycle rather than walk. This person also should consider that part of the path from his/her home to school can become blocked by snow in winter. Then, we can model this situation considering the following extended ordered disjunction program P:

$$winter.$$

$$(travelBus \wedge drinkTea) \times travelBicycle \times walk \leftarrow winter, \neg pathBlocked.$$

□

Now, we present the semantics of extended ordered disjunction programs. Most of the definitions presented here are taken from [7, 9]. The relevant difference is the satisfaction degree. The reader may see that the satisfaction degree as defined here is just a straightforward generalization of Brewka's definition [7, 9], according to our notation and Theorem 4.1 (see Section 4.2). Hence, standard ordered programs are special cases of extended ordered programs, thus all results hold for this restricted class as well.

Definition 6.2. [7]Let $r := f_1 \times \ldots \times f_n \leftarrow g$ be an extended ordered rule. For $1 \leq k \leq n$ the *k-th option of* r is defined as follows:

$$r^k := f_k \leftarrow g, \neg f_1, \ldots \neg f_{k-1}$$

□

Definition 6.3. [7] Let P be an extended ordered program. P' is a *split program of* P if it is obtained by replacing each rule $r := f_1 \times \ldots \times f_n \leftarrow g$ in P by one of its options $r^1, \ldots, r^k$. □

Definition 6.4. [9] Let P be an extended ordered program. M is an answer set of P iff it is an answer set[2] of a split program P' of P. □

Definition 6.5. [7]Let M be an answer set of an extended ordered program P and let $r := f_1 \times \ldots \times f_n \leftarrow g$ be a rule of P. We define the *satisfaction degree of r with respect to M*, denoted by $deg_M(r)$, as follows:

- if $M \cup \neg(\mathcal{L}_P \setminus M) \not\vdash_{\mathrm{I}} g$, then $deg_M(r) = 1$.
- if $M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{\mathrm{I}} g$ then $deg_M(r) = min_{1 \leq i \leq n} \{i \mid M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{\mathrm{I}} f_i\}$. □

Example 6.2. If we consider the extended ordered program P of Example 6.1 then we

[2] Note that since we are not considering strong negation, there is no possibility of having inconsistent answer sets.

can obtain the following split programs from it:

P' :

$winter.$

$(travelBus \wedge drinkTea) \leftarrow winter, \neg pathBlocked.$

P'' :

$winter.$

$travelBicycle \leftarrow winter, \neg pathBlocked, \neg(travelBus \wedge drinkTea).$

P''' :

$winter.$

$walk \leftarrow winter, \neg pathBlocked, \neg(travelBus \wedge drinkTea), \neg travelBicycle.$

Moreover, $M_1 = \{winter, (travelBus \wedge drinkTea)\}$, $M_2 = \{winter, travelBicycle\}$ and $M_3 = \{winter, walk\}$ are answer sets of P since they are the answer sets of each split program respectively.

Finally, if r_1 denotes the rule $winter$ of program P and r_2 denotes the rule $(travelBus \wedge drinkTea) \times travelBicycle \times walk \leftarrow winter, \neg pathBlocked$ of program P, then the satisfaction degrees of each rule according to answer sets M_1, M_2 and M_3 are the following:

$$deg_{M_1}(r_1) = 1, \qquad deg_{M_1}(r_2) = 1.$$

$$deg_{M_2}(r_1) = 1, \qquad deg_{M_2}(r_2) = 2.$$

$$deg_{M_3}(r_1) = 1, \qquad deg_{M_3}(r_2) = 3.$$

□

The following theorem states the relationship between the answer sets of an extended ordered program and the satisfaction degree of each rule in the program.

Theorem 6.1. *[7] Let P be an extended ordered program. If M is an answer set of P then M satisfies all the rules in P to some degree.*

Proof. The key issue is to observe that given a model M then $\models_M f$ iff $M \cup \neg(\mathcal{L}_P \setminus M) \vdash_I f$, since $M \cup \neg(\mathcal{L}_P \setminus M)$ is complete then $M \cup \neg(\mathcal{L}_P \setminus M)$ defines a theory that proves f (See [32]). The rest of the proof follows from the proof in [9], generalized according to Definition 6.5. □

Definition 6.6. [9] Let P be an extended ordered program and M a set of literals. We define:

$$S^i_M(P) = \{r \in P \mid deg_M(r) = i\} \tag{6.2}$$

□

Example 6.3. If we consider the extended ordered program P from Example 6.1 and the satisfaction degrees of rules r_1 and r_2 according to answer sets M_1, M_2 and M_3 from Example 6.2 then we can obtain the following:

$$S^1_{M_1}(P) = \{r_1, r_2\}. \qquad S^2_{M_1}(P) = \{\}. \qquad S^3_{M_1}(P) = \{\}$$

$$S^1_{M_2}(P) = \{r_1\}. \qquad S^2_{M_2}(P) = \{r_2\}. \qquad S^3_{M_2}(P) = \{\}$$

$$S^1_{M_3}(P) = \{r_1\}. \qquad S^2_{M_3}(P) = \{\}. \qquad S^3_{M_3}(P) = \{r_2\}$$

□

With the definitions presented, we now introduce two different types of preference relations among the answer sets of an extended ordered program: inclusion based preference and cardinality based preference.

Definition 6.7. [9] Let M and N be answer sets of an extended ordered program P. Then we say that M is *inclusion preferred* to N, denoted as $M >_i N$, iff there is an i such that $S^i_N(P) \subset S^i_M(P)$ and for all $j < i$, $S^j_M(P) = S^j_N(P)$. □

Example 6.4. If we consider the extended ordered program P from Example 6.1 and the results from Example 6.3, then we can verify that M_1 is inclusion preferred to M_2 and to M_3, i.e., $M_1 >_i M_2$ and $M_1 >_i M_3$ since $S^1_{M_2}(P) \subset S^1_{M_1}(P)$ and $S^1_{M_3}(P) \subset S^1_{M_1}(P)$.

Definition 6.8. [9] Let M and N be answer sets of an extended ordered program P. Then we say that M is *cardinality preferred* to N, denoted as $M >_c N$, iff there is an i such that $|S^i_M(P)| > |S^i_N(P)|$ and for all $j < i$, $\left|S^j_M(P)\right| = \left|S^j_N(P)\right|$. □

Example 6.5. If we consider the extended ordered program P from Example 6.1 and the results from Example 6.3, then we can verify that M_2 is cardinality preferred to M_3, i.e., $M_2 >_c M_3$ since $|S^2_{M_2}(P)| > |S^2_{M_3}(P)|$ and $|S^1_{M_2}(P)| = |S^1_{M_3}(P)|$.
We also can verify that M_1 is cardinality preferred to M_2 and to M_3, i.e., $M_1 >_c M_2$ and $M_1 >_c M_3$ since $|S^1_{M_1}(P)| > |S^1_{M_2}(P)|$ and $|S^1_{M_1}(P)| > |S^1_{M_3}(P)|$. □

Definition 6.9. [9] Let M be an answer set of an extended ordered program P. M is an *inclusion preferred answer set* of P if there is no answer set M' of P, $M \neq M'$, such that $M' >_i M$. □

Definition 6.10. [9] Let M be an answer set of an extended ordered program P. M is a *cardinality preferred answer set* of P if there is no M' answer set of P, $M \neq M'$, such that $M' >_c M$. □

Example 6.6. If we consider the extended ordered program P from Example 6.1, the results from Example 6.3, Example 6.4 and Example 6.5 then we can verify that M_1 is the inclusion preferred answer set of P and the cardinality preferred answer set of P. □

6.2 Translation from an abductive logic program to a standard ordered disjunction program

In Section 4.7 we reviewed CR-programs. We saw that the semantics of a CR-program is defined in terms of the *minimal generalized answer sets* of a particular abductive logic program. This abductive logic program is based on the original program and a set of abducibles which corresponds to a subset of the signature of this original program.

In this section, we propose a characterization of minimal generalized answer sets in terms of ordered disjunction programs. This theoretical result proves that both abductive programs and programs with CR-rules can be properly represented using ordered disjunction.

Definition 6.11. Let $< P, A >$ be an abductive logic program. Then we define a translation into a standard ordered program, denoted by $ord(P, A)$, as follows: First, for any literal $a \in A$ the clause r_a is defined as $a^\bullet \times a$, where $a^\bullet$ is a literal that does not occur in the original program. We define $ord(P, A) = P \cup \{r_a \mid a \in A\}$. □

Lemma 6.1. *$M \cap \mathcal{L}_P$ is a generalized answer set of the abductive logic program $< P, A >$ iff M is an answer set of $ord(P, A)$.*

Sketch. Let $\mathcal{P}(A)$ be all subsets of A and $R_{\mathcal{P}(A)} := \{r_l^1 \mid l \in A \setminus \mathcal{P}(A)\} \cup \{r_l^2 \mid l \in \mathcal{P}(A)\}$. Let us recall that $r_l = l^\bullet \times l$ so that $r_l^1 = l^\bullet$ and $r_l^2 = l \leftarrow \neg l^\bullet$. The proof follows from the fact that all $P \cup R_{\mathcal{P}(A)}$ correspond exactly to the split programs of $ord(P, A)$. □

The next theorem proves the validity of our ordered translation for the semantics of abductive logic programs when it uses set inclusion.

Theorem 6.2. *Let $< P, A >$ be an abductive program and M a set of atoms. $M \cap \mathcal{L}_P$ is a minimal generalized answer set of $< P, A >$ iff M is an inclusion preferred answer set of $ord(P, A)$.*

Sketch. It is only needed to prove that $M \cap \mathcal{L}_P$ is minimal w.r.t. abductive inclusion/cardinality order iff M is minimal w.r.t. inclusion/cardinality preferred order. The proof is straightforward using the fact that for all rules in $ord(P, A)$, which are in the form rules in Definition 6.1, $n \leq 2$ implies that $g = \top$. $\square$

Example 6.7. This example shows how our ordered translation works. It is based on the CR-program P_1' from Example 4.11 of Section 4.7. In particular we consider the abductive logic program $\langle P_1, A\rangle$ used to obtain the semantics of the CR-program P_1', where P_1 is the following program:

$$p \leftarrow \neg q.$$
$$r \leftarrow \neg s.$$
$$q \leftarrow t.$$
$$s \leftarrow t.$$
$$\leftarrow p, r.$$

and $A = \{q, s, t\}$. According to the translation defined in 6.11 we have that $ord(P, A)$ is the following program:

$$p \leftarrow \neg q.$$
$$r \leftarrow \neg s.$$
$$q \leftarrow t.$$
$$s \leftarrow t.$$
$$\leftarrow p, r.$$
$$q^{\bullet} \times q$$
$$s^{\bullet} \times s$$
$$t^{\bullet} \times t$$

Moreover we can verify that the inclusion preferred answer sets o f $ord(P, A)$ are $M_1 = \{q^{\bullet}, q, r\}$, $M_2 = \{s^{\bullet}, s, p\}$ and $M_3 = \{q^{\bullet}, s^{\bullet}, t^{\bullet}, t, q, s\}$. Hence, the *minimal* *generalized answer sets* of $\langle P_1, A\rangle$ are $M_1 = \{q, r\}$, $M_2 = \{s, p\}$ and $M_3 = \{t, q, s\}$ since $\mathcal{L}_{P_1} = \{p, q, r, s, t\}$ as in Example 4.11. □

6.3 Preferences in terms of extended ordered programs with double negation

Preferences are useful when the space of feasible solutions of a given problem is dense but not all these solutions are equivalent w.r.t. some additional requirements. In this case, the goal is to find feasible solutions that most satisfy these additional requirements. In [9] Brewka introduced *logic programs with ordered disjunction (LPODs)* where the connective $\times$, called *ordered disjunction*, allows to express default knowledge with knowledge about preferences in a simple and elegant way. However, if we only want to *specify a preference ordering among the answer sets of a program with respect to an ordered set of atoms* by means of adding a standard ordered disjunction rule to the original program then, ordered disjunction as defined by Brewka does not work since it corresponds to a disjunction where an ordering is defined. For instance, if P is the following program

$$a \leftarrow .$$
$$b \leftarrow \neg c.$$
$$c \leftarrow \neg b.$$
$$d \leftarrow \neg a.$$
$$f \leftarrow c, \neg a.$$
$$e \leftarrow b, \neg a.$$

then the answer sets of this program are $\{a, b\}$ and $\{a, c\}$.

Now, if we want to specify a preference ordering among the answer sets of a program with respect to the ordered set of atoms $[f, c]$ then we could consider to add to the program P the standard ordered disjunction rule $\{f \times c\}$ that stands for "if f is possible then f otherwise c" (see [7]). Then, thinking in a preference sense, the intuition indicates that the inclusion-preferred answer sets of $P \cup \{f \times c\}$ should be $\{a, c\}$. However, we obtain two inclusion-preferred answer sets $\{a, b, f\}$ and $\{a, c, f\}$.

Therefore, in order to specify a preference ordering among the answer sets of a program with respect to an ordered set of atoms, in this section we propose to use a particular set of extended ordered disjunction programs [46]. Specifically, we propose to add to the original program an extended ordered rule such that this rule is defined using the ordered set of atoms and each atom has *double default negation*.

We are going to explain way we propose to use atoms with double default negation. Let us consider $\neg\neg a$ where a is an atom. Since $\neg\neg a$ is equivalent to the restriction $\leftarrow \neg a$, the intuition behind $\neg\neg a$ is to indicate that it is desirable that a holds in the model of a program. Formally, as we defined in Background Section, an atom with double negation corresponds to a *negated negative literal* where the only negation used is *default negation*.

Hence, if we consider again the previous program P and we want to to specify a preference ordering among the answer sets of P with respect to an ordered set of atoms $[f, c]$ we have to do the following: First we define the extended ordered disjunction rule using the ordered set of atoms $[f, c]$ such that each atom has *double default negation*, i.e, we define $\{\neg\neg f \times \neg\neg c\}$. Then, we add this extended ordered disjunction rule to the original program P, i.e., we define the following extended ordered disjunction program: $P \cup \{\neg\neg f \times \neg\neg c\}$. As we will prove in the following subsection, we obtain the desired inclusion-preferred answer set $\{a, c\}$ from this extended ordered disjunction program.

We explained that the intuition behind an extended ordered rule using negated negative literals is to indicate that we want to specify a preference ordering among the answer sets of a program with respect to an ordered set of atoms. However, in case that the answer sets of the program do not contain any of the atoms in the given ordered set of atoms then the extended ordered rule must allow to obtain all the answer sets of the program. In order to obtain all the answer sets of the program we propose to add a positive literal to the end of the extended ordered rule that we defined as we explained above. This positive literal must be an atom that does not occur in the original program, for instance the atom all_pref. For instance,let us consider again the program P that has the answer sets $\{a, b\}$ and $\{a, c\}$. If we want to specify a preference ordering among the answer sets of this program with respect to the ordered set of atoms $C = [f, e]$ then we define the extended ordered rule as we explained above but we add the atom all_pref at the end of the rule, i.e., $\neg\neg f \times \neg\neg e \times all\_pref$. The new extended ordered rule indicates that we prefer the answer sets where f holds to the answer sets where e holds and if there is no answer sets of the program where f or e holds then all answer sets are preferable. It is worth to mentioning that in case that there is no answer sets of the program where f or e holds then all the answer sets will contain the atom all_pref. Hence, the extended ordered program is as follows:

$$a \leftarrow .$$
$$b \leftarrow \neg c.$$
$$c \leftarrow \neg b.$$
$$d \leftarrow \neg a.$$
$$f \leftarrow c, \neg a.$$
$$e \leftarrow b, \neg a.$$
$$\neg\neg f \times \neg\neg e \times all\_pref.$$

As we expected, we obtain the inclusion-preferred answer sets: $\{a, c, all\_pref\}$ and $\{a, b, all\_pref\}$ since there is no answer sets of the program containing f or e.

The following short examples also show the role of negated negative literals in an extended ordered program.

Example 6.8. If consider the following standard ordered program

$$a \times b$$

we can verify that the inclusion preferred answer set of it is $\{a\}$ since ordered disjunction corresponds to a disjunction where an ordering is defined.

However, the following extended ordered program

$$\neg\neg a \times \neg\neg b$$

has no answer sets since the extended ordered rule only indicates that we prefer the answer sets containing a to the answer sets containing b but there is no answer sets. □

Example 6.9. We also can consider the following standard ordered program,

$$a \times b.$$
$$b \leftarrow \neg a$$

we can verify that the inclusion preferred answer set of it is $\{a\}$ since ordered disjunction corresponds to a disjunction where an ordering is defined.

However, the following extended ordered program

$$\neg\neg a \times \neg\neg b$$
$$b \leftarrow \neg a$$

only has $\{b\}$ as its inclusion preferred answer set, since the extended ordered rule only indicates that we prefer the answer sets containing a to the answer sets containing b and program $b \leftarrow \neg a$ has only the answer set $\{b\}$. □

In order to formalize our previous discussion about the specification of an ordering among the answer sets of a normal program with respect to an ordered set of atoms using an extended ordered program, we are going to introduce Lemma 6.7. This lemma allows us to obtain the most preferred answer set of a normal program with respect to an ordered set of formulas in terms of a particular extended ordered program. This particular extended ordered program results from joining the original normal program and a particular ordered disjunction rule that is defined in terms of the ordered set of atoms. It is worth mentioning that the proof of Lemma 6.7 uses other auxiliary lemmas that we are going to present before we present Lemma 6.7. Additionally, we have to mention that most of these auxiliary lemmas are for general purpose.

We start introducing a definition and a proposition that allows us to define the most preferred answer set with respect to a program and an ordered set of formulas.

Definition 6.12. Let P be a normal program and let M and N be two answer sets of P. Let $C = [c_1, c_2, \ldots, c_n]$ be an ordered set of formulas. The answer set M *is preferred to the answer set* N *with respect to* $C \cup P$ (denoted as $M <_{C \cup P} N$) if

1. there exists $i = min(1 \leq k \leq n)$ such that $M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{\mathrm{I}} c_i$ and $N \cup \neg(\mathcal{L}_P \setminus N) \nvdash_{\mathrm{I}} c_i$, and
2. for all $j < i$, $M \cup \neg(\mathcal{L}_P \setminus M) \vdash_{\mathrm{I}} c_j$ and $N \cup \neg(\mathcal{L}_P \setminus N) \vdash_{\mathrm{I}} c_j$ or $M \cup \neg(\mathcal{L}_P \setminus M) \nvdash_{\mathrm{I}} c_j$ and $N \cup \neg(\mathcal{L}_P \setminus N) \nvdash_{\mathrm{I}} c_j$. □

Proposition 6.1. *Let C be an ordered set of formulas; then $<_{C \cup P}$ is a partial order.* □

Given an ordered list of formulas C, an answer set M of a normal program P *is most preferred with respect to* $C \cup P$ if there is no other answer set N of P that is preferred to M with respect to $C \cup P$.

Example 6.10. Let $C = [b, c]$ be an ordered list of atoms. Let P be the following normal program:

$$a \leftarrow .$$

$$c \leftarrow \neg b.$$

$$b \leftarrow \neg c.$$

We can verify that P has two answer sets, $\{a, b\}$ and $\{a, c\}$. We also can verify that the answer set $\{a, b\}$ *is preferred to the answer set* $\{a, c\}$ *with respect to* C, i.e., $\{a, b\} <_{C \cup P} \{a, c\}$ and that $\{a, b\}$ is also *the most preferred answer set.* □

Now, we define the extended ordered rule that we shall join to the original normal program in order to obtain an extended ordered program that we shall use to obtain the most preferred answer sets of the normal program with respect to the ordered set of atoms C in terms.

Definition 6.13. Let P be a normal program and $C = [c_1, c_2, \ldots, c_n]$ be an ordered set of atoms such that $C \subseteq \mathcal{L}_P$. We define an *extended ordered rule defined from* C, denoted as r_C, as follows: $r_C := \neg\neg c_1 \times \neg\neg c_2 \times \ldots \times \neg\neg c_n \times all\_pref$ such that $all\_pref \notin \mathcal{L}_P$. □

Example 6.11. Let us consider again program P from Example 6.10, and also the ordered set of atoms $C = [b, c]$. Then, the *extended ordered rule defined from* C, denoted as r_C is the following: $r_C := \neg\neg b \times \neg\neg c \times all\_pref$. □

The following lemmas are the auxiliary lemmas that we shall use to prove Lemma 6.7. We shall see that some of these auxiliary lemmas are for general purpose, then we could used them not only to prove Lemma 6.7.

The following lemma indicates that given two normal programs and an extended ordered rule, if we add this extended ordered rule to both programs and we assume that the answer sets of one of them are subsets of the answers sets of the other program then the preferred answer sets of one of them are also subsets of the preferred answer sets of the other.

Lemma 6.2. *Let P_1 and P_2 be two normal programs such that $\mathcal{L}_{P_2} \subseteq \mathcal{L}_{P_1}$. Let $r = f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$ be an extended ordered rule such that $\mathcal{L}_r \subseteq (\mathcal{L}_{P_1} \cap \mathcal{L}_{P_2})$.*

We assume that, M is an answer set of $P_1 \cup \{r\}$ iff $M \cap \mathcal{L}_{P_2 \cup \{r\}}$ is an answer set of $P_2 \cup \{r\}$. Then, M is an inclusion preferred answer set of $P_1 \cup \{r\}$ iff $M \cap \mathcal{L}_{P_2 \cup \{r\}}$ is an inclusion preferred answer set of $P_2 \cup \{r\}$. □

Proof of Lemma 6.2 (Sketch).

By hypothesis we know that we can obtain the answer sets of $P_2 \cup \{r\}$ from the answer sets of $P_1 \cup \{r\}$, since M is an answer set of $P_1 \cup \{r\}$ iff $M \cap \mathcal{L}_{P_2 \cup \{r\}}$ is an answer set of $P_2 \cup \{r\}$.

We also know that $P_1 \cup \{r\}$ and $P_2 \cup \{r\}$ have r as their only one ordered disjunction rule. Then, both programs use the same preference criterion over the answer sets of their respective split programs.

Hence, M is an inclusion preferred answer set of $P_1 \cup \{r\}$ iff $M \cap \mathcal{L}_{P_2 \cup \{r\}}$ is an inclusion preferred answer set of $P_2 \cup \{r\}$. □

The following is a corollary of Lemma 6.2. It indicates that given two normal programs and an extended ordered rule, if we add this extended ordered rule to both programs and we assume that the answer sets of one of them are also the answers sets of the other program then the preferred answer sets of one of them are also the preferred answer sets of the other.

Corollary 6.1. *Let P_1 and P_2 be two normal programs such that $\mathcal{L}_{P_1} = \mathcal{L}_{P_2}$. Let $r = f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$ be an extended ordered rule such that $\mathcal{L}_r \subseteq (\mathcal{L}_{P_1} \cap \mathcal{L}_{P_2})$. We assume that, M is an answer set of $P_1 \cup \{r\}$ iff M is an answer set of $P_2 \cup \{r\}$. Then, M is an inclusion preferred answer set of $P_1 \cup \{r\}$ iff M is an inclusion preferred answer set of $P_2 \cup \{r\}$.* □

Proof of Corollary 6.1 (Sketch).

We have to prove two things:

— M is an inclusion preferred answer set of $P_1 \cup \{r\}$ if M is an inclusion preferred answer set of $P_2 \cup \{r\}$, and

— M is an inclusion preferred answer set of $P_2 \cup \{r\}$ if M is an inclusion preferred answer set of $P_1 \cup \{r\}$.

By hypothesis, $\mathcal{L}_{P_2} \subseteq \mathcal{L}_{P_1}$ and M is an answer set of $P_1 \cup \{r\}$ if $M \cap \mathcal{L}_{P_2 \cup \{r\}}$ is an answer set of $P_2 \cup \{r\}$. Then by Lemma 6.2, M is an inclusion preferred answer set of $P_1 \cup \{r\}$ if $M \cap \mathcal{L}_{P_2 \cup \{r\}}$ is an inclusion preferred answer set of $P_2 \cup \{r\}$.

By hypothesis, $\mathcal{L}_{P_1} \subseteq \mathcal{L}_{P_2}$ and M is an answer set of $P_2 \cup \{r\}$ if $M \cap \mathcal{L}_{P_1 \cup \{r\}}$ is an answer set of $P_1 \cup \{r\}$. Then by Lemma 6.2, M is an inclusion preferred answer set of $P_2 \cup \{r\}$ if $M \cap \mathcal{L}_{P_1 \cup \{r\}}$ is an inclusion preferred answer set of $P_1 \cup \{r\}$. □ The following lemma indicates that if a normal program is added a particular restriction then the answer sets of this new program are also answer sets of the program without the restriction.

Lemma 6.3. *Let P be a normal program and let C be an ordered set of atoms such that $C \subseteq \mathcal{L}_P$. If M is an answer set of $P \cup \{\neg(\neg c_1 \wedge \neg c_2 \wedge \cdots \wedge \neg c_n)\}$ then M is an answer set of P.* □

Proof of Lemma 6.3 (Sketch).

By [34] we know that $\{\neg(\neg c_1 \wedge \cdots \wedge \neg c_n)\}$ is strongly equivalent to $\leftarrow \neg c_1, \ldots, \neg c_n$. Then, $\{\neg(\neg c_1 \wedge \cdots \wedge \neg c_n)\}$ corresponds to a constraint. Hence, if M is an answer set of $P \cup \{\neg(\neg c_1 \wedge \cdots \wedge \neg c_n)\}$ then M is an answer set of the program P. □

The following lemma indicates that given a normal program, an extended ordered rule defined in terms of an ordered set of atoms, and a particular rule, then two thing hold:

—the answer sets of the normal program joined with the extended ordered rule defined in terms of an ordered set of atoms are also, the answer sets of the the normal program joined with the extended ordered rule defined in terms of an ordered set of atoms and joined with the particular rule; and

—the answer sets of the the normal program joined with the extended ordered rule defined in terms of an ordered set of atoms and joined with the particular rule are also, the answer sets of the normal program joined only with the particular rule.

We have to mention that the proof of the following lemma uses the previous lemma, i.e., Lemma 6.3.

Lemma 6.4. *Let P be a normal program and let $C = [c_1, c_2, \dots, c_n]$ be an ordered set of atoms such that $C \subseteq \mathcal{L}_P$. Let r_C be the extended ordered rule defined from C. Let r be the following normal rule $all\_pref \leftarrow \neg c_1, \dots, \neg c_n$. Then the following holds:*

1. *M is an answer set of $P \cup \{r_C\}$ iff M is an answer set of $P \cup \{r_C\} \cup \{r\}$.*
2. *M is an answer set of $P \cup \{r_C\} \cup \{r\}$ iff M is an answer set of $P \cup \{r\}$.* □

Proof of Lemma 6.4.

Before we present the proof of this lemma, it is important to recall that by Definition 6.4, M is an answer set of an extended ordered disjunction program iff M is an answer set of one of its split programs. Then, in this proof we are going to consider the split programs of programs $P \cup \{r_C\}$ and $P \cup \{r_C\} \cup \{r\}$. We point out that both programs have only one ordered disjunction rule with at least two choices[3] and at the same time it is the same ordered disjunction rule r_C. Then, both programs have n split programs since they are obtained by replacing r_C by one of its options r_C^i, $1 \leq i \leq n$. In general, each option r_C^i with $1 \leq i \leq n-1$ of r_C has the following form:

$$\neg\neg c_i \leftarrow \neg\neg\neg c_1, \dots, \neg\neg\neg c_{i-1}$$

[3] We recall that if $f_1 \times \dots \times f_n \leftarrow g$ is an extended ordered rule then, the formulas $f_1 \dots f_n$ are usually called the choices of a rule (see Definition 6.1).

Additionally, we know by [34] that r_C^i with $1 \leq i \leq n-1$ is strongly equivalent to $\leftarrow \neg c_1, \ldots, \neg c_i$. And, in particular the option r_C^n has the following form:

$$all\_pref \leftarrow \neg\neg\neg c_1, \ldots, \neg\neg\neg c_n$$

which is strongly equivalent to $all\_pref \leftarrow \neg c_1, \ldots, \neg c_n$.

Then, the split programs of $P \cup \{r_C\}$ have the form: $P \cup \{r_C^i\}$ with $1 \leq i \leq n$ and the split programs of $P \cup \{r_C\} \cup \{r\}$ have the form: $P \cup \{r_C^i\} \cup \{r\}$ with $1 \leq i \leq n$.

Now we are ready to present the proof of both parts.

Proof of (1):

In order to prove this part, we have to prove that M is an answer set of a split program of $P \cup \{r_C\}$ iff M is an answer set of a split program of $P \cup \{r_C\} \cup \{r\}$.

First, let us consider the $n-1$ split programs $P \cup \{r_C^i\}$ and the $n-1$ split programs $P \cup \{r_C^i\} \cup \{r\}$ with $1 \leq i \leq n-1$ since each of them includes the constraint r_C^i, i.e., $\leftarrow \neg c_1, \ldots, \neg c_i$.

We know by [13] that for each $1 \leq i \leq n-1$, $P \cup \{r_C^i\} \cup \{r\}$ is a conservative extension of $P \cup \{r_C^i\}$, i.e., for each $1 \leq i \leq n-1$, M is an answer set of $P \cup \{r_C^i\} \cup \{r\}$ iff $M \cap \mathcal{L}_P$ is an answer set of $P \cup \{r_C^i\}$. Let us recall that in each $P \cup \{r_C^i\} \cup \{r\}$ with $1 \leq i \leq n-1$ the constraint $r_C^i := \leftarrow \neg c_1, \ldots, \neg c_i$, is included. This implies that the atom all_pref does not occur in any of the answers set of the split programs $P \cup \{r_C^i\} \cup \{r\}$ with $1 \leq i \leq n-1$ since the only possibility to infer all_pref is that the body of rule r holds in any of the answer sets of these split programs, however the constraint r_C^i avoids this possibility. Hence, we can infer that for each $1 \leq i \leq n-1$, M is an answer set of $P \cup \{r_C^i\} \cup \{r\}$ iff M is an answer set of $P \cup \{r_C^i\}$.

However, in order to finish with this proof, we also have to consider the split programs $P \cup \{r_C^n\} \cup \{r\}$ and $P \cup \{r_C^n\}$. It is very easy to see that $P \cup \{r_C^n\} \cup \{r\}$ and $P \cup \{r_C^n\}$ are equivalent, since $\{r_C^n\}$ and $\{r\}$ are the same rule. Hence, M is an answer set of $P \cup \{r^n_C\} \cup \{r\}$ iff M is an answer set of $P \cup \{r^n_C\}$.

Proof of (2):

In order to prove this part we have to prove two things:

— if M is an answer set of a split program of $P \cup \{r_C\} \cup \{r\}$ then M is an answer set of $P \cup \{r\}$, and

— if M is an answer set of $P \cup \{r\}$ then M is an answer set of a split program of $P \cup \{r_C\} \cup \{r\}$.

To prove the first part, let us consider the $n-1$ split programs $P \cup \{r^i_C\} \cup \{r\}$ with $1 \leq i \leq n-1$ that include the constraint r^i_C, i.e., $\leftarrow \neg c_1, \ldots, \neg c_i$. We know that each constraint r^i_C with $1 \leq i \leq n-1$ is strongly equivalent to $\{\neg(\neg c_1 \wedge \cdots \wedge \neg c_{i-1})\}$. Then according to Lemma 6.3, if M is an answer set of some $P \cup \{r^i_C\} \cup \{r\}$, $1 \leq i \leq n-1$ then M is an answer set of $P \cup \{r\}$. However, in order to finish with the proof of this part, we also have to consider the split program $P \cup \{r^n_C\} \cup \{r\}$. It is very easy to see that $P \cup \{r^n_C\} \cup \{r\}$ and $P \cup \{r\}$ are equivalent, since $\{r^n_C\}$ and $\{r\}$ are the same rule. Hence, M is an answer set of $P \cup \{r^n_C\} \cup \{r\}$ iff M is an answer set of $P \cup \{r\}$.

Now, we have to prove the second part. If M is an answer set of $P \cup \{r\}$ then M is an answer set of a split program of $P \cup \{r_C\} \cup \{r\}$. Let us consider the split program $P \cup \{r^n_C\} \cup \{r\}$. As we mentioned previously, it is very easy to see that $P \cup \{r^n_C\} \cup \{r\}$ and $P \cup \{r\}$ are equivalent, since $\{r^n_C\}$ and $\{r\}$ are the same rule. □

Thanks to the previous lemma, Lemma 6.4, we know that a normal program joined with a particular extended ordered rule and a particular rule has the same answer sets that the normal program joined only with the extended ordered rule. Now the following lemma indicates that they also have the same inclusion preferred answer sets.

Lemma 6.5. *Let P be a normal program and let $C = [c_1, c_2, \ldots, c_n]$ be an ordered set of atoms such that $C \subseteq \mathcal{L}_P$. Let r_C be the extended ordered rule defined from C. Let r be* *the following normal rule* $all\_pref \leftarrow \neg c_1, \ldots, \neg c_n$. *Then* M *is an inclusion preferred answer set of* $P \cup \{r_C\}$ *iff* M *is an inclusion preferred answer set of* $P \cup \{r_C\} \cup \{r\}$. □

Proof of Lemma 6.5.

By Lemma 6.4 we know that M is an answer set of $P \cup \{r_C\}$ iff M is an answer set of $P \cup \{r_C\} \cup \{r\}$, i.e., both programs have the same answer sets. We also know that $P \cup \{r_C\}$ and $P \cup \{r_C\} \cup \{r\}$ have r_C as their only one ordered disjunction rule that has al least two choices. Then, by Corollary 6.1 M is an inclusion preferred answer set of $P \cup \{r_C\}$ iff M is an inclusion preferred answer set of $P \cup \{r_C\} \cup \{r\}$. □

The following lemma allows one to obtain the most preferred answer sets of a normal program joined with a particular rule with respect to an ordered set of atoms in terms of the inclusion preferred answer sets of a particular extended ordered disjunction program. This particular extended ordered disjunction program is defined in terms of the original normal program joined with the extended ordered rule defined with respect to the ordered set of atoms and the particular rule. Then the proof of this lemma uses Lemma 6.4.

Lemma 6.6. *Let* P *be a normal program and let* $C = [c_1, c_2, \ldots, c_n]$ *be an ordered set of atoms such that* $C \subseteq \mathcal{L}_P$. *Let* r_C *be the extended ordered rule defined from* C. *Let* r *be the following normal rule* $all\_pref \leftarrow \neg c_1, \ldots, \neg c_n$. *Then* M *is an inclusion preferred answer set of* $P \cup \{r_C\} \cup \{r\}$ *iff* M *is the most preferred answer set with respect to* $C \cup (P \cup \{r\})$. □

Proof of Lemma 6.6 (Sketch).

By Lemma 6.4 we know that M is an answer set of $P \cup \{r_C\} \cup \{r\}$ iff M is an answer set of $P \cup \{r\}$, i.e., $P \cup \{r_C\} \cup \{r\}$ and $P \cup \{r\}$ have the same answer sets.

Then, in order to obtain the inclusion preferred answer set of $P \cup \{r_C\} \cup \{r\}$, we can only consider the answer sets of program $P \cup \{r\}$ and then apply the inclusion preference criterion to them. Additionally, in order to obtain the most preferred answer set with respect to $C \cup (P \cup \{r\})$, first we have to obtain the answers sets of $P \cup \{r\}$ and then we have to apply a preference criterion based on the ordered set of atoms C.

We can verify that the preference criterion to obtain the most preferred answer set with respect to $C \cup (P \cup \{r\})$ corresponds to a particular case of the preference criterion to obtain the inclusion preferred answer sets of an extended ordered disjunction program. Specifically, this particular case corresponds to an extended ordered disjunction program that has r_C as its only one extended ordered rule. Then, in this particular case the preference criterion to obtain the inclusion preferred answer sets is reduced exactly to the preference criterion to obtain the most preferred answer set with respect to $C \cup (P \cup \{r\})$.

Hence, M is an inclusion preferred answer set of $P \cup \{r_C\} \cup \{r\}$ iff M is one of the most preferred answer set with respect to $C \cup (P \cup \{r\})$, since we are applying the same preference criterion. □

Finally, we present Lemma 6.7 that allows us to obtain the most preferred answer set of a normal program with respect to an ordered set of formulas in terms of a particular extended ordered program. This particular extended ordered program results from joining the original normal program and a particular ordered disjunction rule that is defined in terms of the ordered set of atoms. The proof of this lemma uses the previous Lemmas 6.5 and 6.6.

Lemma 6.7. *Let P be a normal program and let $C = [c_1, c_2, \ldots, c_n]$ be an ordered list of atoms such that $C \subseteq \mathcal{L}_P$. Let r_C be the extended ordered rule defined from C. Then M is an inclusion preferred answer set of $P \cup r_C$ iff $(M \cap \mathcal{L}_P)$ is the most preferred answer set with respect to $C \cup P$.* □

Proof of Lemma 6.7 (Sketch).

By Lemma 6.5, M is an inclusion preferred answer set of $P \cup \{r_C\}$ iff M is an inclusion preferred answer set of $P \cup \{r_C\} \cup \{r\}$.

By Lemma 6.6, M is an inclusion preferred answer set of $P \cup \{r_C\} \cup \{r\}$ iff M is the most preferred answer set with respect to $C \cup (P \cup r)$.

Finally by [13], we can verify that $C \cup (P \cup r)$ is a conservative extension of $C \cup P$, i.e., M is the most preferred answer set with respect to $C \cup (P \cup r)$ iff $(M \cap \mathcal{L}_P)$ is the most preferred answer set with respect to $C \cup P$. □

Example 6.12. Let us consider again program P from Example 6.10:

$$a \leftarrow .$$
$$c \leftarrow \neg b.$$
$$b \leftarrow \neg c.$$

and also the ordered list of atoms $C = [b, c]$. Then, $P \cup r_C$ is the following program:

$$a \leftarrow .$$
$$c \leftarrow \neg b.$$
$$b \leftarrow \neg c.$$
$$\neg\neg b \times \neg\neg c \times all\_pref.$$

We can verify that $\{a, b\}$ is *the most preferred answer set with respect to* $C \cup P$ and that is also an inclusion preferred answer set of $P \cup r_C$. □

In the following subsection we shall show how a program with extended ordered rules using negated negative literals can be easily translated into a standard ordered program. Then using *PSMODELS* we can obtain the preferred answer sets [46].

6.3.1 Computing preferred answer sets for extended ordered programs

It is worth mentioning that neither running *PSMODELS* [4] [9] nor following the definition given by Brewka [7] for ordered disjunction we can obtain the inclusion preferred answer sets for extended ordered programs. The reason is that the definition given by Brewka for ordered disjunction has syntactical restrictions. However, in particular when this program has extended ordered rules using negated negative literals, we can use Lemma 6.15 that allows us to translate easily this program to a standard ordered program and then use *PSMODELS* to obtain the preferred answer sets [46]. The proof of Lemma 6.15 uses other auxiliary lemmas that we introduce before we introduce Lemma 6.15. We have to mention that most of these auxiliary lemmas are for general purpose.

The following lemma indicates that the answers sets of a given normal program joined to an extended ordered rule are subsets of the answers sets of the same normal program joined to the same extended ordered rule and to a formula $a'_i \leftrightarrow f_i$ such that a'_i is an atom that does not appear in the original program and f_i is a choice in the extended ordered rule.

Lemma 6.8. *Let P be a normal program. Let $r = f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$ be an extended ordered rule. Let r' be the rule $a'_i \leftrightarrow f_i$ such that a'_i is an atom and $a'_i \notin \mathcal{L}_{P \cup \{r\}}$. Then M is an answer set of $P \cup \{r\} \cup \{r'\}$ iff $M \cap \mathcal{L}_{P\cup\{r\}}$ is an answer set of $P \cup \{r\}$.* □

Proof of Lemma 6.8.

Since, $P \cup \{r\} \cup \{r'\}$ and $P \cup \{r\}$ are extended ordered disjunction programs, then in order to prove this lemma we have to consider the split programs of both programs and verify the following: M is an answer set of a split program of $P \cup \{r\} \cup \{r'\}$ iff

[4] http://www.tcs.hut.fi/Software/smodels/priority/

$M \cap \mathcal{L}_{P\cup\{r\}}$ is an answer set of a split program of $P \cup \{r\}$, i.e., M is an answer set of the split program $P \cup \{r^i\} \cup \{r'\}$ iff $M \cap \mathcal{L}_{P\cup\{r\}}$ is an answer set of the split program $P \cup \{r^i\}$ where r^i is an option of r.

By [13], we can verify that $P \cup \{r^i\} \cup \{r'\}$ is a conservative extension of $P \cup \{r^i\}$, i.e., M is an answer set of $P \cup \{r^i\} \cup \{r'\}$ iff $M \cap \mathcal{L}_{P\cup\{r\}}$ is an answer set of $P \cup \{r^i\}$. □

Thanks to the previous lemma, Lemma 6.8, we know that the answers sets of a given normal program joined to an extended ordered rule are subsets of the answers sets of the same normal program joined to the same extended ordered rule and to a formula $a'_i \leftrightarrow f_i$ such that a'_i is an atom that does not appear in the original program and f_i is a choice in the extended ordered rule. Now the following lemma indicates that they also have the same inclusion preferred answer sets.

Lemma 6.9. *Let P be a normal program. Let $r = f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$ be an extended ordered rule. Let r' be the rule $a'_i \leftrightarrow f_i$ such that a'_i is an atom and $a'_i \notin \mathcal{L}_{P\cup\{r\}}$. Then M is an an inclusion preferred answer set of $P \cup \{r\} \cup \{r'\}$ iff $M \cap \mathcal{L}_{P\cup\{r\}}$ is an inclusion preferred answer set of $P \cup \{r\}$.* □

Proof of Lemma 6.9.

By Lemma 6.8 we know that M is an answer set of $P \cup \{r\} \cup \{r'\}$ iff $M \cap \mathcal{L}_{P\cup\{r\}}$ is an answer set of $P \cup \{r\}$. We also know that $P \cup \{r\} \cup \{r'\}$ and $P \cup \{r\}$ have r as their only one ordered disjunction rule that has al least two choices. Then, by Lemma 6.2 M is an an inclusion preferred answer set of $P \cup \{r\} \cup \{r'\}$ iff $M \cap \mathcal{L}_{P\cup\{r\}}$ is an inclusion preferred answer set of $P \cup \{r\}$. □

Now, we define the replacement of a choice in an extended ordered rule by an atom that does not appear in the original extended ordered rule.

Definition 6.14. Let $r = f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$ be an extended ordered rule. Let a be an atom such that $a \notin \mathcal{L}_{\{r\}}$. Given a choice $f_i, 1 \leq i \leq n$ of rule r, the replacement of f_i for a, denoted as $r[f_i/a]$, corresponds to the following extended ordered rule: $f_1 \times \ldots \times a \times \ldots \times f_n \leftarrow g$. □

For instance, if we consider r as the following extended ordered rule $a \times (a \wedge b) \times \neg\neg a \times c \leftarrow e, \neg f$ then $r[\neg\neg a/a'] = a \times (a \wedge b) \times a' \times c \leftarrow e, \neg f$.

The following lemma, Lemma 6.10, indicates that the answers sets of a given normal program joined to an extended ordered rule and to a formula $a'_i \leftrightarrow f_i$ such that a'_i is an atom that does not appear in the original program and f_i is a choice in the extended ordered rule, are also answer sets of the same ordered program but replacing the choice f_i for the atom a'_i. Additionally, Lemma 6.11 indicates that these programs have the same inclusion preferred answer sets.

Lemma 6.10. *Let P be a normal program. Let $r = f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$ be an extended ordered rule. Let r' be the rule $a'_i \leftrightarrow f_i$ such that a'_i is an atom and $a'_i \notin \mathcal{L}_{P \cup \{r\}}$. Then M is an answer set of $P \cup \{r\} \cup \{r'\}$ iff M is an answer set of $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$.* □

Proof of Lemma 6.10.

Since, $P \cup \{r\} \cup \{r'\}$ and $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$ are extended ordered disjunction programs, then in order to prove this lemma we have to consider the split programs of both programs and verify the following: M is an answer set of a split program of $P \cup \{r\} \cup \{r'\}$ iff M is an answer set of a split program of $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$, i.e., M is an answer set of the split program $P \cup \{r^i\} \cup \{r'\}$ iff M is an answer set of the split program $P \cup \{r^i[f_i/a'_i]\} \cup \{r'\}$ where r^i and $r^i[f_i/a'_i]$ are the options of r and $r[f_i/a'_i]$.

By [34], we can verify that $P \cup \{r^i\} \cup \{r'\}$ is equivalent to $P \cup \{r^i[f_i/a'_i]\} \cup \{r'\}$, i.e., M is an answer set of $P \cup \{r^i\} \cup \{r'\}$ iff M is an answer set of $P \cup \{r^i[f_i/a'_i]\} \cup \{r'\}$. □

Lemma 6.11. *Let P be a normal program. Let $r = f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$ be an extended ordered rule. Let r' be the rule $a'_i \leftrightarrow f_i$ such that a'_i is an atom and*

$a'_i \notin \mathcal{L}_{P \cup \{r\}}$. *Then* M *is an inclusion preferred answer set of* $P \cup \{r\} \cup \{r'\}$ *iff* M *is an inclusion preferred answer set of* $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$. □

Proof of Lemma 6.11.

By Lemma 6.10 we know that M is an answer set of $P \cup \{r\} \cup \{r'\}$ iff M is an answer set of $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$. We also know that $P \cup \{r\} \cup \{r'\}$ and $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$ have r and $r[f_i/a'_i]$ as their respective ordered disjunction rules. Then, both programs use the same preference criterion over the answer sets of their respective split programs, except for f_i and a_i.

We have to recall that the criterion to obtain the inclusion preferred answer sets of an extended ordered disjunction program is based on the satisfaction degree of each extended ordered disjunction rule, $f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$, w.r.t. the answer sets of this program. Additionally, the satisfaction degree is defined in terms of the inference of the choice f_i by an answer set of the program (see Definition 6.5, Definition 6.9, Definition 6.7 and Definition 6.6).

Then, it is enough to mention that the satisfaction degree of r is the same for $r[f_i/a'_i]$, since each answer set of $P \cup \{r\} \cup \{r'\}$ that infers f_i also infers a_i.

Hence, M is an inclusion preferred answer set of $P \cup \{r\} \cup \{r'\}$ iff M is an inclusion preferred answer set of $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$. □

The following lemma, Lemma 6.12, indicates that the answers sets of a given normal program joined to an extended ordered rule are subsets of the answer sets of the same normal program joined to the same extended ordered rule where one of its choices f_i is replaced by an atom a'_i and also joined to the formula $a'_i \leftrightarrow f_i$. Additionally, Lemma 6.13 indicates that we can obtain the inclusion preferred answer sets of the second program from the inclusion preferred answer sets of the first one. The proof of Lemma 6.12 uses Lemma 6.10 and Lemma 6.8.

Lemma 6.12. *Let P be a normal program. Let $r = f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$ be an extended ordered rule. Let r' be the rule $a'_i \leftrightarrow f_i$ such that a'_i is an atom and $a'_i \notin \mathcal{L}_{P\cup\{r\}}$. Then M is an answer set of $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$ iff $M \cap \mathcal{L}_{P\cup\{r\}}$ is an answer set of $P \cup \{r\}$.* □

Proof of Lemma 6.12 (Sketch).

By Lemma 6.10 we know that M is an answer set of $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$ iff M is an answer set of $P \cup \{r\} \cup \{r'\}$. Then, we have to prove that M is an answer set of $P \cup \{r\} \cup \{r'\}$ iff $M \cap \mathcal{L}_{P\cup\{r\}}$ is an answer set of $P \cup \{r\}$. By Lemma 6.8, it holds. □

Lemma 6.13. *Let P be a normal program. Let $r = f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$ be an extended ordered rule. Let r' be the rule $a'_i \leftrightarrow f_i$ such that a'_i is an atom and $a'_i \notin \mathcal{L}_{P\cup\{r\}}$. Then M is an an inclusion preferred answer set of $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$ iff $M \cap \mathcal{L}_{P\cup\{r\}}$ is an inclusion preferred answer set of $P \cup \{r\}$.* □

Proof of Lemma 6.13.

By Lemma 6.12 we know that M is an answer set of $P \cup \{r\} \cup \{r'\}$ iff $M \cap \mathcal{L}_{P\cup\{r\}}$ is an answer set of $P \cup \{r\}$.

We also know that $P \cup \{r\} \cup \{r'\}$ and $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$ have r as their only one ordered disjunction rule.

We also know that $P \cup \{r\} \cup \{r'\}$ and $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$ have r and $r[f_i/a'_i]$ as their respective ordered disjunction rules. Then, both programs use the same preference criterion over the answer sets of their respective split programs, except for f_i and a_i.

We have to recall that the criterion to obtain the inclusion preferred answer sets of an extended ordered disjunction program is based on the satisfaction degree of each extended ordered disjunction rule, $f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$, w.r.t. the answer sets of this program. Additionally, the satisfaction degree is defined in terms of the inference of the choice f_i by an answer set of the program (see Definition 6.5, Definition 6.9, Definition 6.7 and Definition 6.6).

Then, it is enough to mention that the satisfaction degree of r is the same for $r[f_i/a'_i]$, since each answer set of $P \cup \{r\} \cup \{r'\}$ that infers f_i also infers a_i.

Hence, M is an inclusion preferred answer set of $P \cup \{r\} \cup \{r'\}$ iff M is an inclusion preferred answer set of $P \cup \{r[f_i/a'_i]\} \cup \{r'\}$. □

Given an extended ordered rule $r = f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$ and an ordered set of atoms $A = [a_1, \ldots, a_n]$ such that $\mathcal{L}_r \cap A = \emptyset$. We define the *ordered replacements* of each f_i, $1 \leq i \leq n$ for atoms in A, denoted as $r[A]_n$, are defined inductively as follows:

$$r[A]_1 = r[f_1/a_1],$$
$$r[A]_{n+1} = r[A]_n[f_{n+1}/a_{n+1}]$$

For instance, if we consider r as the following extended ordered rule $a \times (a \wedge b) \times \neg\neg a \times c \leftarrow e, \neg f$ and the ordered set of atoms $A = [a_1, a_2, a_3, a_4]$ then

$$r[A]_1 = r[f_1/a_1] = a_1 \times (a \wedge b) \times \neg\neg a \times c \leftarrow e, \neg f,$$
$$r[A]_2 = r[A]_1[f_2/a_2] = a_1 \times a_2 \times \neg\neg a \times c \leftarrow e, \neg f,$$
$$r[A]_3 = r[A]_2[f_3/a_3] = a_1 \times a_2 \times a_3 \times c \leftarrow e, \neg f,$$
$$r[A]_4 = r[A]_3[f_4/a_4] = a_1 \times a_2 \times a_3 \times a_4 \leftarrow e, \neg f.$$

The following lemma is a generalization on Lemma 6.13 where instead of replace only one choice in the extended ordered rule, we propose an ordered replacement of the choices of this extended ordered rule.

Lemma 6.14. *Let P be a normal program. Let $r = f_1 \times \ldots \times f_i \times \ldots \times f_n \leftarrow g$ be an extended ordered rule. Let $A = [a_1, \ldots, a_n]$ be an ordered set of atoms such that $\mathcal{L}_{P \cup \{r\}} \cap A = \emptyset$. Let $k \leq n$ and $D^k = \{a_i \leftrightarrow f_i, 1 \leq i \leq k \mid f_i$ is a choice of r and $a_i \in A\}$. Then M is an inclusion preferred answer set of $P \cup \{r[A]_k\} \cup D^k$ iff $M \cap \mathcal{L}_{P \cup \{r\}}$ is an inclusion preferred answer set of $P \cup \{r\}$.*

Proof of Lemma 6.14 (Sketch).

This proof is straightforward by induction on k, using the extended ordered program that results from the *ordered replacements* of choices of r for atoms in A and Lemma 6.13 in each step.

□

Finally, we present Lemma 6.15 that allows us to translate an extended ordered program that results from joining a normal program with an extended ordered disjunction rule with negated negative literals to a standard ordered program. The proof of this lemma uses Lemma 6.14.

In the following definition and lemma the atoms $a^\bullet$, a°, are atoms that do not occur in the original program P.

Definition 6.15. Let $\neg\neg a$ be a negated negative literal. We define the associated set of $\neg\neg a$ as follows:

$R(\neg\neg a) := \{ \ \leftarrow \ \neg a, a^\bullet \ , \qquad a^\bullet \leftarrow \ \neg a^\circ \ , \qquad a^\circ \leftarrow \ \neg a \ , \qquad \leftarrow \ a, a^\circ \ \}.$ □

Lemma 6.15. *Let P be a normal program and let $C = [c_1, c_2, \ldots, c_n]$ be an ordered set of atoms such that $C \subseteq \mathcal{L}_P$. Let $C^\bullet = \{c_1^\bullet, c_2^\bullet, \ldots, c_n^\bullet\}$ be a set of atoms such that $C^\bullet \cap \mathcal{L}_P = \emptyset$. Let r_C be the extended ordered rule defined from C. Let $r_C^\bullet$ be the following ordered rule $c_1^\bullet \times c_2^\bullet \times \ldots \times c_n^\bullet \times all\_pref$. Let $A = \bigcup_{c_i \in C \ and \ 1 \leq i \leq n} R(\neg\neg c_i)$. Then M is an inclusion preferred answer set of $P \cup \{r_C^\bullet\} \cup A$ iff $M \cap \mathcal{L}_P$ is an inclusion preferred answer set of $P \cup \{r_C\}$.* □

Proof of Lemma 6.15 (Sketch).

This proof is straightforward using Lemma 6.14 and Definition 6.15. □

Example 6.13. Let us consider again the program P at the beginning of this Section

$$a \leftarrow .$$

$$b \leftarrow \neg c.$$

$$c \leftarrow \neg b.$$

$$d \leftarrow \neg a.$$

$$f \leftarrow c, \neg a.$$

$$e \leftarrow b, \neg a.$$

and the set of atoms $C = [f, c]$ then $r_C = \neg\neg f \times \neg\neg c \times all\_pref$,

$A = \{\leftarrow\ \neg f, f^{\bullet}.\ f^{\bullet} \leftarrow\ \neg f^{\circ}.\ f^{\circ} \leftarrow\ \neg f.\ \leftarrow\ f, f^{\circ}.$

$\leftarrow\ \neg c, c^{\bullet}.\ \ c^{\bullet} \leftarrow\ \neg c^{\circ}.\ c^{\circ} \leftarrow\ \neg c.\ \leftarrow\ c, c^{\circ}.\quad \}$ and

$r^{\bullet}_C = \{f^{\bullet} \times c^{\bullet} \times all\_pref\}$.

Then, by running *PSMODELS* we obtain the following inclusion preferred answer set of the standard ordered program $P \cup r^{\bullet}_C \cup A$: $\{a, c, c^{\bullet}, f^{\circ}\}$. Finally, we can see that the intersection of the answer set with $\mathcal{L}_P$ corresponds to the inclusion preferred answer sets of the original extended ordered program $P \cup r_C$ as we described before: $\{a, c\}$. □

6.3.2 Obtaining the maximal answer sets of a program with respect to a set of atoms

Additionally, in this section we show how we can also use extended ordered disjunction programs with *negated negative literals* to obtain the maximal answer sets of a program w.r.t. a set of atoms. For instance, if the answer sets of a program P are $\{b, c, e\}$, $\{b, c, d\}$ $\{f, e\}$ and $\{e, a, c\}$ then $\{b, c, d\}$ and $\{f, e\}$ are the maximal answer sets with respect the set of atoms $A = \{b, d, f\}$. A possible real application of this is described in [28]. In [28] there is a description of a real application using ASP to perform decision making based on an Argument Framework (AF) in the domain of organ transplantation.

Then, as a second possible application of extended ordered disjunction programs with *negated negative literals* we proposed to use them to obtain the maximal answer sets of a program characterizing an AF such that these maximal answer sets correspond to the preferred extensions of the AF.

The formal definition of a maximal answer set with respect to a set of atoms is based on the definition of maximal set with respect to a set.

Definition 6.16 (Maximal set w.r.t. a set A)**.** [28] Let $\{S_i : i \in I\}$ be a collection of subsets of U such that $\bigcup_{i \in I} S_i = U$ and $A \subseteq U$. We say that S_i is a maximal set w.r.t. A among the collection $\{S_i : i \in I\}$ iff there is no S_j with $j \neq i$ such that $(S_i \cap A) \subset (S_j \cap A)$. □

Definition 6.17 (Maximal answer set w.r.t. a set A)**.** [28] Let P be a consistent program and $\{M_i : i \in I\}$ be the collection of answer sets of P. Let $A \subseteq \mathcal{L}_P$. We say that M_i is a maximal answer set w.r.t. A iff M_i is an answer set of P such that M_i is a maximal set w.r.t. A among the collection of answer sets of P. □

In order to obtain the maximal answer sets with respect to a set of atoms, the original program P is extended with a set of extended ordered rules using negated negative literals. Each extended ordered rule is defined from an atom in the given set of atoms A. For instance, in the previous example the set of extended ordered rules is the following: $\{\neg\neg b \times b^{\bullet}.\ \neg\neg d \times d^{\bullet}.\ \neg\neg f \times f^{\bullet}.\}$ where $b^{\bullet}$, $d^{\bullet}$ and $f^{\bullet}$ are atoms that do not occur in the original program. Then the extended ordered program is the following: $P \cup \{\neg\neg b \times b^{\bullet}.\ \neg\neg d \times d^{\bullet}.\ \neg\neg f \times f^{\bullet}.\}$

The following Lemma formalizes our previous discussion about the use of negated negative literals in an extended ordered program to obtain the maximal answer sets of a program w.r.t. a set of atoms.

Definition 6.18. Let P be a program and $S \subseteq \mathcal{L}_P$. We define a translations of P w.r.t. S into an ordered program, denoted by $ord_{set}(P, S)$: First, we define a set of ordered clauses w.r.t. S as follows: $C_S = \{\neg\neg a \times a^{\bullet} \mid a \in S \text{ and } a^{\bullet} \notin \mathcal{L}_P\}$. Then, $ord_{set}(P, S) = P \cup C_S$. □

Lemma 6.16. *Let P be a program and M be an answer set of P. Let $S \subseteq \mathcal{L}_P$. Then M is an inclusion preferred answer set of $ord_{set}(P, S)$ iff $M \cap \mathcal{L}_P$ is a maximal answer set of P w.r.t. S.* □

6.4 Conclusion

In this chapter we introduced *extended ordered disjunction programs*. The definition presented here extends ordered disjunction programs to a wider class of logic programs [35]. We also proposed a characterization of minimal generalized answer sets in terms of ordered disjunction programs.

In this chapter we also proposed to use extended ordered disjunction programs with double default negation in each atom to define preferences and to find the maximal answer set of a program. We will see that preferences defined in this way will help us to find the preferred evacuation plans using language $\mathcal{PP}$. Additionally, we propose to use the definition of the maximal answer set of a program to obtain the preferred extensions of an Argumentation Framework.

Finally, we shown how to compute the preferred answer sets for extended ordered programs with double negation using PSMODELS.