A Descriptive Approach to Preferred Answer Sets

Ján Šefránek(✉) and Alexander Šimko

Comenius University, Bratislava, Slovakia
{sefranek, simko}@fmph.uniba.sk

Abstract. We are aiming at a semantics of logic programs with preferences defined on rules, which always selects a preferred answer set, if there is a non-empty set of (standard) answer sets of the given program.

It is shown in a seminal paper by Brewka and Eiter that the goal mentioned above is incompatible with their second principle and it is not satisfied in their semantics of prioritized logic programs. Similarly, also according to other established semantics, based on a prescriptive approach, there are programs with standard answer sets, but without preferred answer sets.

According to the standard prescriptive approach no rule can be fired before a more preferred rule, unless the more preferred rule is blocked. This is a rather imperative approach, in its spirit. According to our background intuition, rules can be blocked by more preferred rules, but the rules which are not blocked are handled in a more declarative style, independent on the given preference relation on the rules.

An argumentation framework (different from Dung's framework) is proposed in this paper. Some argumentation structures are assigned to the rules of a given program. Other argumentation structures are derived using a set of derivation rules. Some of the derived argumentation structures correspond to answer sets. An attack relation on derivations of argumentation structures is defined. Preferred answer sets correspond to complete argumentation structures, which are not blocked by other complete argumentation structures.

Keywords: Extended logic program · Answer set · Preference · Preferred answer set · Argumentation structure

1 Introduction

The meaning of a nonmonotonic theory is often characterized by a set of *(alternative) belief sets*. It is appropriate to select sometimes some of the belief sets as more *preferred*.

We are focused in this paper on *extended logic programs* with a preference relation on rules, see, e.g., [1,2,10,18]. Such programs are denoted by the term *prioritized* logic programs in this paper.

It is suitable to require that some preferred answer sets can be *selected* from a non-empty set of the standard answer sets of a (prioritized) logic program.

H. Tompits et al. (Eds.): INAP/WLP 2011, LNAI 7773, pp. 195–214, 2013.
DOI: 10.1007/978-3-642-41524-1_11,

Unfortunately, there are prioritized logic programs with standard answer sets, but without preferred answer sets according to the semantics of [1] and also of [2] or [18]. This feature is a consequence of the *prescriptive* approach to preference handling [4]. According to that approach no rule can be fired before a more preferred rule, unless the more preferred rule is blocked. This is a rather imperative approach, in its spirit.

An investigation of basic *principles* which should be satisfied by any system containing a preference relation on defeasible rules is of fundamental importance. This type of research has been initialized in the seminal paper [1]. Two basic principles are accepted in the paper.

The second of the principles is violated, if a function is assumed which always selects a non-empty subset of preferred answer sets from a non-empty set of all standard answer sets of a prioritized logic program.

We believe that the possibility to select always a preferred answer set from a non-empty set of standard answer sets is of critical importance. This goal requires to accept a *descriptive* approach to preference handling. The approach is characterized in [3,4] as follows: The order in which rules are applied, reflects their "desirability" – it is desirable to apply the most preferred rules.

Our basic intuition is that rules can be *blocked* by more preferred rules, but the rules which are not blocked are handled in a more declarative style. If we express this in terms of desirability, it is desirable *to apply all (applicable) rules which are not blocked by a more preferred rule.* The execution of non-blocked rules does not depend on their order. Dependencies of more preferred rules on less preferred rules do not prevent the execution of non-blocked rules. However, this approach is computationally more demanding than the prescriptive approach.

A formal elaboration of this intuition resulted in our approach into attack and blocking relations between sets of generating rules (expressed in terms of derivations of complete argumentation structures).

Our goal is:

- to modify the principles proposed by Brewka and Eiter in [1] in such a way that they do not contradict a selection of a non-empty set of preferred answer sets from the underlying non-empty set of standard answer sets, and
- to introduce a notion of preferred answer sets that satisfies the above mentioned modification.

The proposed method is inspired by [8]. While there defeasible rules are treated as (defeasible) arguments, here (defeasible) assumptions, sets of default negations, are considered as arguments. Reasoning about preferences in a logic program is here understood as a kind of argumentation. Sets of default literals can be viewed as defeasible arguments, which may be contradicted by consequences of some applicable rules. The preference relation on rules is used in order to ignore the attacks of less preferred arguments against more preferred arguments. The core problem is to transfer the preference relation defined on rules to a blocking relation between answer sets.[1]

The basic argumentation structures correspond to the rules of a given program. Derivation rules, which enable derivation of argumentation structures from the basic ones are defined. That derivation leads from the basic argumentation structures to argumentation structures corresponding to answer sets of the given program (we use a method of [5]). The argumentation structures, which correspond to answer sets, are called *complete* in this paper.

Derivations of complete argumentation structures play a crucial role in our approach. Attacks of more preferred rules against the less preferred rules are transferred to attacks between derivations of complete argumentation structures. Preferred answer sets are defined over that background. They correspond to complete and non-blocked (warranted) argumentation structures.

The contributions of this paper are summarized as follows. A modified set of principles for preferred answer set specification is proposed. A new type of argumentation framework is constructed, which enables a selection of preferred answer sets. There are basic arguments (argumentation structures) and attacks in the framework. Rules for derivation of argumentation structures are introduced. After that attacks between derivations of complete argumentation structures, acceptable derivations and, finally, warranted and blocked complete argumentation structures are defined. Preferred answer sets are defined in terms of complete and non-blocked (warranted) argumentation structures. Each program with a non-empty set of answer sets has a preferred answer set in our approach.

A preliminary version of the presented research has been published in [12]. This is more than an extended version of [13]. The main differences between the versions are summarized in the Conclusions.

2 Preliminaries

The language of extended logic programs is used in this paper.

Let At be a set of atoms. The set of *objective literals* is defined as $Obj = At \cup \{\neg A : A \in At\}$. If L is an objective literal then an expression of the form $not\ L$ is called *default* literal. Notation: $Def = \{not\ L \mid L \in Obj\}$. The set of literals Lit is defined as $Obj \cup Def$.

A convention: $\neg\neg A = A$, where $A \in At$. If X is a set of objective literals, then $not\ X = \{not\ L \mid L \in X\}$.

A *rule* is each expression of the form $L \leftarrow L_1, \ldots, L_k$, where $k \geq 0$, $L \in Obj$ and $L_i \in Lit$. If r is a rule of the form as above, then L is denoted by $head(r)$ and $\{L_1, \ldots, L_k\}$ by $body(r)$. If R is a set of rules, then $head(R) = \{head(r) \mid r \in R\}$ and $body(R) = \{body(r) \mid r \in R\}$. A finite set of rules is called *extended logic program* (program hereafter).

[1] Our intuitions connected to the notion of argumentation structure and also the used constructions are different from Dung's arguments or from arguments of [8]. This paper does not present a contribution to argumentation theory.

The set of *conflicting literals* is defined as $CON = \{(L_1, L_2) \mid L_1 = not\ L_2 \vee L_1 = \neg L_2\}$. A set of literals S is *consistent* if $(S \times S) \cap CON = \emptyset$. An *interpretation* is a consistent set of literals. A *total* interpretation is an interpretation I such that for each objective literal L either $L \in I$ or $not\ L \in I$. A literal L is *satisfied* in an interpretation I iff $L \in I$ (notation: $I \models L$). A set of literals S is satisfied in I iff $S \subseteq I$ (notation: $I \models S$). A rule r is satisfied in I iff $I \models head(r)$ whenever $I \models body(r)$, notation $I \models r$. An interpretation I is a *model* of a program P, notation $I \models P$, if for each $r \in P$ holds $I \models r$.

If S is a set of literals, then we denote $S \cap Obj$ by S^+ and $S \cap Def$ by S^-. Symbols $(body(r))^-$ and $(body(r))^+$ are used here in that sense (notice that the usual meaning of $body^-(r)$ is different). If $X \subseteq Def$ then $pos(X) = \{L \in Obj \mid not\ L \in X\}$. Hence, $not\ pos((body(r))^-) = (body(r))^-$. If r is a rule, then the rule $head(r) \leftarrow (body(r))^+$ is denoted by r^+.

An answer set of a program can be defined as follows (only consistent answer sets are defined).

A total interpretation S is an *answer set* of a program P iff S^+ is the least model[2] of the program $P^+ = \{r^+ \mid S \models (body(r))^-\}$. Note that an answer set S is usually represented by S^+ (this convention is sometimes used also in this paper).

The set of all answer sets of a program P is denoted by $AS(P)$. A program is called *coherent* iff it has an answer set.

A strict partial order is a binary relation, which is irreflexive, transitive and, consequently, asymmetric.

A *prioritized logic program* is defined in this paper as a pair $(P, \prec)$, where P is a program and $\prec$ a strict partial order on rules of P. Let be $r_1, r_2 \in P$. If $r_1 \prec r_2$ it is said that r_2 is more preferred than r_1.

3 Argumentation Structures

Our aim is to transfer a preference relation defined on rules to a preference relation on answer sets and, finally, to a notion of preferred answer sets. To that end argumentation structures are introduced. The basic argumentation structures correspond to rules. Some more general types of argumentation structures are derived from the basic argumentation structures. A special type of argumentation structures corresponds to answer sets.

Definition 1 ($\ll_P$, [11]). *An objective literal L* depends *on a set of default literals $W \subseteq Def$* with respect to a program P *($L \ll_P W$) iff there is a sequence of rules $\langle r_1, \ldots, r_k \rangle$, $k \geq 1$, $r_i \in P$ such that*

- *$head(r_k) = L$,*
- *$W \models body(r_1)$,*
- *for each i, $1 \leq i < k$, $W \cup \{head(r_1), \ldots, head(r_i)\} \models body(r_{i+1})$.*

The set $\{L \in Lit \mid L \ll_P W\} \cup W$ is denoted by $Cn_{\ll_P}(W)$.[3]

[2] P^+ is treated as definite logic program, i.e., each objective literal of the form $\neg A$, where $A \in At$, is considered as a new atom.

[3] $Cn_{\ll_P}(W)$ could be defined as $T_P^\omega(W)$ and $L \ll_P W$ as $L \in T_P^\omega(W)$.

$W \subseteq Def$ is self-consistent *w.r.t. a program P iff $Cn_{\ll_P}(W)$ is consistent.* □

If $Z \subseteq Obj$, we will sometimes use the notation $Cn_{\ll_{P \cup Z}}(W)$, assuming that the program P is extended by the set of facts Z.

Definition 2 (Dependency structure). *Let P be a program.*

A self-consistent set $X \subseteq Def$ is called an argument *w.r.t. P for a consistent set of objective literals Y, given a set of objective literals Z iff*

1. $pos(X) \cap Z = \emptyset$,
2. $Y \subseteq Cn_{\ll_{P \cup Z}}(X)$.

We will use the notation $\langle Y \hookleftarrow X; Z \rangle$ and the triple denoted by it is called a dependency structure *(w.r.t. P).*[4] □

If $Z = \emptyset$ also a shortened notation $\langle Y \hookleftarrow X \rangle$ can be used. We will sometimes omit the phrase "w.r.t. P" and speak simply about dependency structures and arguments, if the corresponding program is clear from the context.

We are going to define basic argumentation structures, while using the same notation as for dependency structures. It is justified by Proposition 1., saying that basic argumentation structures comply with Definition 2 of dependency structures, if some conditions are satisfied.

Definition 3 (Basic argumentation structure). *Let $r \in P$ be a rule such that*

- $(body(r))^-$ *is self-consistent and*
- $pos((body(r))^-) \cap (body(r))^+ = \emptyset$.

Then $\mathcal{A} = \langle \{head(r)\} \hookleftarrow (body(r))^-; (body(r))^+ \rangle$ is called a basic *argumentation structure.* □

Proposition 1. *Each basic argumentation structure is a dependency structure.*

Proof. Let $\mathcal{A} = \langle \{head(r)\} \hookleftarrow (body(r))^-; (body(r))^+ \rangle$ be a basic argumentation structure for a rule $r \in P$. We show that $\{head(r)\} \subseteq Cn_{\ll_{P \cup (body(r))^+}}((body(r))^-)$.

Assume the program $P \cup (body(r))^+$. Let $(body(r))^+ = \{L_1, \ldots, L_k\}$. We introduce a new rule $r_{L_i} = L_i \leftarrow$ for every $L_i \in (body(r))^+$. Then we create a sequence of rules $\langle r_1, r_2, \ldots, r_n \rangle$ such that

- $n = |(body(r))^+| + 1$,
- $r_n = r$,
- $r_i = r_{L_i}$ where $L_i \in (body(r))^+$, for $1 \leq i < n$,
- $r_i \neq r_j$ for $1 \leq i, j < n$ and $i \neq j$.

[4] This notation does not refer to P explicitly, but the condition $Y \subseteq Cn_{\ll_{P \cup Z}}(X)$ relates a dependency structure to P. Moreover, we will use only a kind of dependency structures, called argumentation structures, derived from a given program P.

This sequence satisfies conditions from Definition 1 for assumption $(body(r))^-$, hence $head(r) \in Cn_{\ll_{P \cup (body(r))^+}}((body(r))^-)$. That is, we have that $\mathcal{A}$ is a dependency structure. □

We emphasize that only *self-consistent* arguments for *consistent* sets of objective literals are considered in this paper. Hence, programs as $P = \{p \leftarrow not\ p\}$ or $Q = \{p \leftarrow not\ q; \neg p \leftarrow not\ q\}$ are irrelevant for our constructions.

Some dependency structures can be derived from the basic argumentation structures. Only the dependency structures derived from the basic argumentation structures using derivation rules from Definition 4 are of interest in the rest of this paper. We will use the term *argumentation structure* for dependency structures derived from basic argumentation structures using derivation rules.

Derivation rules are motivated later in Example 1.

Definition 4 (Derivation rules and argumentation structures). *Let P be a program. An argumentation structure is inductively defined as follows. Each basic argumentation structure is an argumentation structure.*

Other argumentation structures are obtained using derivation rules R1, R2, and R3:

R1 (Unfolding) *Let $r_1, r_2 \in P$, $\mathcal{A}_1 = \langle \{head(r_1)\} \hookleftarrow X_1; Z_1 \rangle$ and $\mathcal{A}_2 = \langle \{head(r_2)\} \hookleftarrow (body(r_2))^-; (body(r_2))^+ \rangle$ be argumentation structures, $head(r_2) \in Z_1$, $X_1 \cup (body(r_2))^- \cup Z_1 \cup (body(r_2))^+ \cup \{head(r_1)\}$ be consistent and $X_1 \cup (body(r_2))^-$ be self-consistent. Then also $\mathcal{A}_3 = \langle head(r_1) \hookleftarrow X_1 \cup (body(r_2))^-; (Z_1 \setminus \{head(r_2)\}) \cup (body(r_2))^+ \rangle$ is an argumentation structure. We also write $\mathcal{A}_3 = u(\mathcal{A}_1, \mathcal{A}_2)$.*

R2 *Let $\mathcal{A}_1 = \langle Y_1 \hookleftarrow X_1 \rangle$ and $\mathcal{A}_2 = \langle Y_2 \hookleftarrow X_2 \rangle$ be argumentation structures and $X_1 \cup X_2$ be self-consistent. Then $\mathcal{A}_3 = \langle Y_1 \cup Y_2 \hookleftarrow X_1 \cup X_2 \rangle$ is an argumentation structure. We also write $\mathcal{A}_3 = \mathcal{A}_1 \cup \mathcal{A}_2$.*

R3 *Let $\mathcal{A}_1 = \langle Y_1 \hookleftarrow X_1 \rangle$ be an argumentation structure and $W \subseteq Def$. If $X_1 \cup W$ is self-consistent, then $\mathcal{A}_2 = \langle Y_1 \hookleftarrow X_1 \cup W \rangle$ is an argumentation structure. We also write $\mathcal{A}_2 = \mathcal{A}_1 \cup W$.* □

Example 1 ([1]). Let the following program P be given as follows (P is used as a running example in this paper):

$$\begin{array}{ll} r_1 & b \leftarrow a, not\ \neg b \\ r_2 & \neg b \leftarrow not\ b \\ r_3 & a \leftarrow not\ \neg a. \end{array}$$

Suppose that $\prec = \{(r_2, r_1)\}$.

Consider the rule r_2. The default negation *not* b plays the role of a *defeasible argument*. If the argument can be consistently evaluated as true with respect to a program containing r_2, then also $\neg b$ can (and must) be evaluated as true.

As regards the rule r_1, default negation *not* $\neg b$ can be treated as an argument for b, if a is true, it is an example of a "conditional argument".

The following basic argumentation structures correspond to the rules of P: $\langle\{b\} \hookleftarrow \{not\ \neg b\};\{a\}\rangle$,$\langle\{\neg b\} \hookleftarrow \{not\ b\}\rangle$, $\langle\{a\} \hookleftarrow \{not\ \neg a\}\rangle$. Let us denote them by $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, respectively.

An example of a derived argumentation structure: $\mathcal{A}_3$ enables to "unfold" the condition a in $\mathcal{A}_1$, the resulting argumentation structure can be expressed as $\mathcal{A}_4 = u(\mathcal{A}_1, \mathcal{A}_3) = \langle\{b\} \hookleftarrow \{not\ \neg b, not\ \neg a\}\rangle$.

Similarly, $\mathcal{A}_5 = \mathcal{A}_3 \cup \mathcal{A}_4 = \langle\{a, b\} \hookleftarrow \{not\ \neg b, not\ \neg a\}\rangle$ can be derived from $\mathcal{A}_3$ and $\mathcal{A}_4$ using the rule R2.

Observe that some argumentation structures correspond to the answer sets. $\mathcal{A}_5$ corresponds to the answer set $\{a, b\}$ and $\mathcal{A}_6 = \langle\{a, \neg b\} \hookleftarrow \{not\ b, not\ \neg a\}\rangle$ to $\{a, \neg b\}$. Notice that $\mathcal{A}_6 = \mathcal{A}_2 \cup \mathcal{A}_3$. The attack relation enables to select the preferred answer set. This will be discussed later. □

Proposition 2. *Each argumentation structure is a dependency structure.*

Proof. We have to show that an application of R1, R2 and R3 preserves the properties of dependency structures.

R1 Since $S_1 = X_1 \cup (body(r_2))^- \cup Z_1 \cup (body(r_2))^+ \cup \{head(r_1)\}$ is consistent then $S_2 = X_1 \cup (body(r_2))^- \cup (Z_1 \setminus \{head(r_2)\}) \cup (body(r_2))^+ \subseteq S_1$ is also consistent. This means that $pos(X_1 \cup (body(r_2))^-) \cap ((Z_1 \setminus \{head(r_2)\}) \cup (body(r_2))^+) = \emptyset$.
Let $Q = P \cup (Z_1 \setminus \{head(r_2)\}) \cup (body(r_2))^+$ and $w = head(r_2) \leftarrow$.
From $head(r_2) \in Cn_{\ll_{P \cup (body(r_2))^+}}((body(r_2))^-)$ we have a sequence R_2 of rules, where $R_2 = \langle q_1, q_2, \ldots, q_m\rangle$, $m > 0$ and $q_m = r_2$.
From $head(r_1) \in Cn_{\ll_{P \cup Z_1}}(X_1)$ we have the sequence $R_1 = \langle p_1, p_2, \ldots, p_n\rangle$ where $n > 0$ and $p_n = r_1$. We assume there is at most one occurrence of w in R_1. Otherwise we can remove all but the leftmost one. Note that since $r_2 \in P$ there is a possibility to satisfy $body(r_1)$ in a different way from using w.
If $w \in R_1$ then we have $p_i = w$ for some $1 \leq i < n$. We construct the sequence $R_3 = \langle q_1, q_2, \ldots, q_m, p_1, p_2, \ldots, p_{i-1}, p_{i+1}, \ldots, p_n\rangle$. If $w \notin R_1$ we construct the sequence $R_3 = \langle q_1, q_2, \ldots, q_m, p_1, p_2, \ldots, p_n\rangle$. In both cases R_3 satisfy the conditions from Definition 1 for the assumption $X_1 \cup (body(r_2))^-$.

R2 The condition $pos(X) \cap Z = \emptyset$ is satisfied both for R2 and R3, because $Z = \emptyset$.
It is supposed that $Y_1 \hookleftarrow X_1$ and $Y_2 \hookleftarrow X_2$ are argumentation structures and $X_1 \cup X_2$ is self-consistent. We have to show that $Y_1 \cup Y_2 \subseteq Cn_P(X_1 \cup X_2)$. Let $L \in Y_1 \cup Y_2$. Then $L \in Cn_P(X_1)$ or $L \in Cn_P(X_2)$, hence $L \in Cn_P(X_1 \cup X_2)$.

R3 Now, we assume that $Y \hookleftarrow X$ is an argumentation structure, i.e., $Y \subseteq Cn_P(X)$ and $X \cup W$ is self-consistent. Clearly, $Cn_P(X) \subseteq Cn_P(X \cup W)$, hence $Y \subseteq Cn_P(X \cup W)$.

Definition 5 (Derivations). *A* derivation *of an argumentation structure $\mathcal{A}$ (w.r.t. P) is a minimal sequence $\langle\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k\rangle$ of argumentation structures (w.r.t. P) such that $\mathcal{A}_1$ is a basic argumentation structure, $\mathcal{A} = \mathcal{A}_k$, and each*

$\mathcal{A}_i$, $1 < i \leq k$, is either a basic argumentation structure or it is obtained by R1 or R2 or R3 from preceding argumentation structures.

An extraordinary attention is devoted to derivations of complete argumentation structures – they correspond to answer sets.

Definition 6 (Complete argumentation structures). *An argumentation structure $\langle Y \hookleftarrow X \rangle$ is called* complete *iff for each literal $L \in Obj$ it holds that $L \in Y$ or $not\ L \in X$.* □

A set of basic argumentation structures is *assigned* to an arbitrary program P.

Proposition 3. *A complete argumentation structure $\langle Y \hookleftarrow X \rangle$ is derived from a set of basic argumentation structures assigned to a program P iff $X \cup Y$ is an answer set of P.*

A proof is based on the method of [5] and on a correspondence between derivations of argumentation structures and $Cn_{\ll_P}(X)$.

We are interested in attacks against derivations of complete argumentation structures.

4 Attacks and Warranted Derivations

Our approach to preferred answer sets is based on a solution of conflicts between complete argumentation structures. We distinguish three steps towards that goal.

Contradictions between argumentation structures represent the elementary step.

Rule preference and contradictions between basic argumentation structures are used to form an *attack* relation on basic argumentation structures. Consider two basic argumentation structures $\mathcal{A}_1$ and $\mathcal{A}_2$. If $\mathcal{A}_1$ contradicts $\mathcal{A}_2$ and corresponds to a more preferred rule, then it *attacks* $\mathcal{A}_2$.

Attacks between derived argumentation structures depend on how argumentation structures are derived, see Example 3 below. Hence, we will introduce an *attack relation on derivations.* The notion of warranted and blocked complete argumentation structures and of preferred answer set is based on this basis.

Definition 7. *Consider the argumentation structures $\mathcal{A} = \langle Y_1 \hookleftarrow X_1; Z_1 \rangle$ and $\mathcal{B} = \langle Y_2 \hookleftarrow X_2; Z_2 \rangle$.*

If there is a literal $L \in Y_1$ such that $not\ L \in X_2$, it is said that the argument X_1 contradicts *the argument X_2 and the argumentation structure $\mathcal{A}$* contradicts *the argumentation structure $\mathcal{B}$.*

It is said that X_1 is a counterargument *to X_2.* □

The basic argumentation structures corresponding to the facts of the given program are not contradicted.

Let $r_1 = a \leftarrow$ be a fact and $not\ a \in (body(r_2))^-$. Then any $W \subseteq Def$ s.t. $(body(r_2))^- \subseteq W$ is not self-consistent and, therefore, it is not an argument.

Example 2. In Example 1, $\mathcal{A}_1$ contradicts $\mathcal{A}_2$ and $\mathcal{A}_2$ contradicts $\mathcal{A}_1$.

Only some counterarguments are interesting: the rule r_1 is more preferred than the rule r_2, therefore the counterargument of $\mathcal{A}_2$ against $\mathcal{A}_1$ should not be "effectual". We are going to introduce a notion of *attack* in order to denote "effectual" counterarguments. □

Similarly as for the case of argumentation structures, the basic attacks are defined first. A terminological convention: if $\mathcal{A}_1$ attacks $\mathcal{A}_2$, it is said that the pair $(\mathcal{A}_1, \mathcal{A}_2)$ is an attack.

Definition 8. *Let* r_1*,* r_2 *be rules, and* $\mathcal{A}_1 = \langle \{head(r_1)\} \hookleftarrow (body(r_1))^-; (body(r_1))^+ \rangle$ *and* $\mathcal{A}_2 = \langle \{head(r_2)\} \hookleftarrow (body(r_2))^-; (body(r_2))^+ \rangle$ *be basic argumentation structures such that* $r_2 \prec r_1$ *and* $\mathcal{A}_1$ *contradicts* $\mathcal{A}_2$*.*

Then $\mathcal{A}_1$ attacks $\mathcal{A}_2$ *and it is said that this attack is* basic*.* □

Next step could be to transfer basic attacks to attacks between derived argumentation structures. However, it is not a straightforward task. Example 3 shows our intuitions. An argumentation structure $\mathcal{B}$ attacks another argumentation structure $\mathcal{A}$ w.r.t. a derivation, but not w.r.t. another derivation.

Example 3. Let P be

r_1	$a \leftarrow not\ b$	r_3	$a \leftarrow not\ c$
r_2	$b \leftarrow not\ a$	r_4	$c \leftarrow b.$

$\prec = \{(r_1, r_2)\}$.

There are two answer sets of P: $S_1 = \{a\}$ and $S_2 = \{b, c\}$. The corresponding argumentation structures are $\mathcal{A} = \langle \{a\} \hookleftarrow \{not\ b, not\ c\} \rangle$ and $\mathcal{B} = \langle \{b, c\} \hookleftarrow \{not\ a\} \rangle$, respectively.

Let $\mathcal{A}_1$ be $\langle \{a\} \hookleftarrow \{not\ b\} \rangle$, $\mathcal{A}_2 = \langle \{b\} \hookleftarrow \{not\ a\} \rangle$, $\mathcal{A}_3 = \langle \{a\} \hookleftarrow \{not\ c\} \rangle$, $\mathcal{A}_4 = \langle \{c\} \hookleftarrow \emptyset; \{b\} \rangle$.

There are two derivations of $\mathcal{A}$: the sequences $\sigma_1 = \langle \mathcal{A}_1, \mathcal{A} \rangle$ and $\sigma_2 = \langle \mathcal{A}_3, \mathcal{A} \rangle$ (remind the minimality condition). They start from a basic argumentation structure and R3 is used.

On the other hand, there is only one[5] derivation of $\mathcal{B}$: $\tau = \langle \mathcal{A}_2, \mathcal{A}_4, \mathcal{B} \rangle$.

The only basic attack is $(\mathcal{A}_2, \mathcal{A}_1)$ ($\mathcal{A}_2$ attacks $\mathcal{A}_1$). Hence, it is intuitive to accept that τ attacks σ_1. However, there is no reason to consider σ_2 as attacked.

A rather credulous approach is accepted in this paper: if there is a derivation of a complete argumentation structure, which is not attacked, then the complete argumentation structure is preferred. However, this is only a rough idea, a more subtle solution is presented below. □

We are going to define attacks between derivations. It is a simple task, but not sufficient for our goals.

[5] If we abstract from the order of argumentation structures in the derivation. This does not influence the attack relation between derivations.

Definition 9. *Let σ be a derivation of an argumentation structure $\mathcal{A}$ and τ a derivation of an argumentation structure $\mathcal{B}$. Suppose that a basic argumentation structure $\mathcal{A}_1$ belongs to σ and a basic argumentation structure $\mathcal{B}_1$ belongs to τ.*

It is said that σ attacks *τ, if $(\mathcal{A}_1, \mathcal{B}_1)$ is a basic attack.*

It is obvious that a derivation σ may attack a derivation τ and τ may attack σ, i.e. mutual attacks are possible. Similarly cyclic attacks are possible.

We intend to define preferred answer sets in terms of preferred complete argumentation structures. A first approximation is to select complete argumentation structures with non-attacked derivations. However, we need to handle the case of mutual or cyclic attacks (i.e., to consider a kind of reinstatement). To this end we borrow a technique from abstract argumentation frameworks [6].

Definition 10. *Consider an argumentation framework (A, α), where A, the set of arguments, is the set of all derivations of all complete argumentation structures and α is the attack relation defined in Definition 9.*

A derivation σ of a complete argumentation structure $\mathcal{A}$ is acceptable *w.r.t. a set $S \subseteq A$ iff for each $\tau \in A$ s.t. $(\tau, \sigma) \in \alpha$ there is some $\sigma' \in S$ s.t. $(\sigma', \tau) \in \alpha$.*
□

Notice that acceptable derivations may be attacked by derivations of non-complete argumentation structures.

Fact

If there is a derivation σ of a complete $\mathcal{A}$, which is not attacked, then σ is acceptable w.r.t. the empty set of derivations.

Example 4. Let P be $\{r_1 : \ a \leftarrow not\ b;\ r_2 : \ b \leftarrow not\ a\}$. If $\prec = \emptyset$ then both the derivation of $\langle \{a\} \hookleftarrow \{not\ b\} \rangle$ and the derivation of $\langle \{b\} \hookleftarrow \{not\ a\} \rangle$ are acceptable w.r.t. the empty set of derivations.

Suppose that $r_1 \prec r_2$. Then $\langle \{b\} \hookleftarrow \{not\ a\} \rangle$ is acceptable w.r.t. the empty set of derivations, but there is no set S of derivations s.t. $\langle \{a\} \hookleftarrow \{not\ b\} \rangle$ is acceptable w.r.t. S.

Let R be $P \cup \{r_3 : c \leftarrow a;\ r_4 : \ d \leftarrow b\}$, $r_1 \prec r_2, r_4 \prec r_3$. Then each derivation σ of $\langle \{a, c\} \hookleftarrow \{not\ b, not\ d\} \rangle$ is acceptable w.r.t. $S = \{\sigma\}$ and each derivation τ of $\langle \{b, d\} \hookleftarrow \{not\ a, not\ c\} \rangle$ is acceptable w.r.t. $S = \{\tau\}$.

Example 5. Consider a program P:

r_1	$a_1 \leftarrow not\ a_3, not\ d_2$
r_2	$d_1 \leftarrow not\ a_3, not\ d_2$
r_3	$a_2 \leftarrow not\ a_1, not\ d_3$
r_4	$d_2 \leftarrow not\ a_1, not\ d_3$
r_5	$a_3 \leftarrow not\ a_2, not\ d_1$
r_6	$d_3 \leftarrow not\ a_2, not\ d_1$

$\prec = \{(r1, r4), (r3, r5), (r6, r2)\}$.

We have three complete argumentation structures:

$\mathcal{A}_1 = \langle \{a_1, d_1\} \hookleftarrow \{not\ a_3, not\ d_2\} \rangle$, $\mathcal{A}_2 = \langle \{a_2, d_2\} \hookleftarrow \{not\ a_1, not\ d_3\} \rangle$,
$\mathcal{A}_3 = \langle \{a_3, d_3\} \hookleftarrow \{not\ a_2, not\ d_1\} \rangle$.

We have a cycle of attacks. Each derivation of $\mathcal{A}_2$ attacks each derivation of $\mathcal{A}_1$, each derivation of $\mathcal{A}_3$ attacks each derivation of $\mathcal{A}_2$, each derivation of $\mathcal{A}_1$ attacks each derivation of $\mathcal{A}_3$.

Let $\sigma_1, \sigma_2, \sigma_3$ be derivations of $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, respectively. It holds that σ_1 is acceptable w.r.t. $\{\sigma_3\}$, σ_2 is acceptable w.r.t. $\{\sigma_1\}$, and σ_3 is acceptable w.r.t. $\{\sigma_2\}$. □

Definition 11 (Warranted and blocked argumentation structures). *Let $\mathcal{A}$ be a complete argumentation structure. If there is an acceptable derivation of $\mathcal{A}$ w.r.t. a set S of some derivations of some complete argumentation structures, then $\mathcal{A}$ is called* warranted*, otherwise it is called* blocked*.* □

5 Preferred Answer Sets

Definition 12 (Preferred answer set). *A complete argumentation structure is* preferred *iff it is warranted.*

$Y \cup X$ is a preferred answer set *iff $\langle Y \hookleftarrow X \rangle$ is a preferred argumentation structure.* □

Notice that our notion of preferred answer set is rather a credulous one.

Example 6. Consider our running Example 1, where we have complete argumentation structures $\mathcal{A}_5 = \langle \{b, a\} \hookleftarrow \{not\ \neg b, not\ \neg a\} \rangle, \mathcal{A}_6 = \langle \{\neg b, a\} \hookleftarrow \{not\ \neg a, not\ b\} \rangle$ and basic argumentation structures $\mathcal{A}_1 = \langle \{b\} \hookleftarrow \{not\ \neg b\}; \{a\} \rangle$, $\mathcal{A}_2 = \langle \{\neg b\} \hookleftarrow \{not\ b\} \rangle$, $\mathcal{A}_3 = \langle \{a\} \hookleftarrow \{not\ \neg a\} \rangle$.

The only basic attack is $(\mathcal{A}_1, \mathcal{A}_2)$, $\mathcal{A}_1$ attacks $\mathcal{A}_2$. Therefore, the derivation $\sigma = \langle \mathcal{A}_1, \mathcal{A}_3, \mathcal{A}_4 = \langle \{b\} \hookleftarrow \{not\ \neg b, not\ \neg a\} \rangle, \mathcal{A}_5 \rangle$ attacks $\tau = \langle \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_6 \rangle$.

There is no derivation of $\mathcal{A}_6$ which is not attacked by σ and no derivation of $\mathcal{A}_6$ counterattacks the derivation σ. $\mathcal{A}_6$ is blocked, on the other hand, $\mathcal{A}_5$ is warranted. Hence, we prefer $\mathcal{A}_5$ over $\mathcal{A}_6$.

Consequently, $\{a, b\}$ is a preferred answer set of the given prioritized logic program. □

The following example shows that the argumentation structure corresponding to the only answer set of a program is preferred, even if each its derivation is attacked by a derivation of an argumentation structure which is not complete. The example demonstrates also that attacks between derivations can not be implemented via conventional attacks on arguments. Anyway, a goal of our future research is to find a method how to minimize comparisons of derivations.

Example 7. Consider the program

r_1	$b \leftarrow not\ a$	r_3	$c \leftarrow a$
r_2	$a \leftarrow not\ b$	r_4	$c \leftarrow not\ c$

$\prec = \{(r_2, r_1)\}$.

Let the basic argumentation structures be denoted by $\mathcal{A}_i$, $i = 1, \ldots, 4$. $(\mathcal{A}_1, \mathcal{A}_2)$ is the only basic attack.

The derivation $\langle \mathcal{A}_1 \rangle$ attacks the derivation $\sigma = \langle \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_5, \mathcal{A}_6 \rangle$, where $\mathcal{A}_5 = \langle \{c\} \hookleftarrow \{not\ b\} \rangle$ and $\mathcal{A}_6 = \langle \{c, a\} \hookleftarrow \{not\ b\} \rangle$.

However, $\mathcal{A}_1$ is not a member of a derivation of a complete argumentation structure. Hence, σ is acceptable w..r.t. the empty set according to Definition 10. Therefore, the complete argumentation structure $\mathcal{A}_6$ is warranted and, consequently, it is the preferred argumentation structure. □

We distinguish between attacking and blocking. If an argumentation structure is blocked then there is no its derivation which counterattacks the attacks of derivations of other *complete* argumentation structures.

Theorem 1. *If S is a preferred answer set of $(P, \prec)$, then S is an answer set of P.*

Proof. If S is a preferred answer set then there is a preferred complete argumentation structure $\mathcal{A} = \langle S^+ \hookleftarrow S^- \rangle$. Hence, S is total. We have to show that $S^+ = Cn_{\ll_P}(S^-) \cap Obj$ using a result of [5].

Clearly, $S^+ \subseteq Cn_{\ll_P}(S^-)$ according to the definition of dependency structure. Let be $L \in Obj$ and $L \in Cn_{\ll_P}(S^-)$. It holds that S^- is self-consistent and S is total. Hence, $not\ L \notin S^-$ and $L \in S^+$. □

Our next goal is to evaluate the presented approach to preferred answer sets selection. To this end some principles and their (un)satisfaction are discussed in the next section.

6 Evaluation

We start with a discussion of principles proposed by [1]. A new principle requiring selection of a preferred answer set from the non-empty set of standard answer sets is added. After that it is proved that the new principle is satisfied by our approach. Finally, an informal and tentative proposal of some new principles, characterizing the descriptive approach to selection of preferred answer sets is presented.

6.1 Principles

The principles (partially) specify what it means that an order on answer sets corresponds to the given order on rules. Let us start with principles proposed in [1] for arbitrary prioritized theories.

Principle I.
Let B_1 and B_2 be two belief sets of a prioritized propositional theory $(T; \prec)$ generated by the rules $R \cup \{d_1\}$ and $R \cup \{d_2\}$, where $d_1, d_2 \notin R$, respectively. If d_1 is preferred over d_2, then B_2 is not a (maximally) preferred belief set of T. □

Principle II.
Let B be a preferred belief set of a prioritized propositional theory $(T; \prec)$ and r a rule such that at least one prerequisite of r is not in B. Then B is a preferred belief set of $(T \cup \{r\}; \prec')$ whenever $\prec'$ agrees with $\prec$ on priorities among rules in T. □

We believe that the possibility to select always a preferred answer set from a non-empty set of standard answer sets is of critical importance. Principle III, accepted in this paper, reproduces the idea of Proposition 6.1 from [1].

Principle III.
Let $\mathcal{B} \neq \emptyset$ be the set of all belief sets of a prioritized theory $(T, \prec)$. Then there is a selection function Σ s.t. $\Sigma(\mathcal{B})$ is the set of all preferred belief sets of $(T, \prec)$, where $\emptyset \neq \Sigma(\mathcal{B}) \subseteq \mathcal{B}$. □

We consider and discuss below only a specific case of prioritized theories, prioritized logic programs. Principle I specifies an attack of a belief set B_1 against a belief set B_2. The attack is based on the preference of the rule d_1 over the rule d_2 (they cannot be applied together for generating a preferred answer set). But Principle I is not appropriate for an approach which considers mutual attacks of preferred answer sets and its main goal is to select at least one preferred answer set – existence of an attack against a candidate for a preferred answer set is not sufficient for its elimination. The attacked answer set can be defended by a counterattack of another answer set. In order to summarize, Principle I is not appropriate for an approach which distinguishes between attacking and blocking.

It was shown in [1], Proposition 6.1, that Principle II is incompatible with the existence of a function which selects a non-empty set of preferred answer sets from a non-empty set of standard answer sets of a given logic program, if the notion of preferred answer set from [1] is accepted. Moreover, we have a basic problem with this principle. First an example.

Example 8 ([1]). Suppose that we accept both Principle II and Principle III.

Consider program P, whose single standard answer set is $S = \{b\}$ and the rule (1) is preferred over the rule (2).

$$c \leftarrow not\ b \tag{1}$$

$$b \leftarrow not\ a \tag{2}$$

S is not a preferred answer set in the framework of [1].

Assume that S, the only standard answer set of P, is selected – according to Principle III – as the preferred answer set of $(P, \prec)$.[6] Let P' be $P \cup \{a \leftarrow c\}$ and $a \leftarrow c$ be preferred over the both rules (1) and (2). P' has two standard answer sets, S and $T = \{a, c\}$.

Note that $\{c\} \not\subseteq S^+$. Hence, S should be the preferred answer set of P' according to the Principle II . However, in the framework of [1] the only preferred answer set of $(P', \prec')$ is T. This selection of preferred answer set satisfies clear

[6] Observe that the only derived complete argumentation structure is $\langle \{b\} \hookleftarrow \{not\ a, not\ c\} \rangle$. Hence, $\{b\}$ is a preferred answer set in our framework.

intuitions – T is generated by the two most preferred rules. A consequence, accepted in]1] is that Principle III is refused.

In our approach the complete argumentation structure $\langle\{a, c\} \hookleftarrow \{not\ b\}\rangle$ is preferred and $\{a, c\}$ is the preferred answer set of P'.

Principle III is of crucial value according to our view, therefore we do not accept Principle II. This example is not the only reason for it. A more fundamental reason is expressed in Sect. 6.2 as a principle called *Nonmonotonicity of selection constraints*. We selected in this example preferred answer sets of P' from a broader variety of possibilities. Consequently, a selection of a preferred answer set from the extended set of possibilities should not be limited to a subset of those possibilities.

A more detailed justification of our decision not to accept Principle II is presented in [12]. □

Principle II is not accepted also in [9]. According to [4] descriptive approaches do not satisfy this principle in general.

In the rest of this subsection satisfaction of the Principle III (more precisely, its specialization for prioritized logic programs) is proved.

Lemma 1. *The attack relation between derivations of complete argumentation structures is irreflexive.* □

Proof. Let $\sigma = \langle \mathcal{A}_1, \dots, \mathcal{A}_k \rangle$ be a derivation of a complete argumentation structure $\mathcal{A}_k$. Suppose to the contrary that σ attacks itself, i.e., there are basic argumentation structures $\mathcal{A}_i, \mathcal{A}_j$ s.t. $\mathcal{A}_i = \langle\{head(r)\} \hookleftarrow \{body^-(r)\}; \{body^+(r)\}\rangle$ attacks $\mathcal{A}_j = \langle\{head(q)\} \hookleftarrow \{body^-(q)\}; \{body^+(q)\}\rangle$, where r, q are rules. It follows that $not\ head(r) \in body^- q$. Contradiction: $\mathcal{A}_k$ is consistent and σ is a minimal sequence with the last member $\mathcal{A}_k$. □

Theorem 2. *Principle III is satisfied.*

Let $\mathcal{P} = (P, \prec)$ be a prioritized logic program and $AS(P) \neq \emptyset$. Then there is a preferred answer set of $\mathcal{P}$ in our approach.

Proof. **Case 1** Assume that P has only one answer set S. if there is only one derivation of $\mathcal{A} = \langle S^+ \hookleftarrow S^- \rangle$, then no complete argumentation structure blocks it (from Lemma 1.). If there are more derivations of $\langle S^+ \hookleftarrow S^- \rangle$, then the argument from the proof of the lemma is applied: $\mathcal{A}$ is consistent and all derivations are minimal sequences with the last member $\mathcal{A}$.

Case 2 Suppose that P has only two answer sets S_1 and S_2. Let the corresponding complete argumentation structures be $\mathcal{A}_1 = \langle S_1^+ \hookleftarrow S_1^- \rangle$ and $\mathcal{A}_2 = \langle S_2^+ \hookleftarrow S_2^- \rangle$, respectively.

Without loss of generality assume that there is a derivation of $\mathcal{A}_1$ which is not attacked by a derivation of $\mathcal{A}_2$. Then P has at least one preferred answer set.

Suppose now that each derivation of $\mathcal{A}_1$ is attacked by a derivation of $\mathcal{A}_2$ and vice versa. Consider a derivation σ of $\mathcal{A}_1$. Let $\{\tau_1, \dots, \tau_k\}$ be the set of all derivations of $\mathcal{A}_2$ attacking σ. Recall that each τ_i is attacked by a derivation of

$\mathcal{A}_1$. Let S be the set of all derivations of $\mathcal{A}_1$ attacking at least one τ_i. It holds that σ is acceptable w.r.t. S, hence $\mathcal{A}_1$ is warranted.

Case 3 Let be $AS(P) = \{S_1, \ldots, S_k\}, k \geq 3$. Assume that the corresponding complete argumentation structures are $\mathcal{A}_i, i = 1, \ldots, k$.

If there is a derivation of some $\mathcal{A}_i$, which is not attacked, then the corresponding answer set is preferred.

Otherwise, all derivations of all complete argumentation structures are attacked by a derivation of a complete argumentation structure. By a generalization of the argument of Case 2 we have that each derivation of each complete argumentation structure is defended by a set of derivations. □

6.2 Discussion – Descriptive Approach

Finally, a discussion of a tentative proposal of some possible principles appropriate for a descriptive approach to preferred answer sets selection is presented. The principles are expressed in a more or less informal way and represent a very preliminary attempt. All the principles are inspired by our definitions and constructions, but they are not intended solely for the framework presented in this paper. A general and more detailed discussion of the postulates is postponed for a future paper.

The following principle represents a more careful, but less deep version of Principle I:

Principle – Defeated answer sets.
Let S_1, S_2 be answer sets of a program P, and let S_2 be not defeated. If S_1 is defeated by S_2, then S_1 cannot be a preferred answer set.[7] □

The principle above and the following one can be considered as expressing a little bit more accurately the main intuitive idea of our stance w.r.t. descriptive approach to preferred answer sets selection: it is desirable to apply all rules of an undefeated set of generating rules. There is a difference between attacking and blocking (defeating).

Below is the other side of this intuitive idea: rules can be blocked by more preferred rules but other rules are handled in a declarative style.

Principle of blocking.
If a standard answer set A is generated by a set R of rules, where no rule is attacked by a more preferred rule then A is a preferred answer set. □

Next principle is inspired by the problem of Example 8. The problem was as follows: a program P with a set $\mathcal{S}$ of answer sets was given together with a program R s.t. $P \subset R$. $\mathcal{M} \neq \mathcal{S}$ is the set of all answer sets of R. According to our view conditions expressed for a selection of preferred members of $\mathcal{S}$ may not constrain a selection of preferred members of the different $\mathcal{M}$. If we select

[7] Of course, there are different possible ways how to specify the notion of defeat. A definition of defeated generating sets of rules can be obtained in a straightforward way from the notion of defeat presented in this paper.

a preferred answer set from $\mathcal{M}$ then we can not limit (constrain) the selection to $\mathcal{S}$.

Principle – Nonmonotonicity of selection constraints.
If $P \subset R$ are programs, then a selection of preferred answer sets of R should not be limited to the set of preferred answer sets of P. □

Attacks of rules, which do not contribute to a generation of a standard answer set, are irrelevant w.r.t. a selection of preferred answer sets:

Principle – Irrelevant attacks.
Let $r_1, r_2 \in P$, r_1 attacks r_2, but r_1 is not a member of a set of generating rules of a standard answer set and $r_2 \in R$, where R is a set of rules generating a standard answer set A.

If there is no other attack against rules of R, then A is a preferred answer set. □

As regards a choice of principles, we accept the position of [1]: even if somebody does not accept a set of principles for preferential reasoning, those (and similar) principles are still of interest as they may be used for classifying different patterns of reasoning.

Of course, some of principles proposed in this subsection may be refused, or some new may be suggested. Different sets of such principles provide different conceptions of a descriptive approach to preferred answer sets selection.

Finally, notice that our descriptive approach can be expressed without argumentation structures using a translation to generating sets of rules.

7 Related Work

D-preference [2], W-preference [18], and B-preference [1]. D/W/B-preferences are representatives of prescriptive approaches. They are based on the view that preference specifies the order in which rules have to be applied. A preferred rule is forced to be applied first. If a more preferred rule has in its body a literal, which is the head of a less preferred rule, then the more preferred rule is not applicable. As a consequence, there are programs with standard answer sets, but without preferred answer sets (hence, Principle III is not satisfied in those approaches).

Our approach enables to select at least one preferred answer set from the non-empty set of standard answer sets of a program. Not all preferences are effective, i.e. not all preferences are transformed to attacks between derivations of argumentation structures.

Therefore D/W/B-preferences are not in direct hierarchic (subset) relation to our semantics.

A fundamental difference between our approach and D/W/B-preference is that testing D/W/B-preference is local. When testing whether an answer set X is preferred, it is not needed to know other answer sets. The computational complexity of those approaches remains within NP. On the other hand, in our approach, all the attacks between derivations of all complete argumentation structures (they correspond to answer sets) are considered. Hence, our conjecture is that the decision problem, whether a complete argumentation structure (an answer set) is a preferred one, is in our approach beyond the class NP.

Sakama and Inoue [9]. Sakama and Inoue have defined an approach that selects preferred answer sets given the preference on literals. A preference relation on literals is transferred to a preference relation on sets of literals. Preferred answer sets are then the maximal (with respect to a preference relation) answer sets. They also provide a way to transform preference on rules to preference on literals. However, structure of the rules, i.e. which rule is blocked by which rule, is not considered during the transformation.

Wakaki [17]. Wakaki has extended Dung's abstract argumentation framework in order to work with preferences. She has introduced preference relation on arguments. Selection of a preferred extension (in a sense of preference on arguments) is done in a similar manner that Sakama and Inoue use to select preferred answer sets. Wakaki then defines a non-abstract logic programming based argumentation framework. Rules of a logic program are transformed to arguments. Preference on literals is transferred to preference on arguments via heads of rules. Wakaki's and our argumentation framework are principally different. Wakaki's goal is to extend Dung's abstract argumentation framework. When selecting a preferred extension in an abstract framework, there is no information about the structure of arguments. On the other hand, approaches for preference handling that work with preference on rules depend on the structure of the rules. The non-abstract argumentation framework proposed by Wakaki deals with preference on literals, which we do not address by our framework. Just to note, Wakaki's non-abstract framework is equivalent with Sakama and Inoue's approach to preference on literals.

Gabaldon [7]. Gabaldon works with extended logic programs and preference (called priority) on rules. His goal it to develop a semantics that always selects (i) a preferred answer set when a standard one exists, and (ii) the only preferred answer set for fully prioritized programs when a standard one exists. The semantics is defined in three steps. First, a partially prioritized program is fully prioritized. Second, a program is transformed to a prerequisite-free program using the unfolding operator. Third, a test is defined to test whether an answer set is preferred. The rules of a program are applied one at the time. A rule is applicable if all its prerequisites were already derived. The order, in which rules are applied, does not have to correspond to priorities. Priorities are used when there are rules with satisfied prerequisites that block each other via default assumptions. Then the preferred rule is used. An answer set is preferred if it can be generated in the aforementioned way.

The main difference between Gabaldon's and our approach is that Gabaldon's semantics does not satisfy Principle III. It guarantees existence of a preferred answer set when a standard one exists only for a subclass of programs (head-cycle-free and head-consistent). We guarantee it for every logic program. The considered subclass comprises the programs without integrity constraints that are encoded as rules that form a negative odd cycle. Gabaldon motivates this focus by complexity concerns. If integrity constraints are allowed we need to know whether there are other preferred answer sets when testing whether an answer set is a preferred one. An answer set that should be preferred from the view of preference can be ruled out by an integrity constraint. A deeper analysis of the relation between our and Gabaldon's semantics is a subject of our future mutual cooperation.

8 Conclusions

An argumentation framework has been constructed, which enables transferring attacks of rules to attacks between derivations of argumentation structures and, consequently, to warranted complete argumentation structures. Preferred answer sets correspond to warranted complete argumentation structures. This construction enables a selection of a preferred answer set whenever there is a non-empty set of standard answer sets of a program. This feature is paid by an increasing computational complexity. The representative approaches based on the prescriptive approach remain in the class NP, but our approach is beyond that class.

We did not accept the second principle from [1] and we needed to modify their first principle.

Among goals for our future research are a development of the set of principles, characterizing a descriptive approach and a continuation of our approach without the transfer to argumentation structures. A consideration of attacks between generating sets of rules represents a natural solution. Preliminary results of this research are published in [14,15] and also in [16]. A more detailed comparison of our approach(es) to other approaches is needed. Also approaches not referenced in this paper are of interest.

Finally, we have to mention the main differences between the preliminary version [12], the version presented at WLP 2011 [13], and this paper.

Both in [12,13] were used attack derivation rules. They were proposed in order to derive attacks between argumentation structures from the basic attacks. However, we did not find a proper version of the rules. The rules of [12] were too liberal, they did not derive all intuitive attacks and, consequently, the set of preferred answer sets was too broad. Moreover, a dependency of attacks against argumentation structures on derivations of argumentation structures was not explicitly stated. As regards attack derivation in [13], a more subtle set of derivation rules is introduced, a superset of attacks was derivable and attacks of argumentation structures were explicitly relativized to derivations of argumentation structures. However, derivation rules Q2 and Q3 were sensitive to arbitrary attacks and, as a consequence, they did not ignore irrelevant attacks (in the sense of a Principle in Sect. 6.2).

A claim that Principle I holds, was in both papers, the proof in [13] was not correct. Principle I does not hold in the current paper and we consider this as an important feature of our descriptive approach.

Attack derivation rules are omitted in this paper. Attacks between derivations of argumentation structures were defined directly. An important new notion is an acceptable derivation. The notion enables a correct handling of mutual and cyclic attacks and a clear characterization of a difference between attacking and blocking. Omitting of attack derivation rules simplifies our approach, enables a more clear, more transparent exposition of our semantics and a more reliable characterizations of its properties.

Acknowledgements. We are grateful to anonymous referees for careful reading, for very valuable, detailed and helpful comments and proposals. This paper was supported by the grants 1/0689/10 and 1/1333/12 of VEGA.

References

1. Brewka, G., Eiter, T.: Preferred answer sets for extended logic programs. Artif. Intell. **109**(1–2), 297–356 (1999)
2. Delgrande, J., Schaub, T., Tompits, H.: A framework for compiling preferences in logic programs. Theory Pract. Logic Program. **3**(2), 129–187 (2003)
3. Delgrande, J., Schaub, T.: Expressing preferences in default logic. Artif. Intell. **123**(1–2), 41–87 (2000)
4. Delgrande, J., Schaub, T., Tompits, H., Wang, K.: A classification and survey of preference handling approaches in nonmonotonic reasoning. Comput. Intell. **20**(2), 308–334 (2004)
5. Dimopoulos, Y., Torres, A.: Graph theoretical structures in logic programs and default theories. Theoret. Comput. Sci. **170**(1–2), 209–244 (1996)
6. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**, 321–357 (1995)
7. Gabaldon, A.: A selective semantics for logic programs with preferences. In: Proceedings of the International Workshop on Nonmonotonic Reasoning, Action and Change (NRAC 11), Barcelona, Spain (2011)
8. García, A.J., Simari, G.R.: Defeasible logic programming: an argumentative approach. Theory Pract. Logic Program. **4**(1–2), 95–138 (2004)
9. Sakama, C., Inoue, K.: Prioritized logic programming and its application to commonsense reasoning. Artif. Intell. **3**(1–2), 185–222 (2000)
10. Schaub, T., Wang, K.: A comparative study of logic programs with preference. In: IJCAI 2001, pp. 597–602 (2001)
11. Šefránek, J.: Rethinking semantics of dynamic logic programs. In: Proceedings of the Workshop NMR (2006)
12. Šefránek, J.: Preferred answer sets supported by arguments. In: Proceedings of the Workshop NMR (2008)
13. Šefránek, J., Šimko, A.: Warranted derivation of preferred answer sets. In: Proceedings of WLP (2011)
14. Šimko, A.: Preferred answer sets - banned generating set. In: Proceedings of the Student Conference, Bratislava 2011, Faculty of Mathematics, Physics and Informatics, Comenius University, pp. 326–333 (2011)

15. Šimko, A.: Selection of a preferred answer sets as a decision between generating sets. In: 6th Workshop on Intelligent and Knowledge Oriented Technologies, pp. 51–56 (2011)
16. Šimko, A.: Accepting the natural order of rules in a logic program with preferences. In: Proceedings of the Doctoral Consortium at ICLP 2011 (2011)
17. Wakaki, T.: Preference-based argumentation handling dynamic preferences built on prioritized logic programming. In: Kinny, D., Hsu, J.-I., Governatori, G., Ghose, A.K. (eds.) PRIMA 2011. LNCS, vol. 7047, pp. 336–348. Springer, Heidelberg (2011)
18. Wang, K., Zhou, L.-Z., Lin, F.: Alternating fixpoint theory for logic programs with priority. In: Palamidessi, C., et al. (eds.) CL 2000. LNCS (LNAI), vol. 1861, p. 164. Springer, Heidelberg (2000)