Technical Report

TU Dortmund, Faculty of Computer Science

Chair VI, Information Engineering Group, February 2008

On the Relationship of Defeasible Argumentation and Answer Set Programming (Extended Version)

Matthias Thimm and Gabriele Kern-Isberner

Technische Universität Dortmund

Abstract. This paper investigates the relationship between defeasible argumentation (DeLP) and answer set programming by transforming a defeasible logic program into an answer set program. We propose two types of conversions that differ with respect to the handling of strict rules. Inference via a dialectical warrant procedure in DeLP turns out to be stronger than credulous answer set inference in both cases, while conversions of the second type bring DeLP inference closer to skeptical answer set inference. Moreover, we investigate some characteristics of the warrant procedure of DeLP which lead to a better understanding of the notion of warrant.

Keywords: Argumentation, defeasible logic programming, answer set programming

1 Introduction

Defeasible Argumentation [8], as proposed with the language DeLP (*Defeasible Logic Programming*) by García and Simari in [6] is an approach for logical argumentative reasoning [1, 9] based on defeasible logic. In DeLP the belief in literals is supported by arguments and in order to handle conflicting information a warrant procedure decides which information has the strongest grounds to believe in. In this way, the notion of warrant induces a nonmonotonic inference relation between a defeasible logic program (consisting of facts as well as strict and defeasible rules) and literals. The exploration of this inference relation in terms of answer set semantics is the topic of this paper.

Indeed, the relationships between defeasible argumentation and other default reasoning systems, especially the relationship of their particular inference mechanisms, have been investigated only little so far. While in [5] default logic and logic programming are characterized as instantiations of Dung's abstract argumentation framework we are interested in a direct relation between default logic and DeLP which can also be characterized as an instantiation of an abstract argumentation framework. In [4] the relationship of DeLP with Reiters default logic [10] is investigated by converting a default logic program into a defeasible logic program and applying the warrant procedure to determine the extensions of the original default logic program. In that paper, a special case of DeLP programs is used, so that the warrant of a literal is equivalent to the sceptical inference of that literal.

In this paper we take the converse point of view by translating a defeasible logic program into an answer set program (ASP) [7] and applying answer set techniques to determine the warranted literals of the original defeasible logic program. First, we investigate some characteristics of the warrant procedure of DeLP which leads to a better understanding of the notion of warrant. As DeLP reasoning is paraconsistent, the handling of inconsistencies under the translation is of major importance. We will propose two approaches to converting a defeasible logic program into an answer set program, dealing with inconsistencies in different ways. The first conversion method respects the substantial difference between strict and defeasible rules but has to take inconsistencies brought about by strict rules into account; the resulting warrant semantics is shown to be weaker than skeptical ASP semantics, but stronger than credulous ASP semantics. The other type of conversion blurs the distinction between strict and default rules and yields better results in computing warrant through answer set techniques. More precisely, we show that all warranted literals are contained in one answer set of the corresponding logic program. In particular, if the preference relation between arguments is empty (so that defeating is reduced to attacking), then inference by a warrant procedure turns out to be even stronger than skeptical inference.

In contrast to [2] this paper does not aim at fixing DeLP regarding some observed flaws in its inference mechanism; instead, we will interpret the original DeLP inference mechanism via answer set semantics.

This paper is structured as follows: in Section 2 and 3 brief overviews over ASP and defeasible logic programming are given. Section 4 investigates the notion of warrant in detail. Section 5 and 6 propose two alternatives of converting a DeLP-program into an answer set program and discuss the results. In Section 7 we conclude. All proofs can be found in the appendix.

2 Answer set programming

In this section we give a brief overview over answer set programming and answer sets as proposed by Gelfond and Lifschitz in [7]. We consider extended logic programs, which distinguish between classical and default negation.

We use a first-order language without function symbols except constants, so let $\mathfrak{L}$ be a set of literals, where a literal h is an atom A or a (classical) negated atom $\neg A$. The symbol $\bar{}$ will be used to denote the complement of a literal with respect to classical negation, i. e. it is $\overline{p} = \neg p$ and $\overline{\neg p} = p$ for a ground atom p.

Definition 1 (Extended logic program). *An* extended logic program P *is a finite set of rules of the form* $r : h \leftarrow a_1, \ldots, a_n, \textsf{not } b_1, \ldots, \textsf{not } b_m$ *where* $h, a_1, \ldots, a_n, b_1, \ldots, b_m \in \mathfrak{L}$. *We denote by* $head(r)$ *the head* h *of the rule* r *and by* $body(r)$ *the body* $\{a_1, \ldots, a_n, \textsf{not } b_1, \ldots, \textsf{not } b_m\}$ *of the rule* r.

If the body of a rule r is empty ($body(r) = \emptyset$), then r is called a *fact*, abbreviated h instead of $h \leftarrow$.

Given a set $X \subseteq \mathfrak{L}$ of literals, then r is *applicable* in X, iff $a_1, \ldots, a_n \in X$ and $b_1, \ldots, b_m \notin X$. The rule r is *satisfied* by X, if $h \in X$ or if r is not applicable in X. X is a model of an extended logic program p iff all rules of P are statisfied by X. The set $X \subseteq \mathfrak{L}$ is *consistent*, iff for every $h \in X$ it is not the case that $\overline{h} \in X$. An answer set is a minimal consistent set of literals that satisfies all rules. This can be characterized as follows.

Definition 2 (Reduct). *Let* P *be an extended logic program and* $X \subseteq \mathfrak{L}$ *a set of literals. The* X-reduct *of* P, *denoted* P^X, *is the union of all rules* $h \leftarrow a_1, \ldots, a_n$ *such that* $h \leftarrow a_1, \ldots, a_n, \textsf{not } b_1, \ldots, \textsf{not } b_m \in P$ *and* $X \cap \{b_1, \ldots, b_m\} = \emptyset$.

For any extended logic program P and a set X of literals, the X-reduct of P is a logic program P' without default-negation and therefore has a minimal model. If P' is inconsistent, then its unique model is defined to be $\mathfrak{L}$.

Definition 3 (Answer set). *Let* P *be an extended logic program. A consistent set of literals* $S \subseteq \mathfrak{L}$ *is an* answer set *of* P, *iff* S *is a minimal model of* P^S.

3 Defeasible Logic Programming

Defeasible Logic Programming (DeLP) [6] is a logic programming language which is capable of modelling defeasible knowledge. With the use of a defeasible argumentation process it is possible to derive conclusive knowledge.

The basic elements of DeLP are facts and rules. The set of rules is divided into strict rules, i. e. rules which derive certain knowledge, and defeasible rules, i. e. rules which derive uncertain or defeasible knowledge. We use the same set $\mathfrak{L}$ of literals as in Section 2 to define the elements of a DeLP-program.

Definition 4 (Fact, strict rule, defeasible rule). *A* fact *is a literal* $h \in \mathfrak{L}$. *A* strict rule *is an ordered pair* $h \leftarrow B$, *where* $h \in \mathfrak{L}$ *and* $B \subseteq \mathfrak{L}$. *A* defeasible rule *is an ordered pair* $h \prec B$, *where* $h \in \mathfrak{L}$ *and* $B \subseteq \mathfrak{L}$.

Syntactically, the symbol "$\prec$" is all that distinguishes a defeasible rule from a strict rule. Pragmatically, a defeasible rule is used to describe uncertain knowledge as in "birds fly". As in ASP we use the functions $body/1$ and $head/1$ to refer to the head resp. body of a defeasible or strict rule.

Definition 5 (Defeasible Logic Program). *A* Defeasible Logic Program $\mathcal{P} = (\mathit{\Pi}, \mathit{\Delta})$*, abbreviated* de.l.p.*, consists of a (possibly infinite) set* $\mathit{\Pi}$ *of facts and strict rules and of a (possibly infinite) set* $\mathit{\Delta}$ *of defeasible rules.*

Example 1 ([6], example 2.1). Let $\mathcal{P} = (\mathit{\Pi}, \mathit{\Delta})$ be given by

$$\mathit{\Pi} = \left\{ \begin{array}{ll} chicken(tina) & scared(tina) \\ penguin(tweety) & (bird(X) \leftarrow chicken(X)) \\ bird(X) \leftarrow penguin(X)) & (\neg flies(X) \leftarrow penguin(X) \end{array} \right\} ,$$

$$\mathit{\Delta} = \left\{ \begin{array}{l} flies(X) \prec bird(X) \\ \neg flies(X) \prec chicken(X) \\ flies(X) \prec chicken(X), scared(X) \\ nests\_in\_trees(X) \prec flies(X) \end{array} \right\} .$$

The program $\mathcal{P}$ contains the facts, that Tina is a scared chicken and that Tweety is penguin. The strict rules state that all chickens and all penguins are birds and penguins cannot fly. The defeasible rules express that birds normally fly, chickens normally do not fly (except when they are scared) and something that flies normally nests in trees. In the following examples we abbreviate the above predicates by their first letters, e. g. in the following the predicate $c/1$ stands for $chicken/1$.

A *de.l.p.* $\mathcal{P} = (\mathit{\Pi}, \mathit{\Delta})$ describes the belief base of an agent and therefore contains not all of its beliefs. With the use of strict and defeasible rules it is possible to derive other literals, which may be in the agent's state of belief.

Definition 6 (Defeasible Derivation). *Let* $\mathcal{P} = (\mathit{\Pi}, \mathit{\Delta})$ *be a* de.l.p. *and let* $h \in \mathfrak{L}$*. A* (defeasible) derivation *of* h *from* $\mathcal{P}$*, denoted* $\mathcal{P} \mathrel{|\!\sim} h$*, consists of a finite sequence* $h_1, \ldots, h_n = h$ *of literals* ($h_i \in \mathfrak{L}$) *such that* h_i *is a fact* ($h_i \in \mathit{\Pi}$) *or there exists a strict or defeasible rule in* $\mathcal{P}$ *with head* h_i *and body* $b_1, \ldots, b_k$*, where every* b_l ($1 \leq l \leq k$) *is an element* h_j *with* $j < i$*. Let* $\mathcal{F}(\mathcal{P})$ *denote the set of all literals that have a defeasible derivation from* $\mathcal{P}$*.*

Example 2. In the *de.l.p.* from Example 1 there is $c(tina)$, $b(tina)$, $f(tina)$ a defeasible derivation of $f(tina)$, where the following rules have been used: $b(tina) \leftarrow c(tina)$ and $f(tina) \prec b(tina)$.

For a literal $h \in \mathfrak{L}$ there can be more than one defeasible derivation as will become clear by the examples in this paper, see also [6]. If the derivation of a literal h only uses strict rules, the derivation is called a *strict* derivation.

As facts and strict rules describe strict knowledge, it is reasonable to assume $\mathit{\Pi}$ to be non-contradictory, i. e. there are no derivations for complementary literals from $\mathit{\Pi}$ only. But if $\mathit{\Pi} \cup \mathit{\Delta}$ is contradictory (denoted $\mathit{\Pi} \cup \mathit{\Delta} \mathrel{|\!\sim} \bot$), then there exist defeasible derivations for complementary literals and in order to decide which of them (or any) should be believed by the agent, a formalism is needed. DeLP uses defeasible argumentation to determine which literal has the "strongest" grounds to believe in.

Definition 7 (Argument, Subargument). *Let $h \in \mathfrak{L}$ be a literal and let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p.. *$\langle \mathcal{A}, h \rangle$ is an* argument *for h, iff $\mathcal{A} \subseteq \Delta$, there exists a defeasible derivation of h from $\mathcal{P}' = (\Pi, \mathcal{A})$, the set $\Pi \cup \mathcal{A}$ is non-contradictory and $\mathcal{A}$ is minimal with respect to set inclusion. The literal h will be called* conclusion *and the set $\mathcal{A}$ will be called* support *of the argument $\langle \mathcal{A}, h \rangle$. An argument $\langle \mathcal{B}, q \rangle$ is a* subargument *of an argument $\langle \mathcal{A}, h \rangle$, iff $\mathcal{B} \subseteq \mathcal{A}$.*

Example 3. In the *de.l.p.* $\mathcal{P}$ from Example 1 $f(tina)$ has the two arguments $\langle \{f(tina) \prec b(tina)\}, f(tina) \rangle$ and $\langle \{f(tina) \prec c(tina), s(tina)\}, f(tina) \rangle$.

If h has a strict derivation from a *de.l.p.* $\mathcal{P}$, then h has the unique argument $\langle \emptyset, h \rangle$. because no defeasible rules are needed for the derivation of h and $\emptyset$ is minimal with respect to set inclusion to any other set. As Π is non-contradictory, there cannot exist a strict derivation for $\overline{h}$ and therefore there is no argument for $\overline{h}$, because for every potential argument $\langle \mathcal{A}, \overline{h} \rangle$ the set $\Pi \cup \mathcal{A}$ would be contradictory [6]. In general, it is possible to have arguments supporting complementary literals.

Definition 8 (Disagreement). *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p.. *Two literals h and h_1* disagree*, iff the set $\Pi \cup \{h, h_1\}$ is contradictory.*

Two complementary literal p und $\neg p$ disagree trivially, because for every *de.l.p.* $\mathcal{P} = (\Pi, \Delta)$ the set $\Pi \cup \{p, \neg p\}$ is contradictory. But two literals which are not contradictory, can disagree either. For $\Pi = \{(\neg h \leftarrow b), (h \leftarrow a)\}$ the literals a and b disagree, because $\Pi \cup \{a, b\}$ is contradictory.

Definition 9 (Counterargument). *An argument $\langle \mathcal{A}_1, h_1 \rangle$ is a* counterargument *to an argument $\langle \mathcal{A}_2, h_2 \rangle$ at a literal h, iff there exists a subargument $\langle \mathcal{A}, h \rangle$ of $\langle \mathcal{A}_2, h_2 \rangle$, such that h and h_1 disagree.*

If $\langle \mathcal{A}_1, h_1 \rangle$ is a counterargument to $\langle \mathcal{A}_2, h_2 \rangle$ at a literal h, then the subargument $\langle \mathcal{A}, h \rangle$ of $\langle \mathcal{A}_2, h_2 \rangle$ is called the *disagreement subargument*. If $h = h_2$, then $\langle \mathcal{A}_1, h_1 \rangle$ is called a *direct attack* on $\langle \mathcal{A}_2, h_2 \rangle$ and *indirect attack*, otherwise.

Example 4. In $\mathcal{P}$ from Example 1 there is $\langle \{\neg f(tina) \prec c(tina)\}, \neg f(tina) \rangle$ a direct attack to $\langle \{f(tina) \prec b(tina)\}, f(tina) \rangle$. Furthermore $\langle \{\neg f(tina) \prec c(tina)\}, \neg f(tina) \rangle$ is an indirect attack on $\langle \{(n(tina) \prec f(tina)), (f(tina) \prec b(tina))\}, n(tina) \rangle$ with the disagreement subargument $\langle \{(f(tina) \prec b(tina))\}, f(tina) \rangle$.

A central aspect of defeasible argumentation is a formal comparison criterion among arguments. For some examples of preference criterions see [6]. For the rest of this paper we use an abstract preference criterion $\succ$ defined as follows.

Definition 10 (Preference Criterion $\succ$). *A preference criterion among arguments is an irreflexive, antisymmetric relation and will be denoted by $\succ$. If $\langle \mathcal{A}_1, h_1 \rangle$ and $\langle \mathcal{A}_2, h_2 \rangle$ are arguments, $\langle \mathcal{A}_1, h_1 \rangle$ will be* strictly preferred *over $\langle \mathcal{A}_2, h_2 \rangle$, iff $\langle \mathcal{A}_1, h_1 \rangle \succ \langle \mathcal{A}_2, h_2 \rangle$.*

In general the totality of $\succ$ cannot be guaranteed. So there are three possible arrangements for an argument $\langle \mathcal{A}_1, h_1\rangle$ and a counterargument $\langle \mathcal{A}_2, h_2\rangle$.

- $\langle \mathcal{A}_2, h_2\rangle \succ \langle \mathcal{A}_1, h_1\rangle$: $\langle \mathcal{A}_2, h_2\rangle$ is called a *proper defeater* of $\langle \mathcal{A}_1, h_1\rangle$.
- $\langle \mathcal{A}_2, h_2\rangle \not\succ \langle \mathcal{A}_1, h_1\rangle$ and $\langle \mathcal{A}_2, h_2\rangle \not\prec \langle \mathcal{A}_1, h_1\rangle$: $\langle \mathcal{A}_2, h_2\rangle$ is called a *blocking defeater* of $\langle \mathcal{A}_1, h_1\rangle$ (and vice versa).
- $\langle \mathcal{A}_2, h_2\rangle \prec \langle \mathcal{A}_1, h_1\rangle$: $\langle \mathcal{A}_2, h_2\rangle$ is not an acceptable attack on $\langle \mathcal{A}_1, h_1\rangle$.

Example 5. A possible preference relation among arguments is *Generalized Specificty* [11]. According to this criterion an argument is preferred to another argument, iff the former one is more *specific* than the latter, i. e. (informally) iff the former one uses more facts or less rules. For example, $\langle\{c \prec a, b\}, c\rangle$ is more specific than $\langle\{\neg c \prec a\}, \neg c\rangle$ (suppose that a, b are facts of a given *de.l.p.*), because the former uses two facts a, b, while the latter only a. For a formal definition see [11, 6].

As $\succ$ is antisymmetric by definition, there is no equipreference among an argument and its counterargument. So we only have to consider the cases, that one argument is better than the other or that two arguments are incomparable.

Definition 11 (Defeater). *An argument $\langle \mathcal{A}_1, h_1\rangle$ is a* defeater *of an argument $\langle \mathcal{A}_2, h_2\rangle$, iff there is a subargument $\langle \mathcal{A}, h\rangle$ of $\langle \mathcal{A}_2, h_2\rangle$, such that $\langle \mathcal{A}_1, h_1\rangle$ is a counterargument of $\langle \mathcal{A}_2, h_2\rangle$ at literal h and either $\langle \mathcal{A}_1, h_1\rangle \succ \langle \mathcal{A}, h\rangle$ (*proper defeat*) or $\langle \mathcal{A}_1, h_1\rangle \not\succ \langle \mathcal{A}, h\rangle$ and $\langle \mathcal{A}, h\rangle \not\succ \langle \mathcal{A}_1, h_1\rangle$ (*blocking defeat*).*

When considering sequences of arguments, then the definition of defeat is not sufficient to describe a conclusive argumentation line. Defeat only takes an argument and its counterargument into consideration, but disregards preceeding arguments. But we expect also properties like *non-circularity* or *concordance* from an argumentation sequence. See [6] for a more detailed description of acceptable argumentation lines.

Definition 12 (Acceptable Argumentation Line). *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p. *and let $\Lambda = [\langle \mathcal{A}_1, h_1\rangle, \ldots, \mathcal{A}_n, h_n\rangle]$ be a sequence of arguments. Λ is called* acceptable argumentation line*, iff 1.) Λ is a finite sequence, 2.) every argument $\langle \mathcal{A}_i, h_i\rangle$ with $i > 1$ is a defeater of his predecessor $\langle \mathcal{A}_{i-1}, h_{i-1}\rangle$ and if $\langle \mathcal{A}_i, h_i\rangle$ is a blocking defeater of $\langle \mathcal{A}_{i-1}, h_{i-1}\rangle$ and $\langle \mathcal{A}_{i+1}, h_{i+1}\rangle$ exists, then $\langle \mathcal{A}_{i+1}, h_{i+1}\rangle$ is a proper defeater of $\langle \mathcal{A}_i, h_i\rangle$, 3.) $\Pi \cup \mathcal{A}_1 \cup \mathcal{A}_3 \cup \ldots$ is non-contradictory (*concordance of supporting arguments*), 4.) $\Pi \cup \mathcal{A}_2 \cup \mathcal{A}_4 \cup \ldots$ is non-contradictory (*concordance of interfering arguments*), and 5.) no argument $\langle \mathcal{A}_k, h_k\rangle$ is a subargument of an argument $\langle \mathcal{A}_i, h_i\rangle$ with $i < k$.*

Let $+$ denote the concatenation of argumentation lines and arguments, e. g. $[\langle \mathcal{A}_1, h_1\rangle, \ldots, \langle \mathcal{A}_n, h_n\rangle] + \langle \mathcal{B}, h\rangle$ stands for $[\langle \mathcal{A}_1, h_1\rangle, \ldots, \langle \mathcal{A}_n, h_n\rangle, \langle \mathcal{B}, h\rangle]$.

In DeLP a literal h is *warranted*, if there exists an argument $\langle \mathcal{A}, h\rangle$ which is non-defeated in the end. To decide whether $\langle \mathcal{A}, h\rangle$ is defeated or not, every acceptable argumentation line starting with $\langle \mathcal{A}, h\rangle$ has to be considered.

Definition 13 (Dialectical Tree). *Let $\langle \mathcal{A}_0, h_0 \rangle$ be an argument of a* de.l.p. $\mathcal{P} = (\Pi, \Delta)$*. A* dialectical tree *for $\langle \mathcal{A}_0, h_0 \rangle$, denoted $\mathcal{T}_{\langle \mathcal{A}_0, h_0 \rangle}$, is defined by*

1. *The root of $\mathcal{T}$ is $\langle \mathcal{A}_0, h_0 \rangle$.*
2. *Let $\langle \mathcal{A}_n, h_n \rangle$ be a node in $\mathcal{T}$ and let $\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle]$ be the sequence of nodes from the root to $\langle \mathcal{A}_n, h_n \rangle$. Let $\langle \mathcal{B}_1, q_1 \rangle, \ldots, \langle \mathcal{B}_k, q_k \rangle$ be the defeaters of $\langle \mathcal{A}_n, h_n \rangle$. For every defeater $\langle \mathcal{B}_i, q_i \rangle$ with $1 \leq i \leq k$, such that the argumentation line $\Lambda' = [\langle \mathcal{A}_0, h_0 \rangle, \ldots, \langle \mathcal{A}_n, h_n \rangle, \langle \mathcal{B}_i, q_i \rangle]$ is acceptable, the node $\langle \mathcal{A}_n, h_n \rangle$ has a child $\langle \mathcal{B}_i, q_i \rangle$. If there is no such $\langle \mathcal{B}_i, q_i \rangle$, the node $\langle \mathcal{A}_n, h_n \rangle$ is a leaf.*

In order to decide whether the argument at the root of a given dialectical tree is defeated or not, it is necessary to perform a *bottom-up*-analysis of the tree. There every leaf of the tree is marked "undefeated" and every inner node is marked "defeated", if it has at least one child node marked "undefeated". Otherwise it is marked "undefeated". Let $\mathcal{T}^*_{\langle \mathcal{A}, h \rangle}$ denote the marked dialectical tree of $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$.

Definition 14 (Warrant). *A literal $h \in \mathfrak{L}$ is* warranted*, iff there exists an argument $\langle \mathcal{A}, h \rangle$ for h, such that the root of the marked dialectical tree $\mathcal{T}^*_{\langle \mathcal{A}, h \rangle}$ is marked "undefeated". Then $\langle \mathcal{A}, h \rangle$ is a* warrant *for h.*

The notion of warrant is the topic of the next section.

4 Some interesting properties of warrant

The warrant procedure of DeLP is a way to compute the strongest beliefs of an agent. Thus the set of warranted literals (including all facts as they are trivially warranted using the empty argument) can be characterized as a belief set. One important property of belief sets is consistency. In this section we investigate the relationships between warranted literals and especially the consistency of the set of warranted literals.

If a literal h is warranted and an argument $\langle \mathcal{A}, h \rangle$ is a warrant for h, then $\langle \mathcal{A}, h \rangle$ is considered a "good" argument for h. But the quality of $\langle \mathcal{A}, h \rangle$ depends on its position in argumentation lines. If $\langle \mathcal{A}, h \rangle$ is at the beginning of an argumentation line, then it will be undefeated, as it is a warrant. It is also a "good" argument for h, if it is at second position in an argumentation line, as the following proposition shows.

Proposition 1. *If an argument $\langle \mathcal{A}, h \rangle$ is undefeated in the dialectical tree $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$, then it is undefeated in every dialectical tree $\mathcal{T}_{\langle \mathcal{A}', h' \rangle}$, where $\langle \mathcal{A}, h \rangle$ is a child of $\langle \mathcal{A}', h' \rangle$.*

But Proposition 1 can not be generalized to *"If an argument $\langle \mathcal{A}, h \rangle$ is undefeated in the dialectical tree $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$, then it is undefeated in every dialectical tree"*, as the following example shows:

Example 6. Consider the following *de.l.p.* $\mathcal{P} = (\Pi, \Delta)$ with $\Pi = \{a_1, a_2, a_3\}$ and $\Delta = \{(c \prec b), (\neg c \prec \neg d), (\neg d \prec a_1), (d \prec a_1, b), (b \prec a_1, a_3), (b \prec a_2), (\neg b \prec a_3)\}$. Let *Generalized Specificity* [11] be the preference relation among arguments. The dialectical tree $\mathcal{T}_{\langle \mathcal{A}, d \rangle}$ for the argument $\mathcal{T}_{\langle \mathcal{A}, d \rangle}$ with $\mathcal{A} = \{(d \prec a_1, b), (b \prec a_2)\}$ consists only of one argumentation line $[\langle \mathcal{A}, d \rangle, \langle \{(\neg b \prec a_3)\}, \neg b \rangle, \langle \{(b \prec a_1, a_3)\}, b \rangle]$. Observe that the argument $\langle \{(\neg d \prec a_1)\}, \neg d \rangle$ is not an attack on $\langle \mathcal{A}, d \rangle$ in $\mathcal{T}_{\langle \mathcal{A}, d \rangle}$, because $\langle \mathcal{A}, d \rangle$ is strictly more specific. Thus the argument $\langle \mathcal{A}, d \rangle$ is undefeated in $\mathcal{T}_{\langle \mathcal{A}, d \rangle}$. Let $\mathcal{T}_{\langle \mathcal{B}, c \rangle}$ be the dialectical tree for the argument $\langle \mathcal{B}, c \rangle$ with $\mathcal{B} = \{(c \prec b), (b \prec a_1, a_3)\}$. In $\mathcal{T}_{\langle \mathcal{B}, c \rangle}$ there is the (incomplete) argumentation line $\Lambda' = [\langle \mathcal{B}, c \rangle, \langle \{(\neg c \prec \neg d), (\neg d \prec a_1)\}, \neg c \rangle, \langle \mathcal{A}, d \rangle]$. As in $\mathcal{T}_{\langle \mathcal{A}, d \rangle}$ the argument $\langle \mathcal{A}, d \rangle$ has exactly one attack in $\mathcal{T}_{\langle \mathcal{B}, c \rangle}$ after Λ', namely $\langle \{(\neg b \prec a_3)\}, \neg b \rangle$. But different from the situation in $\mathcal{T}_{\langle \mathcal{A}, d \rangle}$ the argumentation line $\Lambda' + \langle \{(\neg b \prec a_3)\}, \neg b \rangle$ cannot be extended by the argument $\langle \{(b \prec a_1, a_3)\}, b \rangle$ as $\langle \{(b \prec a_1, a_3)\}, b \rangle$ is a subargument of $\langle \mathcal{B}, c \rangle$ and thus violates the properties of acceptable argumentation lines. Thus $\langle \mathcal{A}, d \rangle$ is defeated in $\mathcal{T}_{\langle \mathcal{B}, c \rangle}$.

Proposition 1 implies an interesting relationship between warranted literals: if an argument $\langle \mathcal{A}, h \rangle$ is a warrant, every argument $\langle \mathcal{A}', h' \rangle$ such that $\langle \mathcal{A}, h \rangle$ is an attack on $\langle \mathcal{A}', h' \rangle$, cannot be a warrant. Furthermore due to the definition of warrant, no two warranted literals can disagree.

Proposition 2. *Let $\mathcal{P}$ be a* de.l.p.. *If h and h' are warranted literals in $\mathcal{P}$, then h and h' cannot disagree.*

Although warranted literals cannot pairwise disagree, the set of all warranted literals might be inconsistent with the strict knowledge as the following example shows:

Example 7. Consider the *de.l.p.* $\mathcal{P} = (\Pi, \Delta)$ with $\Pi = \{a, (h \leftarrow c, d), (\neg h \leftarrow e, f)\}$ and $\Delta = \{(c \prec a), (d \prec a), (e \prec a), (f \prec a)\}$. In $\mathcal{P}$ the literals c, d, e, f are warranted, because for every $\phi \in \{c, d, e, f\}$ there is the argument $\langle \{\phi \prec a\}, \phi \rangle$, which has no counterarguments. But $\Pi \cup \{c, d, e, f\}$ is inconsistent, as there are derivations for h and $\neg h$. However all pairs and even all triples of $\{c, d, e, f\}$ are consistent with Π (e.g. $\Pi \cup \{c, d\} \not\mid\!\sim \bot$), as there cannot be derivations for h and $\neg h$ from them.

As we want to translate the notion of warrant into the terms of answer set semantics, this property of warranted literals will become a problem, as the literals in an answer set are (jointly) consistent. Because this form of disagreement is not captured in the terms of DeLP we formalize it here as *joint disagreement.*

Definition 15 (Joint disagreement). *Let $\mathcal{P} = (\Delta, \Pi)$ be a* de.l.p. *and let $h_1, \ldots, h_n$ be some literals. If $\{h_1, \ldots, h_n\} \cup \Pi \mid\!\sim \bot$, then $h_1, \ldots, h_n$ are said to be in* joint disagreement.

If a set W of literals is given, one might want to determine the literals of W that are not in joint disagreement. The most primitive construction of a set of literals, that do not jointly disagree, is set up by an argument.

Proposition 3. *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p.*, let $\langle \mathcal{A}, h \rangle$ be an argument such that $\{h, h_1, \ldots, h_n\} = \{head(\delta) \mid \delta \in \mathcal{A}\}$. Then $h, h_1, \ldots, h_n$ do not jointly disagree.*

Joint disagreement will play a crucial role when converting a *de.l.p.* into an answer set program in the next two sections.

When considering the set of all warranted literals, another relationship of interest between literals (more precisely between arguments warranting literals) is the subargument relation.

Proposition 4. *Let $\mathcal{P}$ be a* de.l.p. *and $\langle \mathcal{B}, h' \rangle$ an argument. If $\langle \mathcal{B}, h' \rangle$ is defeated in a dialectial process, i. e. $\langle \mathcal{B}, h' \rangle$ is marked "defeated" in $\mathcal{T}^*\langle \mathcal{B}, h' \rangle$, every argument $\langle \mathcal{A}, h \rangle$, such that $\langle \mathcal{B}, h' \rangle$ is a subargument of $\langle \mathcal{A}, h \rangle$, is also defeated in a dialectical process.*

Due to contraposition Proposition 4 implies directly the following corollary.

Corollary 1. *Let $\mathcal{P}$ be a* de.l.p.*. If h is a warranted literal in $\mathcal{P}$ and $\langle \mathcal{A}, h \rangle$ is a warrant for h, then h' is warranted in $\mathcal{P}$ for every subargument $\langle \mathcal{B}, h' \rangle$ of $\langle \mathcal{A}, h \rangle$.*

Current algorithms for computing warrant in DeLP only consider computing warrants for one literal [6, 3]. If all warranted literals are to be determined, the above results can prune the set of literals to be considered, when the warrant status for one literal has been shown.

5 Converting a defeasible logic program into an answer set program

In this section and the next, we present two different conversion techniques to transform a *de.l.p.* into an answer set program. The approach in this section aims at an intuitively correct way to transform defeasible and strict rules into answer set programming. Since the set of all warranted literals might be in joint disagreement, the activation of a transformed defeasible rule must be prohibited when leading to inconsistency. This leads to the notion of minimal disagreement sets.

Definition 16 (Minimal disagreement set). *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p.*. A* minimal disagreement set *$\mathcal{X} \subseteq \mathcal{F}(\mathcal{P})$ is a set of derivable literals such that $\mathcal{X} \cup \Pi \mathrel{|\!\sim} \perp$ and there is no proper subset $\mathcal{X}'$ of $\mathcal{X}$ with $\mathcal{X}' \cup \Pi \mathrel{|\!\sim} \perp$. Let furthermore $\mathfrak{X}(\mathcal{P})$ be the set of all minimal disagreement sets of $\mathcal{P}$.*

Example 8. Consider the *de.l.p.* $\mathcal{P} = (\Pi, \Delta)$ with $\Pi = \{a, b, (h \leftarrow c, d), (\neg h \leftarrow e)\}$ and $\Delta = \{(p \prec a), (\neg p \prec b), (c \prec b), (d \prec b), (e \prec a)\}$. The minimal disagreement sets are $\{h, \neg h\}, \{h, e\}, \{c, d, \neg h\}, \{c, d, e\}$ and $\{p, \neg p\}$.

Now joint disagreement can be subsumed by minimal disagreement sets: some literals $\{h_1, \ldots, h_n\}$ are in joint disagreement, iff there is a minimal disagreement set $\mathcal{X}$ with $\mathcal{X} \subseteq \{h_1, \ldots, h_n\}$.

Minimal disagreement sets will constrain the derivation of literals in the translated answer set program. If all but one literal of a minimal disagreement set are in the state under consideration, then the derivation of the last literal should be prohibited, in order to maintain consistency of the resulting answer set.

Definition 17 (Guard literals, guard rules). *Let $\mathcal{P}$ be a* de.l.p.. *The set of* guard literals $\mathit{GuardLit}(\mathcal{P})$ *for $\mathcal{P}$ is defined as* $\mathit{GuardLit}(\mathcal{P}) = \{\alpha_h | h \in \mathcal{F}(\mathcal{P})\}$ *with new symbols α_h. The set of* guard rules $\mathit{GuardRules}(\mathcal{P})$ *of $\mathcal{P}$ is defined as* $\mathit{GuardRules} = \{\alpha_h \leftarrow h_1, \ldots, h_n | \{h, h_1, \ldots, h_n\} \in \mathfrak{X}(\mathcal{P})\}$.

Example 9. We continue Example 8. Here we have $\{(\alpha_h \leftarrow \neg h), (\alpha_{\neg h} \leftarrow c, d), (\alpha_c \leftarrow d, \neg h), (\alpha_c \leftarrow d, e), (\alpha_d \leftarrow c, e)\} \subseteq \mathit{GuardRules}(\mathcal{P})$

We are now in the situation to propose our first translation of a *de.l.p.* into an answer set program.

Definition 18 (*de.lp*-induced answer set program). *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p.. *The $\mathcal{P}$*-induced answer set program $\mathrm{ASP}(\mathcal{P})$ *is defined as the minimal extended logic program satisfying 1.) for every $a \in \Pi$, $a \in \mathrm{ASP}(\mathcal{P})$, 2.) for every $r : h \leftarrow b_1, \ldots, b_n \in \Pi$, $r \in \mathrm{ASP}(\mathcal{P})$, 3.) for every $h \prec b_1, \ldots, b_n \in \Delta$, $h \leftarrow b_1, \ldots, b_n, \mathsf{not}\ \alpha_h \in \mathrm{ASP}(\mathcal{P})$ and 4.) $\mathit{GuardRules}(\mathcal{P}) \subseteq \mathrm{ASP}(\mathcal{P})$.*

This translation converts strict and defeasible rules in an intuitively correct manner in ASP-rules. Strict rules are applied whenever possible and defeasible rules are applied whenever consistency is preserved.

Example 10. From the *de.l.p.* of Example 8, the complete $\mathcal{P}$-induced answer set program $\mathrm{ASP}(\mathcal{P})$ arises as $\mathrm{ASP}(\mathcal{P}) = \{a, b, (h \leftarrow c, d), (\neg h \leftarrow e), (p \leftarrow a, \mathsf{not}\ \alpha_p), (\neg p \leftarrow b, \mathsf{not}\ \alpha_{\neg p}), (c \leftarrow b, \mathsf{not}\ \alpha_c), (d \leftarrow b, \mathsf{not}\ \alpha_d), (e \leftarrow a, \mathsf{not}\ \alpha_e)\}$ $\cup \mathit{GuardRules}(\mathcal{P})$ where some guard rules of $\mathcal{P}$ are as in Example 9.

We now investigate the relationship between arguments in a *de.l.p.* $\mathcal{P}$ and the answer sets of the $\mathcal{P}$-induced answer set program. Let $\mathcal{F}_\alpha(\mathcal{P}) = \mathcal{F}(\mathcal{P}) \cup \mathit{GuardLit}(\mathcal{P})$ denote the set of all derivable literals and their guard literals.

Proposition 5. *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p., *let $\langle \mathcal{A}, h \rangle$ be an argument such that $\{h, h_1, \ldots, h_n\} = \{\mathit{head}(\delta) \mid \delta \in \mathcal{A}\}$. Let $S \subseteq \mathcal{F}_\alpha(\mathcal{P})$ be a maximal subset such that 1.) $\{h, h_1, \ldots, h_n\} \subseteq S$, 2.) for all $l \in S \cap \mathcal{F}(\mathcal{P})$, there is an argument $\langle \mathcal{B}, l \rangle$ such that $\{\mathit{head}(\delta) \mid \delta \in \mathcal{B}\} \subseteq S$, 3.) S is consistent, i.e. no subset of S is an element of $\mathfrak{X}(\mathcal{P})$ and 4.) $\alpha_l \in S$ iff there is $X \in \mathfrak{X}(\mathcal{P})$ such that $X \backslash \{l\} \subseteq S$. Then S is an answer set of* $\mathrm{ASP}(\mathcal{P})$.

Theorem 1. *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p. *and* $\mathrm{ASP}(\mathcal{P})$ *the $\mathcal{P}$-induced answer set program. If h warranted in $\mathcal{P}$ then there exists at least one answer set M of* $\mathrm{ASP}(\mathcal{P})$ *with $h \in M$.*

But as the set of all warranted literals might be in joint disagreement, there can be in general no answer set S such that all warranted literals are in S.

Example 11. Consider the *de.l.p.* $\mathcal{P} = (\Pi, \Delta)$ with $\Pi = \{a, (h \leftarrow c, d), (\neg h \leftarrow e, f)\}$ and $\Delta = \{(c \prec a), (d \prec a), (e \prec a), (f \prec a)\}$. The literals c, d, e, f are warranted in $\mathcal{P}$, see Example 7. The $\mathcal{P}$-induced answer set program is given by

$$\mathrm{ASP}(\mathcal{P}) = \left\{ \begin{array}{lll} a. & & \\ h \leftarrow c, d & \neg h \leftarrow e, f & c \leftarrow a, \mathsf{not}\ \alpha_c \\ d \leftarrow a, \mathsf{not}\ \alpha_d & e \leftarrow a, \mathsf{not}\ \alpha_e & f \prec a, \mathsf{not}\ \alpha_f \\ \alpha_h \leftarrow \neg h & \alpha_h \leftarrow c, d & \alpha_{\neg h} \leftarrow h \\ \alpha_{\neg h} \leftarrow c, d & \alpha_c \leftarrow d, e, f & \alpha_c \leftarrow d, \neg h \\ \alpha_d \leftarrow c, e, f & \alpha_d \leftarrow c, \neg h & \alpha_e \leftarrow c, d, f \\ \alpha_e \leftarrow f, h & \alpha_f \leftarrow c, d, e & \alpha_f \leftarrow e, h \end{array} \right\} .$$

The answer sets of ASP($\mathcal{P}$) (without guard literals) are

$$\{c, d, e, h\}, \{c, d, f, h\}, \{f, e, f, \neg h\} \text{ and } \{c, e, f, \neg h\}.$$

Hence, there is no projected answer set S with $c, d, e, f \in S$.

As strict rules are the cause for minimal disagreement sets with cardinality greater than two, we can sharpen the above results for the special case that there are no strict rules.

Corollary 2. *Let* $\mathcal{P} = (\Pi, \Delta)$ *be a* de.l.p. *and* ASP($\mathcal{P}$) *the* $\mathcal{P}$*-induced answer set program. If* Π *does not contain any strict rule and* M *is the set of all warranted literals of* $\mathcal{P}$ *then there exists an answer set* M' *of* ASP($\mathcal{P}$) *with* $M \subseteq M'$*.*

The above results show the advantages and disadavantages of the $\mathcal{P}$-induced answer set program:

- Theorem 1 shows that every warranted literal is a credulous inference from $ASP(\mathcal{P})$ and more precisely: every answer set contains a subset of warranted literals. But as Example 11 shows, there does not necessarily exist an answer set, which contains all warranted literals.
- We can not determine the whole set of warranted literals by just computing the answer sets of $ASP(\mathcal{P})$ without doing any argumentation. A sceptical inference, i e. the intersection of all answer sets, is empty in most cases. On the other hand, if a literal can be inferred sceptically from $ASP(\mathcal{P})$, then it is warranted. But the conversion is not always true, as Example 11 shows, where the intersection of all answer sets is empty.

If we want to model warrant in general DeLP as a credulous inference from the induced answer set program, then it would be convenient, if we can determine one specific answer set to infer from (as in Corollary 2). This is the topic of the next section.

6 A simplified conversion

In this section we present an alternative conversion method to translate a *de.l.p.* into an answer set program. The method presented here is very trivial, but leads to quite stronger results than the above for a special case of preference relation among arguments and also solves the discrepancy described at the end of the last section for arbitrary preference relations.

In [4] the empty preference relation is used to translate a default logic program into a *de.l.p.*. Then the warrant of a literal is equivalent to the sceptical inference of that literal in the original default logic program. By translating a *de.l.p.* into an answer set program, we present here the other direction of this translation.

Definition 19 (*de.l.p**-induced answer set program). *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p.. *The $\mathcal{P}^*$*-induced answer set program $\mathrm{ASP}^*(\mathcal{P})$ *is defined as the minimal extended logic program satisfying 1.) for every $a \in \Pi$ it is $a \in \mathrm{ASP}^*(\mathcal{P})$ and 2.) for every (strict or defeasible) rule $h \multimap b_1, \ldots, b_n \in \Pi \cup \Delta$ it is $h \leftarrow b_1, \ldots, b_n, \mathsf{not}\ b'_1, \ldots, \mathsf{not}\ b'_m \in \mathrm{ASP}^*(\mathcal{P})$ where $\{b'_1, \ldots, b'_m\} = \{b | b$ and h disagree$\}$.*

Note that for this conversion into answer set semantics, only pairwise disagreement relations are taken into account. Moreover, strict and defeasible rules are treated likewise. This seems reasonable as Example 11 shows, that strict rules turn out to be the culprits for undercutting a general correspondence between warrant and sceptical inference.

Example 12. From the *de.l.p.* of Example 8, the complete $\mathcal{P}^*$-induced answer set program $\mathrm{ASP}^*(\mathcal{P})$ arises as $\mathrm{ASP}^*(\mathcal{P}) = \{a, b, (h \leftarrow c, d, \mathsf{not}\ \neg h, \mathsf{not}\ e), (\neg h \leftarrow e, \mathsf{not}\ h,), (p \leftarrow a, \mathsf{not}\ \neg p), (\neg p \leftarrow b, \mathsf{not}\ p), (c \leftarrow b), (d \leftarrow b), (e \leftarrow a, \mathsf{not}\ h)\}$. The resulting answer sets of $\mathrm{ASP}^*(\mathcal{P})$ are $\{a, b, c, d, e, \neg h, p\}$, $\{a, b, c, d, e, \neg h, \neg p\}$, $\{a, b, c, d, h, p\}$ and $\{a, b, c, d, h, \neg p\}$. If the preference relation is *Generalized Specificity* [11], then the set of warranted literals of $\mathcal{P}$ is $\{a, b, c, d\}$.

As one can see for the special case of a *de.l.p.* $\mathcal{P}$ with no strict rules, the $\mathcal{P}^*$- and the $\mathcal{P}$-induced translations collapse (in the sense of semantic equivalence). For general DeLP applying the *de.l.p.**-induced translation yields the following result for warranted literals:

Theorem 2. *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p.. *Let furthermore $\mathrm{ASP}^*(\mathcal{P})$ be the $\mathcal{P}^*$-induced answer set program. If M is the set of all warranted literals of $\mathcal{P}$, then there exists an answer set M' of $\mathrm{ASP}^*(\mathcal{P})$ with $M \subseteq M'$.*

This theorem states, that every warranted literal can be inferred credulously from its *-induced answer set program and even more, that the set of all warranted literals can be inferred credulously using one common answer set. But the inverted statement "If a literal can be inferred credulously, then it is warranted in the original *de.l.p.*" is not always true as Example 12 shows, where e can be inferred credulously, but is not warranted. More precisely, the answer set containing all warranted literals can also contain literals, that are not warranted.

We investigate now the implications of the above results for the special case $\mathsf{DeLP}^{\emptyset}$ of defeasible logic programs with empty preference relation. This yields a very specific characterization of warranted literals.

Proposition 6 (Remark 3.4 in [4]). *In $\mathsf{DeLP}^{\emptyset}$, a literal l is warranted iff there exists an argument for l that is not attacked.*

When the preference relation under consideration is empty, then warranted literals can be inferred sceptically from the resulting answer set program.

Theorem 3. *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p. *with the empty preference relation. Let* $\mathrm{ASP}^*(\mathcal{P})$ *the $\mathcal{P}^*$-induced answer set program and $M_1, \ldots, M_n$ be the answer sets of* $\mathrm{ASP}^*(\mathcal{P})$. *If M is the set of all warranted literals of $\mathcal{P}$ then $M \subseteq M_1 \cap \ldots \cap M_n$.*

Equality of M with the intersection of all answer sets does not always hold. Consider a *de.l.p.* $\mathcal{P} = (\Pi, \Delta)$ is given by $\Pi = \{q, r, h \leftarrow p, h \leftarrow \neg p\}$ and $\Delta = \{p \prec q, \neg p \prec r\}$. The $\mathcal{P}$-induced answer set program has two answer sets, each of which contains the literal h. But h is not warranted as every argument for h has a defeater attacking either the subargument for p or $\neg p$.

Theorem 3 can also easily give a result for arbitrary preference relations.

Corollary 3. *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p. *with an arbitrary preference relation. Let furthermore* $\mathrm{ASP}^*(\mathcal{P})$ *be the $\mathcal{P}^*$-induced answer set program and $M_1, \ldots, M_n$ be the answer sets of* $\mathrm{ASP}^*(\mathcal{P})$. *If $M' \subseteq \mathcal{F}(\mathcal{P})$ is the set of all literals that have an argument which is not attacked at all then $M' \subseteq M_1 \cap \ldots \cap M_n$.*

Sceptical ASP-inference does not cover all warranted literals for a *de.l.p.* with an arbitrary preference relation, but so does credulous inference as was shown with Theorem 2.

7 Conclusion and future work

Defeasible logic programming provides a framework for paraconsistent reasoning on the basis of dialectical argumentation. Answer set programming is one of the most popular approaches to default reasoning, which is similar to defeasible reasoning in that both methodologies aim at realizing nonmonotonic inferences. There is, however, a substantial difference between defeasible and default rules: While Reiter-style default rules have to be blocked specifically in order not to make their consequents believed, the validity of the consequents of defeasible rules must be evaluated in a complex process, taking the global interactions of all rules into account.

In this paper, we studied transformations of defeasible logic programs into answer set programs in order to make relationships between inference via a dialectical warrant procedure, on the one side, and answer set semantics, on the other side, explicit. We presented two types of conversions that differ with respect to the treatment of strict rules. While conversions of the first type maintain the distinction between strict and defeasible rules, conversions of the second type transform all rules into default rules. We proved that for conversions of both types, warrant implies credulous inference. For conversions of the second type, we obtained the stronger result that all warranted literals of the defeasible logic program are contained in one and the same answer set of the transformed logic program. Moreover, in some cases, we were able to show that warranted literals can be inferred skeptically in the answer set environment. In general, however, conversions of the first type establish a much weaker relationship between defeasible logic programming and answer set programming, as strict rules may lead to conflicting defeasible derivations. Of course, in the case that the defeasible logic program does not contain any strict rules, both conversions coincide.

As part of our ongoing work, we will combine our approach with ideas from [4] to obtain a complete picture of the links between defeasible argumentative reasoning in DeLP and answer set semantics. Furthermore it would be interesting to investigate these links when considering an altered version of DeLP using the techniques described in [2].

References

1. Ph. Besnard and A. Hunter. Towards a logic-based theory of argumentation. In *Proc. of the 17th American Nat. Conf. on Artif. Intelligence (AAAI'2000)*, pages 411–416, 2000.
2. Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5-6):286–310, 2007.
3. Carlos I. Chesñevar and Guillermo R. Simari. Towards computational models of natural argument using labelled deductive systems. In *Proc. of the 5th Intl. Workshop on Computational Models of Natural Argument (CMNA 2005)*, 2005.
4. Telma Delladio and Guilermo R. Simari. Relating DeLP and default logic. *Inteligencia Artificial, Revista Iberoamericana Inteligencia Artificial*, 35:101–109, 2007.
5. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *AIJ*, 77(2):321–358, 1995.
6. A. García and G. Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4(1-2):95–138, 2002.
7. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.
8. Henry Prakken and Gerard Vreeswijk. Logics for defeasible argumentation. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 4, pages 218–319. Kluwer Academic Publishers, Dordrecht, 2 edition, 2002.
9. Iyad Rahwan and Leila Amgoud. An argumentation-based approach for practical reasoning. In Gerhard Weiss and Peter Stone, editors, *5th International Joint Conference on Autonomous Agents and Multi Agent Systems, AAMAS'2006*, pages 347–354, 2006.
10. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.
11. F. Stolzenburg, A. García, C. Chesñevar, and G. Simari. Computing generalized specificity. *Journal of Non-Classical Logics*, 13(1):87–113, 2003.

A Proofs

Proposition 1. *If an argument $\langle \mathcal{A}, h \rangle$ is undefeated in the dialectical tree $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$, then it is undefeated in every dialectical tree $\mathcal{T}_{\langle \mathcal{A}', h' \rangle}$, where $\langle \mathcal{A}, h \rangle$ is a child of $\langle \mathcal{A}', h' \rangle$.*

Proof. Let $\langle \mathcal{A}, h \rangle$ be undefeated in the dialectical tree $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$. It is clear, that the subtree rooted at $\langle \mathcal{A}, h \rangle$ after $\langle \mathcal{A}', h' \rangle$ in $\mathcal{T}_{\langle \mathcal{A}', h' \rangle}$ is a subtree of $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$, because if any argumentation line $\Lambda = [\langle \mathcal{B}_1, h_1 \rangle, \ldots, \langle \mathcal{B}_n, h_n \rangle]$ is acceptable, then the argumentation line $\Lambda' = [\langle \mathcal{B}_2, h_2 \rangle, \ldots, \langle \mathcal{B}_n, h_n \rangle]$ is also acceptable, because the constraints on acceptance are harder in Λ than in Λ'. We show now that if $\Lambda = [\langle \mathcal{A}', h' \rangle, \langle \mathcal{A}, h \rangle, \langle \mathcal{C}_1, h_1 \rangle, \ldots, \langle \mathcal{C}_n, h_n \rangle]$ is an acceptable argumentation line in $\mathcal{T}_{\langle \mathcal{A}', h' \rangle}$ and $\langle \mathcal{C}_n, h_n \rangle$ is a supporting argument in $\mathcal{T}_{\langle \mathcal{A}', h' \rangle}$, then all attacks on $\langle \mathcal{C}_n, h_n \rangle$ in the argumentation line $\Lambda' = [\langle \mathcal{A}, h \rangle, \langle \mathcal{C}_1, h_1 \rangle, \ldots, \langle \mathcal{C}_n, h_n \rangle]$ in $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$ are valid attacks on $\langle \mathcal{C}_n, h_n \rangle$ in $\mathcal{T}_{\langle \mathcal{A}', h' \rangle}$. So let $\Lambda = [\langle \mathcal{A}', h' \rangle, \langle \mathcal{A}, h \rangle, \langle \mathcal{C}_1, h_1 \rangle, \ldots, \langle \mathcal{C}_n, h_n \rangle]$ be an acceptable argumentation line in $\mathcal{T}_{\langle \mathcal{A}', h' \rangle}$ and $\langle \mathcal{C}_n, h_n \rangle$ be a supporting argument in $\mathcal{T}_{\langle \mathcal{A}', h' \rangle}$. Then the argumentation line $\Lambda' = [\langle \mathcal{A}, h \rangle, \langle \mathcal{C}_1, h_1 \rangle, \ldots, \langle \mathcal{C}_n, h_n \rangle]$ is also acceptable in $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$. Let $\langle \mathcal{D}, g \rangle$ be an attack on $\langle \mathcal{C}_n, h_n \rangle$ in $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$ after Λ'. Then the argumentation line $\Lambda + \langle \mathcal{D}, g \rangle$ is acceptable:

1. *$\Lambda + \langle \mathcal{D}, g \rangle$ is a finite sequence as Λ is an acceptable argumentation line.*
2. *As $\langle \mathcal{D}, g \rangle$ is a valid defeater for $\langle \mathcal{C}_n, h_n \rangle$ in Λ', the same is true in $\Lambda + \langle \mathcal{D}, g \rangle$.*
3. *The set of supporting arguments of $\Lambda + \langle \mathcal{D}, g \rangle$ is the same as in Λ, because $\langle \mathcal{D}, g \rangle$ is an interfering argument in $\Lambda + \langle \mathcal{D}, g \rangle$.*
4. *The set of interfering arguments in $\Lambda + \langle \mathcal{D}, g \rangle$ is the same as the set of supporting arguments in Λ' and as Λ' is an acceptable argumentation line, the union of these arguments is non-contradictory.*
5. *The argument $\langle \mathcal{D}, g \rangle$ is not a subargument of $\langle \mathcal{A}, h \rangle, \langle \mathcal{C}_1, h_1 \rangle, \ldots, \langle \mathcal{C}_n, h_n \rangle$, because Λ' is an acceptable argumentation line. Furthermore $\langle \mathcal{D}, g \rangle$ is not a subargument of $\langle \mathcal{A}', h' \rangle$, because then the supporting arguments $\langle \mathcal{A}', h' \rangle$ and $\langle \mathcal{C}_n, h_n \rangle$ would contradict, as $\langle \mathcal{D}, g \rangle$ is a counterargument of $\langle \mathcal{C}_n, h_n \rangle$.*

As $\langle \mathcal{A}, h \rangle$ is undefeated in $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$ and no needed supporting argument of $\langle \mathcal{A}, h \rangle$ in $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$ gets lost in $\mathcal{T}_{\langle \mathcal{A}', h' \rangle}$, the argument $\langle \mathcal{A}, h \rangle$ is also undefeated in $\mathcal{T}_{\langle \mathcal{A}', h' \rangle}$

Proposition 2. *Let $\mathcal{P}$ be a* de.l.p.. *If h and h' are warranted literals in $\mathcal{P}$, then h and h' cannot disagree.*

Proof. As h is warranted, there exists an argument $\langle \mathcal{A}, h \rangle$ for h which is undefeated in a dialectial process. Suppose h' is a warranted literal and h and h' disagree. As h' is warranted there exists an argument $\langle \mathcal{A}', h' \rangle$ for h which is undefeated in a dialectial process. As h and h' disagree $\langle \mathcal{A}, h \rangle$ is a counterargument for $\langle \mathcal{A}', h' \rangle$ and vice versa. Either one of them is an attack on the other regarding a given preference relation. Without loss of generality let $\langle \mathcal{A}', h' \rangle$ be an attack on $\langle \mathcal{A}, h \rangle$. As $\langle \mathcal{A}', h' \rangle$ is a warrant for h' it is undefeated in a dialectical process, and according to Proposition 1 is marked undefeated in the dialectical tree of $\langle \mathcal{A}, h \rangle$ and so $\langle \mathcal{A}, h \rangle$ is defeated. So h is not warranted in $\mathcal{P}$ in contradiction to the assumption.

Proposition 3. *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p.*, let $\langle \mathcal{A}, h\rangle$ be an argument such that $\{h, h_1, \ldots, h_n\} = \{head(\delta) \mid \delta \in \mathcal{A}\}$. Then $h, h_1, \ldots, h_n$ do not jointly disagree.*

Proof. As $\langle \mathcal{A}, h\rangle$ is an argument, $\Pi \cup \mathcal{A}$ is non-contradictory and thus does not cause the derivation of complementary literals. As $\Pi \cup \mathcal{A} \mid\!\sim h, h_1, \ldots h_n$ the literals $h, h_1, \ldots, h_n$ do not jointly disagree.

Proposition 4. *Let $\mathcal{P}$ be a* de.l.p. *and $\langle \mathcal{B}, h'\rangle$ an argument. If $\langle \mathcal{B}, h'\rangle$ is defeated in a dialectial process, every argument $\langle \mathcal{A}, h\rangle$, such that $\langle \mathcal{B}, h'\rangle$ is a subargument of $\langle \mathcal{A}, h\rangle$, is also defeated in a dialectical process.*

Proof. Let $\langle \mathcal{B}, h'\rangle$ be defeated in a dialectical process and let $\langle \mathcal{A}, h\rangle$ be an argument such that $\langle \mathcal{B}, h'\rangle$ is a subargument of $\langle \mathcal{A}, h\rangle$. Let furthermore υ be the dialectical tree with root $\langle \mathcal{A}, h\rangle$ and υ' be the dialectical tree with root $\langle \mathcal{B}, h'\rangle$. Suppose $\langle \mathcal{A}, h\rangle$ is undefeated in υ, then all attacks on $\langle \mathcal{A}, h\rangle$ in υ are defeated in υ. As $\langle \mathcal{B}, h'\rangle$ is defeated in υ' there exists an argument $\langle \mathcal{C}, h''\rangle$ which is an undefeated attack on $\langle \mathcal{B}, h'\rangle$ in υ'. $\langle \mathcal{C}, h''\rangle$ is also an attack on $\langle \mathcal{A}, h\rangle$, because $\langle \mathcal{B}, h'\rangle$ is a subargument of $\langle \mathcal{A}, h\rangle$. It is clear due to the additional rules of $\mathcal{A}$ in comparison to $\mathcal{B}$ that the tree rooted at $\langle \mathcal{C}, h''\rangle$ under $\langle \mathcal{A}, h\rangle$ is a subtree of the tree rooted at $\langle \mathcal{C}, h''\rangle$ under $\langle \mathcal{B}, h'\rangle$. As $\langle \mathcal{C}, h''\rangle$ is undefeated in υ' but defeated in υ, there must exist at least one node $\langle \mathcal{D}, g\rangle$ in the tree rooted at $\langle \mathcal{C}, h''\rangle$ and interfering with $\langle \mathcal{B}, h'\rangle$ in υ' that is not in υ, provided its parentnode exists in υ. But if

$$\Lambda' = [\langle \mathcal{B}, h'\rangle, \langle \mathcal{C}, h''\rangle, \langle \mathcal{B}_1, k_1\rangle, \ldots, \langle \mathcal{B}_n, b_n\rangle, \langle \mathcal{D}, g\rangle]$$

is acceptable in υ' and $\Lambda = [\langle \mathcal{A}, h\rangle, \langle \mathcal{C}, h''\rangle, \langle \mathcal{B}_1, k_1\rangle, \ldots, \langle \mathcal{B}_n, b_n\rangle]$ is acceptable in υ, then $\Lambda + \langle \mathcal{D}, g\rangle$ is also accetable in υ:

- $\langle \mathcal{D}, g\rangle$ is concordant with all interfering arguments in Λ' and therefore $\langle \mathcal{D}, b\rangle$ is concordant with all interfering arguments in Λ as these two are the same.
- $\langle \mathcal{D}, g\rangle$ is not a subargument of any of the $\langle \mathcal{C}, h''\rangle, \langle \mathcal{B}_1, k_1\rangle, \ldots, \langle \mathcal{B}_n, b_n\rangle$ and also not of $\langle \mathcal{A}, h\rangle$ (because then $\langle \mathcal{B}_n, b_n\rangle$ would be non-concordant with the supporting arguments as $\langle \mathcal{D}, g\rangle$ attacks $\langle \mathcal{B}_n, b_n\rangle$).

So there is no node $\langle \mathcal{D}, g\rangle$ in the tree rooted at $\langle \mathcal{C}, h''\rangle$ and interfering with $\langle \mathcal{B}, h'\rangle$ in υ' that is not in υ, provided its parentnode exists in υ. Hence $\langle \mathcal{C}, h''\rangle$ is undefeated in υ in contradiction to the assumption and so $\langle \mathcal{A}, h\rangle$ is defeated in υ.

Corollary 1. *Let $\mathcal{P}$ be a* de.l.p.. *If h is a warranted literal in $\mathcal{P}$ and $\langle \mathcal{A}, h\rangle$ is a warrant for h, then h' is warranted in $\mathcal{P}$ for every subargument $\langle \mathcal{B}, h'\rangle$ of $\langle \mathcal{A}, h\rangle$.*

Proof. Suppose $\langle \mathcal{B}, h'\rangle$ is a subargument of $\langle \mathcal{A}, h\rangle$ and h' is not warranted. Then $\langle \mathcal{B}, h'\rangle$ is defeated in a dialectical process and according to Proposition 4 $\langle \mathcal{A}, h\rangle$ is also defeated in a dialectial process and cannot be a warrant for h in contradiction to the assumption.

Proposition 5. *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p.*, let $\langle \mathcal{A}, h\rangle$ be an argument such that $\{h, h_1, \ldots, h_n\} = \{head(\delta) \mid \delta \in \mathcal{A}\}$. Let $S \subseteq \mathcal{F}_\alpha(\mathcal{P})$ be a maximal subset* *such that 1.)* $\{h, h_1, \ldots, h_n\} \subseteq S$*, 2.) for all* $l \in S \cap \mathcal{F}(\mathcal{P})$*, there is an argument* $\langle \mathcal{B}, l\rangle$ *such that* $\{head(\delta) \mid \delta \in \mathcal{B}\} \subseteq S$*, 3.)* S *is consistent, i.e. no subset of* S *is an element of* $\mathfrak{X}(\mathcal{P})$ *and 4.)* $\alpha_l \in S$ *iff there is* $X \in \mathfrak{X}(\mathcal{P})$ *such that* $X\backslash\{l\} \subseteq S$*. Then* S *is an answer set of* $\mathrm{ASP}(\mathcal{P})$*.*

Proof. Let S_1 be the answer set of $\mathrm{ASP}(\mathcal{P})^S$. We have to show that $S = S_1$. We start with proving that $S \subseteq S_1$. First, let $l \in S \cap \mathcal{F}(\mathcal{P})$. By presupposition, there is an argument $\langle \mathcal{B}, l\rangle$ with $\{head(\delta) \mid \delta \in \mathcal{B}\} \subseteq S$. Let $k \prec b_1, \ldots, b_n \in \mathcal{B}$ a defeasible rule which translates into $k \leftarrow b_1, \ldots, b_n, not\ \alpha_k \in \mathrm{ASP}(\mathcal{P})$. Since $k \in S$ and S consistent, for no $X \in \mathfrak{X}(\mathcal{P})$ it holds that $X \setminus \{k\} \subseteq S$. Hence $\alpha_k \notin S$, which implies $k \leftarrow b_1, \ldots, b_n \in \mathrm{ASP}(\mathcal{P})^S$. Because l is derivable from $\Pi \cup \mathcal{B}$, it is also derivable via $\mathrm{ASP}(\mathcal{P})^S$, hence $l \in S_1$. Now, let $\alpha_l \in S$. Then there is $X \in \mathfrak{X}(\mathcal{P})$ such that $X\backslash\{l\} \subseteq S$. As a strict rule, $\alpha_l \leftarrow X\backslash\{l\} \in \mathrm{ASP}(\mathcal{P})^S$, and by what has just been shown, $X\backslash\{l\} \subseteq S \cap \mathcal{F}(\mathcal{P}) \subseteq S_1$. As S_1 is closed under application of rules from $\mathrm{ASP}(\mathcal{P})^S$, $\alpha_l \in S_1$. To show the converse subset relation to hold, assume $l \in S_1 \cap \mathcal{F}(\mathcal{P})$. Then there is a rule $r \in \mathrm{ASP}(\mathcal{P})$ such that $pos(r) \subseteq S_1, neg(r) \cap S_1 = \emptyset$ and $head(r) = l$. We prove $l \in S$ by (informal) induction on the length of the derivation of l via $\mathrm{ASP}(\mathcal{P})^S$. Assume first l to be a fact. Since all arguments of $\mathcal{P}$ must be consistent with all facts, and facts can be derived by an empty argument, and S is assumed to be maximal, $l \in S$. By induction hypothesis, assume $l \leftarrow b_1, \ldots, b_n$ to be a rule in $\mathrm{ASP}(\mathcal{P})^S$ with $\{b_1, \ldots, b_n\} \subseteq S$ which is induced either by a strict or by a defeasible rule of $\mathcal{P}$.

Case 1: $l \leftarrow b_1, \ldots, b_n$ is a strict rule of $\mathcal{P}$. By presupposition on S, each b_i is derivable via an argument $\mathcal{B}_i$ of $\mathcal{P}$ such that $\{head(\delta) \mid \delta \in \mathcal{B}_i\} \subseteq S$, hence $\{head(\delta) \mid \delta \in \bigcup \mathcal{B}_i\} \subseteq S$. As S is consistent, l can therefore be consistently derived by an argument $\langle \mathcal{B}, l\rangle$ of $\mathcal{P}$ such that $\mathcal{B} \subseteq S$. To make use of the maximality of S, we also have to show that $S \cup \{l\} \cup \Pi\mathsf{not} \vdash \bot$. However, whenever $S \cup \{l\} \cup \Pi \vdash \bot$, then already $S \cup \Pi \vdash \bot$, because l is the head of a strict rule all body literals of which are in S.

Case 2: $l \prec b_1, \ldots, b_n$ is a defeasible rule of $\mathcal{P}$. Then $l \leftarrow b_1, \ldots, b_n, not\ \alpha_l \in \mathrm{ASP}(\mathcal{P})$ and $l \leftarrow b_1, \ldots, b_n \in \mathrm{ASP}(\mathcal{P})^S$, hence $\alpha_l \notin S$. That means there is no $X \in \mathfrak{X}(\mathcal{P})$ such that $X\backslash\{l\} \subseteq S$. Since $b_1, \ldots, b_n \in S$, just as in Case 1, an argument for l can be constructed all rule heads of which – except possibly l – are in S. If l were not in S, by the maximality of S, this can only be if there were $S' \subseteq S$ such that $X = S' \cup \{l\} \in \mathfrak{X}(\mathcal{P})$; but then $X\backslash\{l\} \subseteq S$, a contradiction. Therefore, $l \in S$.

Finally, let $\alpha_l \in S_1$. Then α_l must be derivable via $\mathrm{ASP}(\mathcal{P})^S$, which means that there is $X \in \mathfrak{X}(\mathcal{P})$ such that $X\backslash\{l\} \subseteq S_1$ and $\alpha_l \leftarrow X\backslash\{l\} \in \mathrm{ASP}(\mathcal{P})^S$. As $X\backslash\{l\} \subseteq S_1 \cap \mathcal{F}(\mathcal{P})$, in particular, $X\backslash\{l\} \subseteq S$, therefore $\alpha_l \in S$ by definition of S. This finishes the proof of the proposition.

Theorem 1. *Let* $\mathcal{P} = (\Pi, \Delta)$ *be a* de.l.p. *and* $\mathrm{ASP}(\mathcal{P})$ *the* $\mathcal{P}$*-induced answer set program. If* h *warranted in* $\mathcal{P}$ *then there exists at least one answer set* M *of* $\mathrm{ASP}(\mathcal{P})$ *with* $h \in M$*.*

Proof. As h is warranted there exists at least one warrant $\langle \mathcal{A}, h\rangle$ for h with $\{h, h_1, \ldots, h_n\} = \{head(\delta) \mid \delta \in \mathcal{A}\} \cup \{h\}$. We have to show that a set S

with $\{h, h_1, \ldots, h_n\} \subseteq S$ as described in Proposition 5 exists. By Proposition 3, $h, h_1, \ldots, h_n$ do not jointly disagree, and set $S_0 = \{h, h_1, \ldots, h_n\}$. Assume $S_i, i \geq 0$, to be constructed, and choose $l \in \mathcal{F}(\mathcal{P}) \backslash S_i$ such that there is an argument $\langle \mathcal{B}, l\rangle$ with $\{head(\delta) \mid \delta \in \mathcal{B}\} \cup S_i$ consistent. Set $S_{i+1} = \{head(\delta) \mid \delta \in \mathcal{B}\} \cup S_i$. Continue this construction until no such further literal $l \in F\backslash S_i$ can be found; set $S' = S_i$. Finally, let $S = S' \cup \{\alpha_l \mid \exists\ X \in \mathfrak{X}(\mathcal{P})$ such that $X\backslash\{l\} \subseteq S'\}$. Then S satisfies all required properties, and it is clear that it is maximal among all such subsets of $\mathcal{F}(\mathcal{P})$.

Corollary 2. *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p. *and* ASP($\mathcal{P}$) *the $\mathcal{P}$-induced answer set program. If Π does not contain any strict rule and M is the set of all warranted literals of $\mathcal{P}$ then there exists an answer set M' of* ASP($\mathcal{P}$) *with $M \subseteq M'$.*

Proof. The answer set M' can be constructed using Proposition 5. As no two warranted literals disagree, there can be no minimal disagreement set completly in M'. Furthermore as every warranted literal has at least one argument supporting it, the set $S = M'$ in Proposition 5 can be derived straightforward.

Theorem 2. *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p.. *Let furthermore* $\mathrm{ASP}^*(\mathcal{P})$ *be the $\mathcal{P}^*$-induced answer set program. If M is the set of all warranted literals of $\mathcal{P}$, then there exists an answer set M' of* $\mathrm{ASP}^*(\mathcal{P})$ *with $M \subseteq M'$.*

Proof. Suppose a maximal subset $S \subseteq \mathcal{F}(\mathcal{P})$ with 1.) $M \subseteq S$ and 2.) no two literals $b, b' \in S$ disagree. According to Proposition 2 no two warranted literals can disagree with each other, so the constraints on S in 2. are well defined. We show now, that S is an answer set of $\mathrm{ASP}^*(\mathcal{P})$. *So let S_1 be the answer set of* $\mathrm{ASP}^*(\mathcal{P})^S$ *and prove $S = S_1$.*

$S \subseteq S_1$ Let $h \in S$. As $S \subseteq \mathcal{F}(\mathcal{P})$, the literal h has a defeasible derivation from rules $r_1, \ldots, r_m$ of $\mathcal{P}$. As no literal disagreeing with h is in S, the positive forms of the translated rules $r_1, \ldots, r_m$ in $\mathrm{ASP}^*(\mathcal{P})$ *are in* $\mathrm{ASP}^*(\mathcal{P})^S$ *and applicable. Thus it follows, that h can be derived in* $\mathrm{ASP}^*(\mathcal{P})^S$ *and so $h \in S_1$.*

$S \supseteq S_1$ Suppose $S_1 \setminus S = S' \neq \emptyset$ and $h \in S'$. Then h cannot disagree with any $h' \in S$, because then at least one rule used in the derivation of h in $\mathrm{ASP}^*(\mathcal{P})^S$ *would be ommitted when constructing the reduct* $\mathrm{ASP}^*(\mathcal{P})^S$. *This contradicts the maximality of S, as $S \cup \{h\}$ also fulfills the constraints given by 1. and 2.; so it follows $S' = \emptyset$.*

Proposition 6 (Remark 3.4 in [4]). *In* DeLP$^{\emptyset}$*, a literal l is warranted iff there exists an argument for l that is not attacked.*

Proof. The proof can be found in [4].

Theorem 3. *Let $\mathcal{P} = (\Pi, \Delta)$ be a* de.l.p. *with the empty preference relation. Let* $\mathrm{ASP}^*(\mathcal{P})$ *the $\mathcal{P}^*$-induced answer set program and $M_1, \ldots, M_n$ be the answer sets of* $\mathrm{ASP}^*(\mathcal{P})$. *If M is the set of all warranted literals of $\mathcal{P}$ then $M \subseteq M_1 \cap \ldots \cap M_n$.*

Proof. Let $h \in M$ be a warranted literal of $\mathcal{P}$. According Proposition 6, a literal h is warranted, if and only if there exists an argument $\langle \mathcal{A}, h\rangle$ of h that is not *attacked. Let* $\{b'_1, \ldots, b'_m\}$ *be the set of literals disagreeing with* h *or disagreeing with the conclusion of any subargument of* $\langle \mathcal{A}, h \rangle$. *If one* $b \in \{b'_1, \ldots, b'_m\}$ *could be derived in* $\mathrm{ASP}^*(\mathcal{P})$ *using rules* $r_1, \ldots, r_m$ *then* $\langle \{r_1, \ldots, r_m\}, b \rangle$ *would be an attack on* $\langle \mathcal{A}, h \rangle$ *in* $\mathcal{P}$. *As* $\langle \mathcal{A}, h \rangle$ *cannot be attacked, there is no derivation of a literal* $b \in \{b'_1, \ldots, b'_n\}$. *So* h *can be derived in every answer set of* $\mathrm{ASP}^*(\mathcal{P})$. *Thus it follows* $h \in M_1 \cap \ldots \cap M_n$.

Corollary 3. *Let* $\mathcal{P} = (\Pi, \Delta)$ *be a* de.l.p. *with an arbitrary preference relation. Let furthermore* $\mathrm{ASP}^*(\mathcal{P})$ *be the* $\mathcal{P}^*$*-induced answer set program and* $M_1, \ldots, M_n$ *be the answer sets of* $\mathrm{ASP}^*(\mathcal{P})$. *If* $M' \subseteq \mathcal{F}(\mathcal{P})$ *is the set of all literals, that have an argument, which is not attacked then* $M' \subseteq M_1 \cap \ldots \cap M_n$.

Proof. Follows directly from Theorem 3.