Updates of argumentation frameworks

Ján Šefránek
Comenius University, Bratislava, Slovakia, sefranek@ii.fmph.uniba.sk

Abstract

Two main topics are studied in this work. First, updates of assumption-based frameworks over deductive systems. Second, a problem of an inertia of an admissible set after an update of an abstract argumentation framework.

We consider an assumption-based framework over a logic program as composed of three parts – an argumentation framework, a deduction machinery and a knowledge base (a logic program). If the logic program is updated, the updates are transferred to the updates of the assumption-based framework.

In the second part an original abstract argumentation framework AF_o and an updating argumentation framework AF_u are assumed. We present a construction testing an inertia of a selected admissible set of AF_o, i.e., looking for an admissible set of the updated argumentation framework. which differs minimally from the selected admissible set.

Keywords: argumentation framework; assumption-based framework; logic program; update

Introduction

Our everyday argumentation is a dynamic process – new arguments and new attacks are discovered in the frame of disputes, controversies or internal dialogues. It can be said, that our (intuitive, implicit) argumentation frameworks are updated continuously. Attempts to understand and to model complex reasoning processes naturally lead to combining belief change and argumentation (Falappa et al. 2011).

However, updates of argumentation frameworks require a subtle formulation of research problems. As regards abstract argumentation frameworks, a straightforward addition of new arguments and new attacks may be sufficient. There is no need to solve conflicts after an update of an abstract argumentation framework, they are solved on the level of argumentation semantics. Of course, general properties of such updates represent interesting research problems (Cayrol, de Saint-Cyr, and Lagasquie-Schiex 2010).

On the other side, if we consider also a background knowledge base and/or a deductive system under the argumentation framework, at least two classes of problems are interesting for us.

In the first part of the paper we study assumption-based framework over an evolving knowledge base. The evolving knowledge base is represented by updates of logic programs. We try to use some techniques known from updates of logic programs and apply them to updates of assumption-based frameworks over a logic program (Bondarenko et al. 1997). Updates of logic programs from (Sefranek 2011) are transferred to updates of assumption-based frameworks over a logic program. Main contributions of the approach are: updates are distinguished from conflicts solving, irrelevant updates are disabled, the result of an update is expressible in the same language as the original and updating program and the third postulate of (Katsuno and Mendelzon 1991) is satisfied – if the original and updating program are coherent, then also the updated program is coherent. Most important are the last two features, they enable a straightforward transfer of logic program updates to updates of assumption-based frameworks. Methods, based on causal rejection principle (Leite 2003) do not have the features mentioned above.[1]

Second, it may be interesting to consider updates of an original (abstract) argumentation framework by an updating argumentation framework, to select an admissible set of the original framework and to look for an admissible set of the updated argumentation framework, which differs minimally from the input. This problem is studied in the second part of our paper. A motivation for studying this problem is connected to Dung's basic decision in his seminal paper (Dung 1995). He pointed out that a single argument cannot be considered as justified or correct. Only a justification with respect to a set of arguments makes sense. An *inertia of the given set of arguments* is expected or usual in many fields of human activities and in human reasoning. We can use a term strongly believed arguments.

The paper is structured as follows. After technical preliminaries we start with the part devoted to updates of assumption-based frameworks over a logic program. Conflicts solving in a sequence of logic programs and updating of original logic programs are described. Finally, consequences for updates of the whole assumption-based framework are presented. The second part contains a description of the problem, a procedure, solving the problem and a for-

[1] Rules are rejected only if there is a cause for it – in the case of conflict between rules a less preferred rule is rejected.

mulation of some more general properties. Finally, related work is overviewed and main contributions, open problems and future goals are summarized in Conclusions.

Preliminaries

Some basic notions of argumentation frameworks and logic programs are introduced in this section.

Argumentation frameworks An *argumentation framework* (Dung 1995) is a pair $AF = (AR, attacks)$, where AR is a set (of arguments) and $attacks \subseteq AR \times AR$ is a binary relation. Let be $a, b \in AR$; if $(a, b) \in attacks$, it is said that a attacks b.

Let be $S \subseteq AR$. It is said that S is *conflict-free* if for no $a, b \in S$ holds $(a, b) \in attacks$.

A set of arguments $S \subseteq AR$ attacks $a \in AR$ iff there is $b \in S$ s.t. $(b, a) \in attacks$.

A conflict-free set of arguments S is *admissible* in AF iff for each $a \in S$ holds: if there is $b \in AR$ s.t. $(b, a) \in attacks$, then S attacks b. An admissible set of arguments counterattacks each attack against its members.

Dung defined some semantic characterizations (extensions) of argumentation frameworks as sets of conflict-free and admissible arguments, which satisfy also some other conditions. We are interested here only in stable extensions.

A conflict-free $S \subseteq AR$ is a *stable extension* of AF iff S attacks each $a \in AR \setminus S$.

An *argumentation semantics* $\mathcal{S}$ of AF is a mapping, which assigns a set of sets of arguments (extensions) to AF.

An argumentation framework is *finitary*, if for each argument a is only finite number of arguments attacking a. We will assume only finitary argumentation frameworks in this paper.

Logic programs Only propositional extended logic programs are considered in this paper. Let $\mathcal{A}$ be a set of atoms. The set of objective literals is $\mathcal{O} = \mathcal{A} \cup \neg\mathcal{A}$, where $\neg\mathcal{A} = \{\neg A \mid A \in \mathcal{A}\}$. A convention: if $L = \neg A$, then $\neg L = A$.

The set of default literals is $\mathcal{D} = not\ \mathcal{O} = \{not\ L \mid L \in \mathcal{O}\}$. A literal is an objective literal or a default literal. The set of all literals is denoted by $\mathcal{L}$. A rule r is an expression of the form

$$L \leftarrow L_1, \ldots, L_k, not\ L_{k+1}, \ldots, not\ L_{k+m},$$

where $k \geq 0, m \geq 0$. If $k = 0$ and $m = 0$, it is said that the rule is a fact. L is called the head of the rule and denoted by $head(r)$. The set of literals $\{L_1, \ldots, L_k, not\ L_{k+1}, \ldots, not\ L_{k+m}\}$ is called the body of r and denoted by $body(r)$.

An extended logic program is a finite set of rules. We will often use only the term program. If a program P contains only literals from a set $\mathcal{L}$, it is said that P is a program over $\mathcal{L}$.

An interpretation is a consistent subset of $\mathcal{L}$, i.e., $I \subseteq \mathcal{L}$ s.t. for no atom A holds that $\{A, \neg A\} \subseteq I$ and for no objective literal L holds $\{L, not\ L\} \subseteq I$. An interpretation I is total iff for each objective literal L either $L \in I$ or $not\ L \in I$. A rule r is satisfied in an interpretation I iff $head(r) \in I$ whenever $body(r) \subseteq I$. An interpretation I is a model of P iff each rule $r \in P$ is satisfied in I.

Dynamic argumentation semantics

Natural argumentation is a dynamic process – new arguments and new attacks are usually discovered. Increasing research interest in the dynamics of argumentation is practically motivated and is challenging also from the theoretical point of view.

A kind of dynamics – updates – is introduced into a slightly simplified version of the assumption-based framework over a deductive system of (Bondarenko et al. 1997) in this paper. Some basic argumentation-theoretic semantics are defined in Bondarenko et al. for assumption-based frameworks and a set of nonmonotonic semantics is characterized in terms of argumentation semantics.

In this paper we are focused only on the specialization of the framework for logic programs. The specialization was constructed in Bondarenko et al., too. Only a simplified version of the construction is used in our paper. However, it is sufficient for our goals and its complete elaboration is straightforward. On the other side, we use a precise notion of an assumption-based framework over a logic program – it is specified in the next subsection.

Updates are introduced into assumption-based frameworks using dynamic answer set semantics of (Sefranek 2011). We stress that our approach to logic program updates is compatible with the framework of Bondarenko et al., it is also based on assumptions. *Dynamic answer set* semantics of Sefranek is intended as an extension of answer set semantics to sequences of logic programs, which is able to solve conflicts and to specify updates. Our approach to dynamic answer set semantics follows the construction of stable models in (Dimopoulos and Torres 1996).

We start with a presentation of the assumption-based framework for a logic program. Subsequently, the framework is applied to a sequence of two logic programs, where the second program specifies an update. Some conflicts between both programs might occur and we describe a preferential solving of conflicts. After that we propose a discrimination between preferential conflicts solving and updating. The dynamic answer set semantics is introduced. Notions and constructions proposed for logic program updates are applied to the assumption-based framework over logic programs. Finally, consequences for updates of arbitrary argumentation frameworks and argumentation semantics are outlined.

An assumption-based framework over a logic program can be viewed as a triple: an abstract argumentation framework (consisting of arguments and attacks) plus a deductive machinery (bottom up evaluation of the rules of a logic program) plus an evolving logic program (knowledge base). Updates of the logic program and their impact on a change of the argumentation framework are studied.

Assumption-based framework

An *assumption* is a default literal, assumptions are considered as (defeasible) *arguments* in this paper. Our attention is focused on sets of assumptions (sets of arguments).

An assumption-based framework is defined over a deductive system in Bondarenko et al.[2] If a set of arguments Δ is given,the behaviour of the deductive system is precisely specified by $\Delta^{\rightsquigarrow^P}$ of Dimopoulos and Torres:

$\Delta^{\rightsquigarrow^P}$ is a set of objective literals, dependent on Δ w.r.t. a program P in the following sense. Let P_Δ be the set of all rules from P, but with elements from Δ deleted from the bodies of the rules. Next, P^+_Δ is obtained from P_Δ by deleting all rules r with bodies containing assumptions.

Then $\Delta^{\rightsquigarrow^P} = \{L \in Obj \mid P^+_\Delta \models L\}$, where P^+_Δ is considered as a definite logic program with explicitly negated atoms as new symbols.

We will use sometimes rather loosely a phrase that Δ *generates* M w.r.t. a set of rules R, where M may be $\Delta^{\rightsquigarrow^R}$ or its subset, sometimes it may be also a union of Δ with sets mentioned here.

An argument $not\ A$ *attacks* an argument $not\ B$ w.r.t. a program P iff there is $B \in \{not\ A\}^{\rightsquigarrow_P}$. Notice that we have a notion of argument and a notion of attack. Hence, an *abstract argumentation framework* is specified for the assumption-based framework over a program P. The framework is denoted by $\mathcal{Q} = (\mathcal{D}, attacks)$, the symbol $\mathcal{Q}$ will be used later.

A set of arguments Δ *attacks* another set of arguments Δ' w.r.t. a program P if there is $A \in \Delta^{\rightsquigarrow_P}$ s.t. $not\ A \in \Delta'$ (Bondarenko et al.).

Δ is a set of *arguments for* an objective literal L w.r.t. P iff $L \in \Delta^{\rightsquigarrow^P}$; in that case, it is said also that there is a deduction of L from $P \cup \Delta$. The program P plays a role of a knowledge base in this deduction machinery.

An interpretation $S = \Delta \cup \Delta^{\rightsquigarrow^P}$ is an answer set of P iff S is a total interpretation (Dimopoulos and Torres 1996). The set of all answer sets of a program P is denoted by $AS(P)$. It is said, that a program P is *coherent*, if $AS(P) \neq \emptyset$. Otherwise it is incoherent.

An interesting case is, when the knowledge base is updated, i.e., the program P is updated by another program U. As a result, we may get an updated set of arguments, an updated attack relation and updated set of extensions. Note that a conflict may be derived after an update and $P \cup U$ may be incoherent, even if both P and U are coherent.

Preferential conflicts solving

Assume that an original program P was updated by a program U. Program $P \cup U$ may contain some conflicts.

The next example illustrates two types of conflicts.

[2]We do not define the deductive system, we do not need that concept here. An argumentation-theoretic framework over a deductive system is a triple in Bondarenko et al., but we need here only the set of all sets of arguments $\mathcal{H} = 2^{\mathcal{D}}$. Intuitively, a deductive system is understood in this paper as a bottom-up evaluation of logic programs, using T_P operator.

Example 1 *1. Let P be $a \leftarrow$ and U be $\neg a \leftarrow$, Δ be $\emptyset$. Then $\Delta^{\rightsquigarrow^P} = \{a\}$, $\Delta^{\rightsquigarrow^U} = \{\neg a\}$ and $\Delta^{\rightsquigarrow^{P\cup U}} = \{a, \neg a\}$. In order to define a dynamic semantics we have to resolve the conflict between $\Delta^{\rightsquigarrow^P}$ and $\Delta^{\rightsquigarrow^U}$, while preferring the updating program U.*

2. Consider $P = \{obedient \leftarrow punish\}$ and $U = \{punish \leftarrow not\ obedient\}$. Then for $\Delta = \{not\ obedient\}$ we have $\Delta^{\rightsquigarrow^{P\cup U}} = \{punish, obedient\}$; again, we have to resolve a conflict: $not\ obedient \in \Delta$, but $obedient \in \Delta^{\rightsquigarrow^{P\cup U}}$.

There is an analogy to types of conflicts recognized by the argumentation community – the first type of conflicts corresponds to *rebuttals*, and the second type to *assumption attack*. Some researchers use also the term *undercutting*, but (Pollock 1987) introduced originally this term with another meaning. Notice that approaches to logic program updates based on causal rejection principle solve only the first type.

It is said that $\Delta^{\rightsquigarrow^{P\cup U}}$ *contains a conflict* iff

- both $L, \neg L \in \Delta^{\rightsquigarrow^{P\cup U}}$, or
- $not\ L \in \Delta$ and $L \in \Delta^{\rightsquigarrow^{P\cup U}}$,

where $\langle P, U\rangle$ is a sequence of two programs, Δ is a set of arguments and $L \in Obj$.

Notice that $\Delta^{\rightsquigarrow^{P\cup U}}$ may contain a conflict even if Δ is conflict-free (w.r.t. $P \cup U$). Let P be $a \leftarrow not\ b$ and U be $\neg a \leftarrow not\ b$ and Δ be $\{not\ b\}$. Then Δ is conflict-free w.r.t. $P \cup U$, but $\Delta^{\rightsquigarrow^{P\cup U}}$ contains a conflict. It holds that $\Delta^{\rightsquigarrow^{P\cup U}}$ does not contain a conflict (is conflict-free)[3] iff $\Delta \cup \Delta^{\rightsquigarrow_P}$ is an interpretation.

We present a stepwise definition (see (Kruempelmann 2012)) of a solution of a conflict. First without consideration of a preference relation on programs. The preference relation is considered in the second step.

Definition 1 *A* solution *of a conflict C w.r.t. a set of arguments Δ is a minimal set of rules R s.t. $\Delta^{\rightsquigarrow^{(P\cup U)\setminus R}}$ does not contain C.*

Let Δ be a set of arguments, $\langle P, U\rangle$ a sequence of programs, $R \subseteq (P \cup U)$. We use a notation $\Delta^{\rightsquigarrow^R}_{cf}$ for a *conflict-free* set of conclusions of Δ.

Definition 2 *$\Delta^{\rightsquigarrow^R}_{cf}$ is a* maximal *conflict-free set of conclusions of Δ, if there is no R' s.t. $R \subset R' \subseteq P \cup U$ and $\Delta^{\rightsquigarrow^{R'}}$ is a conflict-free set of conclusions of Δ.* □

Notice that the notion of maximal conflict-free set of conclusions enables to meet the principle of minimal change.

Example 2 *Let $P = \{a \leftarrow; b \leftarrow\}$, $U = \{\neg a \leftarrow b\}$, $R_1 = \{a \leftarrow; \neg a \leftarrow b\}$, $R_2 = \{b \leftarrow; \neg a \leftarrow b\}$, $\Delta = \emptyset$.*

Then both $\Delta^{\rightsquigarrow^{R_1}}_{cf} = \{a\}$ and $\Delta^{\rightsquigarrow^{R_2}}_{cf} = \{b, \neg a\}$ are maximal conflict-free sets of conclusions of Δ.

[3]Remind that conflict-free set of conclusions of Δ and conflict-free Δ are different notions.

Let us analyze a slight modification of our example. Assume $P = \{b \leftarrow\}$, $U = \{a \leftarrow; \neg a \leftarrow b\}$. In this case maximal conflict-free sets of conclusions of Δ are the same as in the previous one, but it is not intuitive to take them as equivalent from a dynamic semantics point of view - information of U should be more preferred than the information of P, consequently, the rejection of $b \leftarrow$ is preferred to the rejection of $a \leftarrow$. $\Box$

Now, our first task is to introduce preferences on rules.

Consider two rules $r_1, r_2 \in P \cup U$. We say that r_2 is more preferred than r_1 iff $r_2 \in U$ and $r_1 \in P$ (notation: $r_1 \prec r_2$).

We can now define preferred sets of conclusions of a set of arguments.

Definition 3 *Let a sequence of programs $\langle P, U\rangle$ and a set of arguments Δ be given. Suppose that $R_1, R_2 \subseteq P \cup U$ and both $\Delta_{cf}^{\leadsto^{R_1}}$, $\Delta_{cf}^{\leadsto^{R_2}}$ are conflict-free sets of conclusions of Δ.*

If $\exists r_1 \in R_1 \setminus R_2\ \exists r_2 \in R_2 \setminus R_1\ r_2 \prec r_1$ and $\neg \exists r_3 \in R_2 \setminus R_1\ \exists r_4 \in R_1 \setminus R_2\ r_4 \prec r_3$ then $\Delta_{cf}^{\leadsto^{R_1}}$ is more preferred than $\Delta_{cf}^{\leadsto^{R_2}}$.

$\Delta_{cf}^{\leadsto^{R}}$ is a preferred *set of conclusions of Δ iff there is no more preferred set of conclusions of Δ than $\Delta_{cf}^{\leadsto^{R}}$. Notation: $\Delta_{cf+pref}^{\leadsto^{R}}$.*

$\Delta_{cf+pref}^{\leadsto^{R}}$, a preferred set of conclusions of Δ, is maximal*, if there is no R' s.t. $R \subset R' \subseteq P \cup U$ and $\Delta^{\leadsto^{R'}}$ is a preferred set of conclusions of Δ.* $\Box$

Example 3 *Remind Example 2. $\Delta_{cf+pref}^{\leadsto^{R_1}} = \{a\}$ is the maximal preferred set of conclusions of Δ.* $\Box$

Updating

An important point of our approach is that we distinguish between preferential conflict solving and updating.

Assume that we have an argument Δ and a maximal conflict-free preferred set of conclusions $\Delta_{cf+pref}^{\leadsto^{R}}$, $R \subseteq P \cup U$ and $\Delta \cup \Delta_{cf+pref}^{\leadsto^{R}}$ is a complete interpretation. It could be a basic candidate for a dynamic answer set. However, we have good reasons for some restrictions on P, U and Δ.

Some decisions are needed, in order to proceed from a task of (preferential) conflicts solving to a task of updating. We believe that it is not sufficient only to solve conflicts w.r.t. a preference relation in order to realize an update.

First, we assume that the original program P is consistent, i.e. there is a model of P. We follow a stance of (Katsuno and Mendelzon 1991): if a knowledge base is inconsistent, there is no way to eliminate it by *using update*. We accept it – inconsistent programs (knowledge bases) should be revised. Consequently, if P is inconsistent, then we will accept that there is no dynamic answer set of $\langle P, U\rangle$. On the other hand, conflicts between a knowledge base and a new information should be solved in the frame of an update.

The second decision: as regards the updating program U, a stronger condition is chosen.[4] If $AS(U) = \emptyset$, there are no dynamic answer sets of $\langle P, U\rangle$.

The third decision: *Inertia of the current state*. This is our most important decision. We believe that turning back at the semantic roots of updates is needed.[5] Consider an original program P and the set of all its answer sets $AS(P)$. $AS(P)$ can be viewed as a set of alternative descriptions of the current state of the world. Those descriptions are determined by some sets of (defeasible) arguments.

We claim that it is not reasonable to solve conflicts in $\Delta^{\leadsto^{P \cup U}}$ for arbitrary Δ. We express a reasonability criterion first in argumentation terms. Let a preference relation on arguments be induced by the proper subset relation, defeats are defined as (assumption) attacks or rebuttals of more preferred sets of arguments against less preferred sets of arguments and, finally, defeated sets of arguments do not generate reasonable results of updates.

Definition 4 (Defeat) *Let Δ, Ω be sets of arguments, R be a set of rules.*

It is said that Δ rebuts *Ω w.r.t. R, if there is $L \in \mathcal{O}$ s.t. $L \in \Delta^{\leadsto_R}$ and $\neg L \in \Omega^{\leadsto_R}$.*

If $\Delta \subset \Omega$, then Δ is more preferred than Ω (and Ω less preferred than Δ).

In that case, Δ defeats *Ω iff (Δ attacks or rebuts Ω w.r.t. R).*

If Ω attacks or rebuts Δ w.r.t. R, but it is less preferred than Δ, then Ω does not defeat Δ. Our point is that it is reasonable to accept minimal sets of assumptions (Occam's razor !). Consequently, it is not reasonable to accept a set of objective literals, dependent on defeated sets of arguments (and, e.g., to solve conflicts for Ω of the previous definition). More intuitions you can find in Example 4.

Example 4 *Let be*

$$P = \{d \leftarrow not\ n \qquad\qquad U = \{s \leftarrow s\}$$
$$n \leftarrow not\ d$$
$$s \leftarrow n, not\ c$$
$$\neg s \leftarrow\}$$

Consider the conflict $\{s, \neg s\} \subseteq \Delta^{\leadsto P \cup U}$ dependent on $\Delta = \{not\ d, not\ c\}$. Suppose that we solve the conflict by deleting the less preferred rule $\neg s \leftarrow$. However, $\neg s$ is in P (in $P \cup U$, too) dependent on $\emptyset$ and s is dependent on Δ. We do not accept solutions of conflicts based on non-minimal sets of assumptions in accordance with Occam's razor. $\Box$

Let us close our intuitions behind the idea of inertia of the current state. A description of a current state is dependent on a set of arguments Δ, which plays a role of a hypothesis.

[4]Our design decisions are not dogmas, different decisions are possible and reasonable.

[5]According to our view, a free selection of an interpretation checked by a fixpoint condition in approaches based on the causal rejection principle should be somehow restricted. We propose a restriction based on a notion of inertia of the current state.

Suppose that $\Delta \subset \Omega$ and $\Omega^{\rightsquigarrow^R}$ is a maximal conflict-free preferred set of conclusions of Ω, where $R \subseteq P \cup U$. Suppose also that there is $L \in \Delta^{\rightsquigarrow^P}$ and $\neg L \in \Omega^{\rightsquigarrow^R}$. We decided to consider Ω as a more extended set of assumptions than is necessary w.r.t. the current state of the world (in accordance with Occam's razor). Hence, we do not accept Ω or its supersets as a basis for a specification of a dynamic semantics.

Our third decision is expressed as a semi-formal Principle of the inertia of the current state and in terms of minimal active set of arguments (see Definition 6).

Principle of the inertia of the current state

Let Δ be a set of arguments and $\langle P, U \rangle$ a sequence of programs. Let $\Delta \cup \Delta^{\rightsquigarrow_{P \cup U}}$ be an answer set of P and also of $P \cup U$.

Then no set of arguments Ω, defeated by Δ, may generate an update of P by U. $\Box$

Definition 5 *A* cautious solution *of a conflict $C = \{A, \neg A\}$ dependent on a set of arguments Δ is a solution R which satisfies the conditions as follows:*

If $A \in \Delta^{\rightsquigarrow_R}_{cf+pref}$ (or $\neg A \in \Delta^{\rightsquigarrow_R}_{cf+pref}$, respectively) then there is no Ω, a proper subset of Δ and a set of rules R' s.t. $\neg A \in \Omega^{\rightsquigarrow^{R'}}_{cf+pref}$ (or $A \in \Omega^{\rightsquigarrow^{R'}}_{cf+pref}$, respectively). $\Box$

Cautious solutions are defined for conflicts dependent on arbitrary, unconstrained sets of assumptions. However, dynamic answer sets are expected to be total interpretations (with completed sets of assumptions) and we need to close our constructions for the case of *completed* sets of assumptions.

Example 5 [6] *Let P be $a \leftarrow$ and U be $\neg a \leftarrow not\ b$. Notice that $P \cup U$ is not coherent, hence inertia of the current state cannot be applied. Moreover, a preferential solution of the conflict $\{a, \neg a\}$ dependent on $\{not\ b\}$ is not cautious.*

We now motivate the last concept needed for defining dynamic answer sets. It is a concept of minimal active set of arguments (we use again a kind of Occam's razor).

Example 6 *Let P be $\{a \leftarrow; b \leftarrow a\}$ and U be $\{\neg a \leftarrow not\ b\}$.*

First, let be $\Delta_2 = \emptyset$. Then $\Delta_2^{\rightsquigarrow_{P \cup U}} = \{a, b\} = S^+$. We want to define dynamic answer set, and a natural requirement is that it is a total interpretation. Our goal is to find a set of assumptions S^-, a completion of Δ_2 s.t. $S^+ \cup S^-$ is a total interpretation. Thus, S^- is $\{not\ \neg a, not\ \neg b\}$.

Now, let be $\Delta_1 = \{not\ b\}$. In order to resolve the conflict $\{a, \neg a\} \subseteq \Delta_1^{\rightsquigarrow_{P \cup U}}$ in accordance with the preference relation, we have to consider the set of rules $R = (P \cup U) \setminus \{a \leftarrow\}$. Then $\Delta_1^{\rightsquigarrow_R} = \{\neg a\}$ and the corresponding total interpretation is $\{not\ b, not\ \neg b, not\ a\} \cup \{\neg a\}$.

The set of arguments $\Delta_1 = \{not\ b\}$ is a superset of the set of arguments $\Delta_2 = \emptyset$. Both can be considered as active sets of arguments *used in derivation of $\{a, b\}$ and $\{\neg a\}$, respectively. Supersets (completions) of Δ_1 and Δ_2 are needed only to obtain total interpretations. Notice that a subset relation, which holds for active sets of arguments, may not hold for corresponding completions.*

Only minimal active sets of arguments (Δ_2 in this example) are interesting from our point of view. In accordance with Occam's razor we do not assume more than is necessary for obtaining a reasonable semantic characterization of $\langle P, U \rangle$. $\Box$

Definition 6 *Let $\Delta \cup \Delta^{\rightsquigarrow_R}$ be a total interpretation. Let Ω be a minimal subset of Δ s.t. $\Delta^{\rightsquigarrow_R} = \Omega^{\rightsquigarrow_R}$. Then Ω is a* minimal active set of arguments *supporting $\Delta^{\rightsquigarrow_R}$.* $\Box$

Definition 7 (Dynamic Answer Set) *The set of all dynamic answer sets of $\langle P, U \rangle$ is denoted by $\Sigma_D(\langle P, U \rangle)$. If P has no model or U is incoherent then $\Sigma_D(\langle P, U \rangle) = \emptyset$.*

Otherwise, let a set of literals $S = \Delta \cup \Delta^{\rightsquigarrow_R}_{cf+pref}$ be a total interpretation. Then S is a dynamic answer set *of $\langle P, U \rangle$, if it satisfies the condition as follows:*

If Ω is a minimal active set of arguments supporting $\Delta^{\rightsquigarrow_R}_{cf+pref}$, then there is no $\Theta \cup \Theta^{\rightsquigarrow^{R'}}_{cf+pref}$, a total interpretation of $P \cup U$, $R' \subseteq P \cup U$, s.t. a minimal active set of arguments supporting $\Theta^{\rightsquigarrow^{R'}}_{cf+pref}$ is a proper subset of Ω. $\Box$

Consequences

In this subsection some observations, propositions and remarks aiming at a characterization of an assumption-based framework over an evolving logic program are provided. The characterization is focused on the questions as follows: if a set of arguments Δ generates a dynamic answer set what can or cannot be said about the Δ from the abstract argumentation point of view. What are the consequences of some interesting and new features of our approach to logic program updates for a transfer of updates to argumentation frameworks.

Proposition 1 *If $\Delta \cup \Delta^{\rightsquigarrow_R}_{cf+pref}$ is a dynamic answer set of $\langle P, U \rangle$ then Δ is a stable extension of the argumentation framework $\mathcal{Q}$.*

Proof: Δ is conflict-free. Suppose that $\Delta = \mathcal{D}$. Then there is no argument outside Δ and no attack against Δ. Notice that in this case there is the only dynamic answer set of $\langle P, U \rangle$. If an argument $not\ a$ is not in Δ, then $a \in \Delta^{\rightsquigarrow_R}_{cf+pref}$ (a dynamic answer set is a total interpretation). Therefore, Δ attacks $not\ a$. $\Box$

Remark 1 *If Δ is a stable extension of $\mathcal{Q}$, it may not generate a dynamic answer set of $\langle P, U \rangle$.*

Remind Example 4. Let Δ be $\{not\ c, not\ d, not\ \neg s, not\ \neg d, not\ \neg c, not\ \neg n\}$. Then $\{n, s\} = \Delta^{\rightsquigarrow_R}_{cf+pref}$, where $R = (P \cup U) \setminus \{\neg s \leftarrow\}$. Hence Δ attacks each argument outside it – i.e., Δ attacks $not\ n$ and $not\ s$.

But it does not generate a dynamic answer set of $\langle P, U \rangle$. $\{not\ c, not\ d\}$ is a minimal active set of arguments generating the set $\{n, s\}$, but it does not satisfy the conditions of

[6] Michal Strížzenec, personal communication

Definition 7 – $\neg s$ depends on $\emptyset$ w.r.t. $\langle P, U\rangle$. Observe that our notion of dynamic answer set prevents tautological and cyclic updates.

Moreover, Example 6 illustrates that our approach to logic program updates does not execute irrelevant updates, see (Sefranek 2011). A further research of irrelevant updates is planned and needed.

Next proposition expresses an important feature of the presented semantics of logic program updates. The result of an update is expressible in the same language as $\langle P, U\rangle$. This feature is not satisfied by approaches based on the causal rejection principle. Importance of this feature is based on the fact that the set of arguments over the logic program $P \cup U$ remains the same also after the update.

Proposition 2 (Representation, Sefranek) *Let $\langle P, U\rangle$ be a sequence of logic programs over the language $\mathcal{L}$. Then there is a logic program Q over the language $\mathcal{L}$ s.t. $AS(Q) = \Sigma_D(\langle P, U\rangle)$. It is said that Q* represents *the update of P by U.*

Consequence 3 *Let $P \cup U$ be a program over a set of literals L. Let $\mathcal{D} \subset \mathcal{L}$ be the set of arguments. Consider the update $\langle P, U\rangle$.*

If Q represents the update of P by U then each argument (assumption) of Q is a member of $\mathcal{D}$.

Our approach to logic program updates has also another feature, which is not satisfied by the approaches, based on the causal rejection principle. The feature corresponds to a postulate by (Katsuno and Mendelzon 1991) and its consequence for an assumption-based framework states that conflict-freeness of a set of assumptions remains satisfied after an update of a knowledge base under the assumption-based framework.

Proposition 4 (Sefranek) *Let $AS(P) \neq \emptyset \neq AS(U)$. Then $\Sigma_D(\langle P, U\rangle) \neq \emptyset$.*

Consequence 5 *Let there is a conflict-free set of assumptions w.r.t. a program P and a conflict-free set of assumptions w.r.t. U.*

Then there is a conflict-free set of assumptions w.r.t. Q, where $AS(Q) = \Sigma_D(\langle P, U\rangle)$.

As regards the impact of Consequence 5, remind Example 1, case 2. If a semantics of logic program updates solves only rebuttals (conflicts between heads of rules) – as each semantics based on the causal rejection principle – then in considered example is no dynamic semantics of $\langle P, U\rangle$, even if both P and U have answer sets. In terms of conflict-free sets of assumptions: both P and U has conflict-free sets of assumptions generating an answer set, but there is no conflict-fee set of assumptions generating a dynamic semantics of $\langle P, U\rangle$.

Remark 2 *Consider a set of arguments and an attack relation over an evolving knowledge base, i.e., over an update of logic programs. Then an arbitrary argumentation semantics can be specified for this (updated) assumption-based framework and also for its component, an abstract argumentation framework.*

We presented some consequences of a conception of logic program updates for updates of an assumption-based framework over a logic program. We now sketch a way how to make this conception of updates more general.

We describe a translation: Consider an assumption-based framework over a deductive system, a particular theory T, a set of assumptions A and a literal L. The relation $T \cup A \vdash L$ is defined in the framework. If we assign T to P (a logic program), A to Δ (a set of default negations) then we can replace each occurrence of $L \in \Delta^{\leadsto^P}$ by $T \cup A \vdash L$ in each definition, construction and proposition presented above, in such a way we get those constructions for an arbitrary assumption-based framework over a deductive system.

We suppose that the presented approach can be adapted also for structured argumentation frameworks.

Inertia of admissible sets

Some sets of arguments are for human or artificial reasoner often more preferred. We are interested in the second part of the paper in a problem of an inertia of sets of arguments after an update of an abstract argumentation framework.

Imagine some exceptional sets of strongly believed arguments, let us call them ideologies or belief patterns. We intend to formalize a behaviour of a rational reasoner, able to change his strong beliefs, if there are clear reasons. Hence, our problem in the following is that an update of an argumentation framework is given and a rational reasoner wants to check, whether a set of strongly believed arguments is justified also after the update.

We will formalize sets of strongly believed arguments by admissible sets and an update operation by a pair of argumentation frameworks – an original argumentation framework AF_o, and a new, updating argumentation framework AF_u.

Representation by logic programs

We will specify updates of argumentation frameworks in terms of a special kind of updates of very simple logic programs. Motivations for accepted design decisions are explained below in examples.

Definition 8 *Let an argumentation framework $AF = \langle AR, \alpha\rangle$ be given. We represent AF by a logic program P^{AF} as follows*

- *for each $(a, b) \in \alpha$ (i.e., a attacks b) there is a pair of rules in P^{AF}*

$$b \leftarrow not\ a$$
$$\neg b \leftarrow a$$

- *if $a \in AR$ is not attacked in AF then $a \leftarrow \in P^{AF}$.*

□

Our running example starts below.

Example 7 *Let $AF_o = (AR_o, \alpha_o)$ be given, where $AR_o = \{a, b, c, d\}$ and $\alpha_o = \{(a, b), (c, b), (b, d)\}$. Suppose that the set of strongly believed arguments is represented by an admissible set $A_4 = \{a, c\}$.*

P^{AF_o}, the logic program representing AF_o, is as follows:

$$
\begin{array}{ll}
r_1 : b \leftarrow not\ a & r'_1 : \neg b \leftarrow a \\
r_2 : b \leftarrow not\ c & r'_2 : \neg b \leftarrow c \\
r_3 : d \leftarrow not\ b & r'_3 : \neg d \leftarrow b \\
r_4 : a \leftarrow & \\
r_5 : c \leftarrow &
\end{array}
$$

Let us explain the construction of an arbitrary P^{AF} (from the viewpoint of admissibility). If $a \leftarrow$ is in P^{AF}, then a can be a member of an admissible set. If x is a member of an admissible set S and a rule $b \leftarrow not\ x$ is in P^{AF} then b cannot be a member of – the conflict-free – S. The use of "prime sibling" rules is related to updates and will be explained in Example 8. □

Definition 9 (Expansion) *Let $AF_o = (AR_o, \alpha_o)$ and $AF_u = (AR_u, \alpha_u)$ be given. Then $(AR_o \cup AR_u, \alpha_o \cup \alpha_u)$ is called the* expansion *of AF_o by AF_u. It is also said that AF_o is updated by AF_u.*

Example 8 (Continuation) *Suppose AF_o of Example 7., Let AF_u be (AR_u, α_u), where $AR_u = \{b, c, e, f\}$ and $\alpha_u = \{(e, c), (f, e), (e, f)\}$. Then the expansion $\mathcal{U}$ of AF_o by AF_u is $(\{a, b, c, d, e, f\}, \{(a, b), (c, b), (b, d), (e, c), (f, e), (e, f)\})$.*

We are going to construct the representation of $\mathcal{U}$, the program $P^{\mathcal{U}}$:

$$
\begin{array}{ll}
r_1 : b \leftarrow not\ a & r'_1 : \neg b \leftarrow a \\
r_2 : b \leftarrow not\ c & r'_2 : \neg b \leftarrow c \\
r_3 : d \leftarrow not\ b & r'_3 : \neg d \leftarrow b \\
r_4 : a \leftarrow & \\
r_5 : c \leftarrow & \\
r_6 : c \leftarrow not\ e & r'_6 : \neg c \leftarrow e \\
r_7 : e \leftarrow not\ f & r'_7 : \neg e \leftarrow f \\
r_8 : f \leftarrow not\ e & r'_8 : \neg f \leftarrow e
\end{array}
$$

We have to explain two design decisions. First, we use $P^{\mathcal{U}}$ instead of P^{AF_u}, i.e. the representation of the expansion instead of the representation of an updating argumentation framework. The reason is as follows – if an argument b is not attacked in AF_u then P^{AF_u} contains a fact $b \leftarrow$. This can cause problems for an update – b is attacked in AF_o, but the more preferred fact from the updating program should override rules expressing attacks against b in the corresponding logic program.

On the other hand, c is not attacked in AF_o, but it is attacked in AF_u, we need a "prime sibling" rule $\neg c \leftarrow e$ in order to specify the update correctly. We will see that sibling rules are not necessary, but we use them in accord with traditional patterns of logic program updates. □

Let us proceed now to the problem of "ideologies" updating. A game is played, where an admissible set S of AF_o is selected (strongly believed arguments) and our goal is to find an admissible set of the expansion, which differs from S minimally, in the sense that minimal number of "old" arguments is rejected. There are also other possible interpretations of a minimal difference between an original admissible set and an admissible set of the expansion. We will return to this question later.

Example 9 *We continue with the running example. Assume that we are interested in admissible sets of $\mathcal{U}$ with a minimal change w.r.t $A_4 = \{a, c\}$.*

There is a simple algorithmic solution of this task – a computation of all admissible sets of $\mathcal{U}$ and after that a selection of those with a minimal difference w.r.t. A_4.

However, an update of the logic program P^{AF_o} by the logic program $P^{\mathcal{U}}$ offers a computationally less demanding task. □

Our running example proceeds to an illustration of an update of logic programs. Note that the method presented below differs essentially from the method presented in the first part of this paper. We do not need such general method, simple logic programs representing abstract argumentation frameworks have some special features.

Example 10 *We will update the program $P^{\mathcal{U}}$. The program contains two strata. The first corresponds to the original argumentation framework.*

An update of $P^{\mathcal{U}}$ is processed as follows. First, fact r_5 will be removed because of a conflict with the rule r'_6 – c is not attacked in AF_o, but it is attacked in AF_u. Note that an "epistemic status" of facts in our representations of argumentation frameworks differs essentially from facts of usual logic programs.

Second, if all facts overridden by a rule with explicit negation in the head are removed then all rules with explicit negation in the head are removed, too (the only use of those rules is to update information about attacked rules, if needed; you can see that the same effect can be obtained also without sibling rules, but the explicit negations in heads enable a more natural procedure from the logic programs point of view).

The updated program is:

$$
\begin{array}{ll}
r_1 : & b \leftarrow not\ a \\
r_2 : & b \leftarrow not\ c \\
r_3 : & d \leftarrow not\ b \\
r_4 : & a \leftarrow \\
r_6 : & c \leftarrow not\ e \\
r_7 : & e \leftarrow not\ f \\
r_8 : & f \leftarrow not\ e
\end{array}
$$

The updated program may be used for the case of updates of admissible sets as a tester. The admissible set $A_4 = \{a, c\}$ is an input of the test, a is accepted, because of r_4, but *c is rejected:* $head(r_6)$ *matches* c*,* $body(r_6) = \{not\ e\}$*, but there is no rule with the head* e *and the body* $not\ a$ *or* $not\ c$*. Hence, the attack of* e *against* c *is not counterattacked and the ideology* $\{a, c\}$ *is reduced to the ideology* $\{a\}$*.*

Definition 10 *Let* $\mathcal{U}$ *be the expansion of the given argumentation frameworks* AF_o *and* AF_u*,* $P^{\mathcal{U}}$ *represents* $\mathcal{U}$*,* P^{AF_o} *represents* AF_o*. Let* A *be an arbitrary atom.*

Then an update of $P^{\mathcal{U}}$ *is the set of rules obtained by deleting*

1. *all facts of the form* $A \leftarrow$ *s.t. a rule* r *with* $head(r) = \neg A$ *is in* $P^{\mathcal{U}}$*,*
2. *subsequently, all rules* r *with* $head(r) = \neg A$*.*

In the next section the procedure illustrated in Example 10 is described. A simple encoding by an answer set program is not included because of the limited space.

Procedure

The procedure processes the update of $P^{\mathcal{U}}$, which was defined in Definition 10. It finds admissible sets of updated argumentation framework (i.e., of the expansion of AF_o by AF_u), which differs minimally from an input. The input is a selected admissible set of arguments of AF_o.

Arguments contained in the input are ground instances of $inIN(X)$. Arguments, contained in the output, are ground instances of $inOUT(X)$. Arguments and attacks are encoded as meta-facts denoting objects (facts, rules, heads and bodies of the rules) of $P^{\mathcal{U}}$.

A test of conflict-freeness of the input after the update is needed: attacks between arguments in input are solved (attacked arguments are out).

Resulting admissible sets are computed as follows. Preserved facts from the input belong to the output; out are arguments, which do not belong to the input and arguments from the input, which are not defended against attack by output; finally, arguments from the input, which are not out, belong to output.

A conservative stance is behind this procedure – some arguments may be rejected, but no new arguments can be accepted. An alternative, less conservative, procedure is as follows. If the difference between the cardinality of the input admissible set and an output admissible set is greater than 1, new arguments are inserted in a stepwise manner, conflict-freeness of the output and its ability to counterattack each attack is checked. Stop, if there is no additional argument suitable for the output or if the cardinality of new arguments is not less or equal as the cardinality of the difference.

Yet another strategy can be focused on looking for an admissible set that, in order to keep a maximum of its original arguments accepted, would add a minimum number of arguments, which defend some original arguments against attacks after the update.[7]

Properties

The following property is inspired by (Cayrol, de Saint-Cyr, and Lagasquie-Schiex 2010).

[7] This is a proposal of an anonymous reviewer.

Definition 11 *An update of an argumentation framework* AF_o *by an argumentation framework* AF_u *satisfies a property of* selective monotony*, if at least one admissible set of* AF_o *is also an admissible set of the expansion of* AF_o *by* AF_u*.*

Observation 6 *Let* S *be an admissible set of* AF_o *and* $AF_u = (A_u, \alpha_u)$ *be an update of* AF_o*.*

If for each $(b, a) \in \alpha_u$*, where* $a \in S$*, holds that there is* $c \in S$ *s.t.* $(c, b) \in \alpha_o \cup \alpha_u$ *then the update satisfies the* selective monotony *property.*

If an non-empty, but proper, part of an admissible set is preserved, we can speak about a weaker property, called partial selective monotony.

Related work

According to (Falappa et al. 2011) some argumentation formalisms can be used to define belief change operators, and belief change techniques have been used for modeling the dynamics of beliefs in argumentation formalisms. Our paper follows the second type of relations between argumentation and belief change. Complementary roles of belief change and argumentation in understanding and modeling complex reasoning processes are stressed in (Falappa et al. 2011).

According to our knowledge (and ignorance) most close to the problems of our paper are the problems studied in (Cayrol, de Saint-Cyr, and Lagasquie-Schiex 2010). Authors specified four basic change operations on argumentation frameworks – adding one interaction between arguments, removing one interaction, adding one argument together with its interactions, removing one argument and its interactions. An update of an original argumentation framework by another argumentation framework, which is studied in our paper, can be composed of a set of both basic addition operations. Main problems studied in (Cayrol, de Saint-Cyr, and Lagasquie-Schiex 2010) are as follows. How the set of extensions is changed, when arguments or attacks are changed. Conditions under which the change will not modify the previous extensions (here is a relation to our problem of an inertia of "ideologies"). Impact of a change on the structure of extensions and on the status of some particular arguments (the second of those problems is close to our goals in this paper). Hence, properties of the impact of a change studied in (Cayrol, de Saint-Cyr, and Lagasquie-Schiex 2010), are interesting from the point of view of our paper. Notions of monotony, introduced in (Cayrol, de Saint-Cyr, and Lagasquie-Schiex 2010) are too strong for our goals. But they inspired our property of (partial) selective monotony.

An interesting idea is to include argumentation dynamics into a context of complex reasoning tasks and systems. Reception of new information, evaluation of new information, change of beliefs and inference are the basic reasoning steps, while processing a new information according to (Falappa, Kern-Isberner, and Simari 2009).

A promising idea and method is presented in (Liao, Jan, and Koons 2011). An updated argumentation framework is divided into three parts. Arguments, affected by the update, unaffected and conditioning. Thus, a kind of modularity and abstraction is reached, when a status of particular arguments is computed. We will try to apply or modify the method in continuations of our research, presented in the second part of this paper.

Much work has been done in the field of logic program updates. According to (Slota and Leite 2010) handling the evolution of rule bases is still a largely open problem. We mention only some – say fundamental – problems of approaches based on causal rejection principle. The principle of minimal change is not satisfied. The third postulate of Katsuno and Mendelzon 1991 is not satisfied. The result of an update is not in general expressible in the same language as the original program and the updating program.

Conclusions

We presented a method enabling to transfer updates of logic programs to updates of assumption-based frameworks over a logic program. The method can be applied also to updates of general assumption-based frameworks over a deductive system and also to structured argumentation frameworks. In the second part of the paper a problem of a minimal change of a given admissible set after an update of an argumentation framework was studied.

Main contributions. According to our best knowledge both main problems of the paper are new. As regards logic program updates, we accepted improved constructions of (Sefranek 2011). New is an explicit formulation of the principle of inertia of the current state (based on a notion of defeat) and a new definition of dynamic answer set.

As regards the updates of assumption-based frameworks over logic programs the contributions may be summarized as follows. Relations between dynamic answer sets of $\langle P, U \rangle$ and stable extensions over the evolving logic program. Consequences of some important features of logic program updates according to (Sefranek 2011) are presented: set of arguments remains unchanged after an update of a logic program and we can get a conflict-free set of arguments after an update, if the original program and updating program were coherent. Both features are not guaranteed, if logic program updates comply with the causal rejection principle. Finally, it is outlined how this approach may be generalized for an arbitrary assumption-based framework over a deductive system.

Open problems, future research. A generalization of the presented approach to arbitrary assumption-based frameworks can be a rather straightforward research goal. A generalization to structured argumentation frameworks would deserve a rather extensive elaboration. A less conservative test of a minimal change (accepting also new arguments, two possible strategies are sketched in the paper) is a goal for a future research. Similarly, we intend to focus future research to a problem of an inertia of stable extensions after an update. Ideas, sketched in the second part of this paper, deserve an investigation in a future paper, an application of the method of (Liao, Jan, and Koons 2011) for that goal seems to be interesting.

Acknowledgements: This paper was supported by the grant 1/1333/12 of VEGA. Thanks to anonymous reviewers for valuable comments and suggestions.

References

Bondarenko, A.; Dung, P.; Kowalski, R.; and Toni, F. 1997. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence* 93:63–101.

Cayrol, C.; de Saint-Cyr, F.; and Lagasquie-Schiex, M.-C. 2010. Change in abstract argumentation frameworksl adding an argument. *Journal of Artificial Intelligence Research* 38:49–84.

Dimopoulos, Y., and Torres, A. 1996. Graph theoretical structures in logic programming and default theories. *Theoretical Computer Science* 170:209–244.

Dung, P. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77:321–357.

Falappa, M.; Garcia, A.; Kern-Isberner, G.; and Simari, G. 2011. On the evovling relation between belief revision and argumenation. *The Knowledge Engineering Review* 26:1:35–43.

Falappa, M.; Kern-Isberner, G.; and Simari, G. 2009. *Belief Revision and Argumentation Theory*. Springer. 341–360.

Katsuno, H., and Mendelzon, A. 1991. On the difference between updating a knowledge base and revising it. In *Proc. of KR'91*.

Kruempelmann, P. 2012. Dependency semantics for sequences of extended logic programs. *Logic Journal of the IGPL Oxford Journals*.

Leite, J. 2003. *Evolving Knowledge Bases: Specification and Semantics*. IOS Press.

Liao, B.; Jan, L.; and Koons, R. 2011. Dynamics of argumentation systems: a division-based method. *Artiificial Intelligence* 175:1790–1814.

Pollock, J. L. 1987. Defeasible reasoning. *Cognitive Science* 11(4):481–518.

Sefranek, J. 2011. Static and dynamic semantics. In *Proc. of MICAI. Special Session*.

Slota, M., and Leite, J. 2010. On semantic update operators for answer-set programs. In *Proc. of ECAI*.