Static and Dynamic Semantics. Preliminary Report

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

***Abstract*—This paper is aiming at a contribution to the logic program updates research. A (dynamic) semantics of sequences of logic programs is presented. The semantics is closely connected to the answer set semantics. We follow the construction of answer sets proposed by Dimopoulos and Torres. Solutions of three kinds of conflicts are added to the construction. After that, some decisions aiming at a step from inconsistencies handling to updating, are presented and analyzed. We present also some postulates characterizing relations between static and dynamic semantics. Subsequently a comparison to other postulates for logic program updates is provided. Presented postulates are satisfied by our dynamic semantics.**

Keywords: logic program updates, rationality postulates, answer set semantics, non-monotonic knowledge base

I. Introduction

Background. Representation of knowledge about dynamic world is a challenging topic with an impact on real applications. Recent research in logic program updates is aiming at understanding of the potential of logic programs to represent dynamic knowledge. It can be said that the dominating approach is based on a causal rejection principle, e.g, [12], [2], [4], [7]. However, also other important types of solutions should be recalled [14], [18], [5], [17], [19].

Some researchers devoted their attention to principles governing logic program updates, and/or to an analysis of well-known postulates for updates [10] from the logic program updates point of view (e.g. [7], [9], [12]).

Problem. According to [15] handling the evolution of rule bases is still a largely open problem. There is no generally accepted and exhaustive set of rationality postulates for logic program updates. The principle of minimal change is not satisfied in various approaches to logic program updates. According to our view among most serious faults of some approaches to logic program updates belong:

- the third postulate of Katsuno and Mendelzon[1] is not complied with;
- the result of an update is not in general expressible in the same language as the original program and the updating program;
- an appearance of irrelevant updates [16] (they should not change the original program, but they do it).

We believe that a new approach to logic program updates is needed. Open problems, described above, are addressed in this paper.

[1]If the original knowledge base and the updating formula are satisfiable then also the result of the update is satisfiable.

Goal and proposed solution We intend to present a new semantics of sequences of logic programs (a dynamic answer set semantics) and to introduce a set of rationality postulates for logic program updates. Usually rationality postulates for revisions or updates are expressed in terms of belief sets or belief bases, i.e. on a syntactic level. Our postulates are focused on a semantic level of updates. We believe that after turning the attention (of dynamic logic programming) from models to programs [1] and understanding the results of that, it would be important to look back at the semantic roots of updates. Our postulates specify relations between static and dynamic semantics. Answer set semantics plays the role of the static semantics in this paper. Dynamic semantics is intended in this paper as an extension of answer set semantics to sequences of logic programs, which is able to solve conflicts and to specify updates. Our notion of dynamic answer set explicitly follows the construction of stable models in [6].

Dynamic answer set semantics and postulates are main contributions of the paper. Significance of our approach is outlined by a discussion of postulates and by an analysis of (un)satisfaction of some postulates by some well known approaches to logic program updates in Section V.

II. Preliminaries

We assume a propositional language, i.e, a set $\mathcal{L}$ of propositional symbols (*atoms*). An *objective literal* is an atom A or *explicit negation* $\neg A$ of an atom. If L is an objective literal ($L \in Obj$), then $not\ L$ is called a default literal (notation: $not\ L \in Def$), not is called a default negation and the set of literals (Lit) is defined as $Obj \cup Def$. A convention: $\neg L = A$, if L is $\neg A$. A propositional rule is an expression of the form

$$L \leftarrow L_1, \ldots, L_k, \tag{1}$$

where $k \geq 0$ and $L \in Obj, L_i \in Lit$. The left (right) side of the rule r of the form (1) is called the head (body) of the rule, and denoted by $head(r)$ ($body(r)$). We will consider also integrity constraints (the rules with empty heads).

A propositional extended logic program (program hereafter) is a finite set of expressions of the form (1). If all atoms, used in a program P, are from a language $\mathcal{L}$, we will say that P is *over* $\mathcal{L}$.

The set of *conflicting literals* is defined as $CON = \{(L_1, L_2) \mid\ L_1 = not\ L_2 \vee L_1 = \neg L_2\}$. A set of literals S is *consistent* if it does not contain a pair of conflicting literals, i.e., $(S \times S) \cap CON = \emptyset$. An *interpretation* is a consistent set of literals. A *total* interpretation is an interpretation I such that for each $L \in Obj$ either $L \in I$ or $not\ L \in I$.

A literal is *satisfied* in an interpretation I iff $L \in I$. A set of literals S is satisfied in I iff $S \subseteq I$. A rule r is satisfied in I, if $head(r) \in I$ whenever $body(r) \subseteq I$. A program P is satisfied in I, if each $r \in P$ is satisfied in I. An interpretation, which satisfies a program P, is called a model of P.

Let S be a set of literals. Then $S^+ = S \cap Obj$ and $S^- = S \cap Def$. Let r^+ be $head(r) \leftarrow (body(r))^+$. A total interpretation S is an *answer set* of a program P, if S^+ is a minimal model of $\{r^+ \mid r \in P, (body(r))^- \subseteq S\}$. S is an interpretation, hence we consider only consistent answer sets. *A convention*: we will specify in examples usually only S^+ when speaking about an answer set S of a given logic program. 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.

III. Dynamic semantics

In this section a new semantics of sequences of logic programs is introduced. We follow a general non-monotonic semantic framework, which emphasize the role of

- (non-monotonic, defeasible) assumptions,
- dependencies on assumptions,
- and sets of accepted literals, dependent on assumptions.

An important representative of such framework is [6].

A formal construction of such a framework for a general non-monotonic knowledge bases is possible, however, we are going to build it for a special case, for the language of extended logic programs. We follow the approach of Dimopoulos and Torres [6], their definition of answer set (stable model) and we will try to adapt their approach to a construction of a (dynamic) semantics of sequences of logic programs, which conforms to the answer set semantics of logic programs.

Our construction consists of three steps. First we will adapt some basic definitions from [6] to the language of extended logic programs. The second step is focused on the case when the set of literals, dependent on assumptions is inconsistent. We define three kinds of conflicts, solution of conflicts and solutions based on a preference relation. Dynamic answer sets are defined at the third step after a presentation of our design decisions aiming at distinguishing updates from conflicts handling.

A. Dependencies on assumptions

An *assumption* is in [6] a default literal. A set of assumptions Δ is called a *hypothesis*. $\Delta^{\leadsto^P}$ is a set of objective literals, dependent on Δ w.r.t. a program (set of rules) P, here is a precise definition:

Let Δ, a hypothesis be given. P_Δ is the set of all rules from P, where elements from Δ are deleted from the bodies of the rules and P_Δ^+ is obtained from P_Δ by deleting all rules r with bodies containing assumptions.

Then $\Delta^{\leadsto^P} = \{L \in Obj \mid P_\Delta^+ \models L\}$, where P_Δ^+ is considered as a definite logic program with explicitly negated atoms as new symbols (atoms). We will consider only consistent $\Delta^{\leadsto^P}$.

Proposition 1 ([6]) *An interpretation $S = \Delta \cup \Delta^{\leadsto^P}$ is an answer set of P iff S is total interpretation.*

If S satisfies condition of Proposition 1 and the program P contains also integrity constraints, a further condition has to be satisfied: for each integrity constraint $\leftarrow L_1, \dots, L_k$ holds $\{L_1, \dots, L_k\} \not\subseteq S$.

B. Inconsistencies handling

In order to define a *dynamic* semantics following the approach of Dimopoulos and Torres, we have to add solutions of conflicts to their framework.

The next example illustrates three types of conflicts. It should be noticed that approaches based on causal rejection principle solve only the first type.

Example 2
1) Let P be $a \leftarrow$ and U be $\neg a \leftarrow$, Δ be $\emptyset$. Then $\Delta^{\leadsto^P} = \{a\}$ and $\Delta^{\leadsto^U} = \{\neg a\}$. In order to define a dynamic semantics we have to resolve the conflict between $\Delta^{\leadsto^P}$ and $\Delta^{\leadsto^U}$, while preferring the updating program U.
2) Consider $P = \{\text{obedient} \leftarrow \text{punish}\}$ and its more modern update $U = \{\text{punish} \leftarrow not\ \text{obedient}\}$. Then for $\Delta = \{not\ \text{obedient}\}$ we have $\Delta^{\leadsto^{P \cup U}} = \{\text{punish, obedient}\}$; again, we have to resolve a conflict: $not\ punish \in \Delta$, but $punish \in \Delta^{\leadsto^{P \cup U}}$.
3) Finally, suppose that $P = \{politician \leftarrow\}, U = \{honest \leftarrow; \leftarrow politician, honest\}$. $P \cup U$ does not satisfy the integrity constraint, we have to resolve that conflict.

□

Only sequences of the form $\langle P, U\rangle$ are considered in this paper. A straightforward extension to the general case is discussed in Section VI.

Definition 3 Consider $\langle P, U\rangle$, let Δ be a hypothesis and $L, L_i \in Obj$. It is said that $\Delta^{\leadsto^{P \cup U}}$ contains a conflict, if

- both $L, \neg L \in \Delta^{\leadsto^{P \cup U}}$, or
- $not\ L \in \Delta$ and $L \in \Delta^{\leadsto^{P \cup U}}$, or
- $\{L_1, \dots, L_k\} \subseteq \Delta^{\leadsto^{P \cup U}}$ and $\{not\ L_{k+1}, \dots not\ L_{k+m}\} \subseteq \Delta$, if $\leftarrow L_1, \dots, L_k, not\ L_{k+1}, \dots not\ L_{k+m}$ is an integrity constraint. □

We present a stepwise definition (see [11]) 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 4 A *solution* of a conflict C w.r.t. a hypothesis Δ is a minimal set of rules R s.t. $\Delta^{\leadsto^{(P \cup U) \setminus R}}$ does not contain C □

We abstract here from the question, whether an integrity constraint can be included in a solution.

Definition 5 Let Δ be a hypothesis, $\langle P, U\rangle$ a sequence of programs, $R \subseteq (P \cup U)$. Then $\Delta^{\leadsto^R}$ is a *conflict-free* set of conclusions of Δ if it contains no conflict.

$\Delta_{cf}^{\leadsto^R}$ 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 $\Delta^{\leadsto^{R'}}$ is a conflict-free set of conclusions of Δ. $\Box$

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

Example 6 Let be $P = \{a \leftarrow; b \leftarrow\}$, $U = \{\neg a \leftarrow b\}$, $R_1 = \{a \leftarrow; \neg a \leftarrow b\}$, $R_2 = \{b \leftarrow; \neg a \leftarrow b\}$, $\Delta = \emptyset$.

Then both $\Delta_{cf}^{\leadsto^{R_1}} = \{a\}$ and $\Delta_{cf}^{\leadsto^{R_2}} = \{b, \neg a\}$ are maximal conflict-free sets of conclusions of Δ. Both sets can be considered as intuitive candidates for dynamic answer sets of $\langle P, U\rangle$.

Intuitions are different, if a slight modification of our example is analyzed. Let be $P = \{b \leftarrow\}$, $U = \{a \leftarrow; \neg a \leftarrow b\}$, 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 is more preferred than the information of P, consequently, the rejection of $b \leftarrow$ is preferred to the rejection of $a \leftarrow$. Hence, $\{a\}$ seems to be an intuitive candidate for the (only) dynamic answer set of $\langle P, U\rangle$. $\Box$

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 hypothesis.

Definition 7 Let a sequence of programs $\langle P, U\rangle$ and $\Delta \subseteq Def$ 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 8 Remind Example 6. $\Delta_{cf+pref}^{\leadsto^{R_1}} = \{a\}$ is the maximal preferred set of conclusions of Δ. $\Box$

C. Updating

We intend to define a (dynamic) semantics of $\langle P, U\rangle$, which specifies a meaning of an update of P by U.

Assume that we have a hypothesis Δ and a maximal conflict-free preferred set of conclusions $\Delta^{\leadsto^R}$, $R \subseteq P \cup U$ and $\Delta \cup \Delta^{\leadsto^R}$ is a complete interpretation. It could be a basic candidate for 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 inconsistencies handling to a task of updating. We believe that it is not sufficient to consider only a preference relation for that goal.

We start with intuitions, after that follow some examples and, finally, definitions.

First, we assume that the original program P is consistent, i.e. there is a model of P. We follow a stance of [10]: if an inconsistency is introduced in a knowledge base, there is no way to eliminate it by *using update*. We accept it – inconsistencies resolving is a task for revisions Consequently, if P is inconsistent, then we will accept that there is no dynamic answer set of $\langle P, U\rangle$.

The second decision: as regards the updating program U, a stronger condition is chosen.[2] If $AS(U) = \emptyset$, there are no dynamic answer sets of $\langle P, U\rangle$.

If P is consistent, but incoherent, an update can (and should) be reduced to conflicts solving (if we want define an answer-set-like dynamic semantics). It is assumed below that P is coherent.

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.[3] 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 hypotheses. A goal is to specify how and when an update can change a hypothesis.

A crucial step is to decide what to do, if $\Delta^{\leadsto^{P \cup U}}$ contains a conflict for a hypothesis Δ. We claim that it is not reasonable to solve conflicts in $\Delta^{\leadsto^{P \cup U}}$ for arbitrary Δ.

An intuition behind the idea of inertia of the current state can be expressed informally as follows. Let $\langle P, U\rangle$ be given, $S = \Delta \cup \Delta^{\leadsto^P}$ be an answer set of P. We decided to accept S as a (dynamic) answer set of $\langle P, U\rangle$ unless there is a reason to reject it, i.e., if $\Delta \cup \Delta^{\leadsto^{P \cup U}}$ contains a conflict. On the other hand, the other – and more important – side of the inertia of the current state is focused on the hypothesis Δ. Suppose that $\Delta \subset \Omega$ and $\Omega^{\leadsto^R}$ is a maximal conflict-free preferred set of conclusions of Ω, where $R \subseteq P \cup U$. Suppose also that there is $L \in \Delta^{\leadsto^P}$ and $\neg L \in \Omega^{\leadsto^R}$. 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 $\Gamma \cup \Gamma^{\leadsto^Q}$ as a dynamic answer set of $\langle P, U\rangle$ ($\Omega \subseteq \Gamma$, $Q \subseteq P \cup U$ and $\Gamma^{\leadsto^Q}$ is a maximal conflict-free preferred set of conclusions of Γ). Our third decision is formalized in terms of cautious solution (see Definition 10) and minimal active hypothesis (see Definition 13).

[2] Our design decisions are not dogmas. They correspond to some intuitions and we are aiming to investigate a kind of reasoning based on them. However, different decisions are possible and reasonable. E.g., a possibility to solve incoherence of U using P is based on a "conservative" view, that a given state of knowledge can overcome or complete by inertia an incoherent update.

[3] 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.

Example 9 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 $\{s, \neg s\} \subseteq \Delta^{\rightsquigarrow P \cup U}$ dependent on $\Delta = \{not\ d, not\ c\}$. Suppose that we solve the conflict by deleting the rule $\neg s \leftarrow$. However, $\neg s$ is in P (in $P \cup U$, too) dependent on $\emptyset$ and s is dependent only on Δ. We do not accept solutions based on non-minimal sets of assumptions (a kind of Occam's razor). $\Box$

Definition 10 A *cautious solution* of a conflict $C = \{A, \neg A\}$ dependent on a hypothesis 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). $\Box$

Definition 11 (Dynamic View) Let $\langle P, U\rangle$ be given, P be consistent and U be coherent.

A *dynamic view* on $\langle P, U\rangle$ is a set of literals $\Delta \cup \Delta^{\rightsquigarrow R}_{cf+pref}$ s.t.

- $R \subseteq P \cup U$, R is a cautious solution of all conflicts in $P \cup U$ w.r.t. Δ,
- $\Delta^{\rightsquigarrow R}_{cf+pref}$ is a maximal preferred conflict-free set of conclusions of Δ, $\Box$

Cautious solutions are defined for conflicts dependent on arbitrary hypotheses. However, dynamic answer sets are expected to be total interpretations (with completed hypotheses) and we need to close our constructions for the case of completed hypotheses. To that end, we now motivate the last concept needed for defining dynamic answer sets. It is the concept of minimal active hypothesis (we use again a kind of Occam's razor).

Example 12 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 hypothesis 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 $\{not\ b, not\ \neg b, not\ a\} \cup \{\neg a\}$.

Hypothesis $\Delta_1 = \{not\ b\}$ is a superset of the hypothesis $\Delta_2 = \emptyset$. Both can be considered as *active hypotheses* 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 hypotheses, might not hold for corresponding completions.

Only minimal active hypotheses (Δ_2 in this example; notice that Δ_1 does not generate a cautious solution of the given conflict) 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 $\langle P, U\rangle$.

The notion of minimal active hypothesis will be used also in the definition of irrelevant updates (Postulate 2). We do not consider the update (the solution of conflicts), based here on Δ_1, as a relevant one. $\Box$

Definition 13 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 hypothesis* supporting $\Delta^{\rightsquigarrow R}$. $\Box$

Definition 14 (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 dynamic view on $\langle P, U\rangle$ and it is 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 hypothesis supporting $\Delta^{\rightsquigarrow R}_{cf+pref}$, then there is no $\Theta \cup \Theta^{\rightsquigarrow R}_{cf+pref}$, a dynamic view on $\langle P, U\rangle$, s.t. a minimal active hypothesis supporting $\Theta^{\rightsquigarrow R}_{cf+pref}$ is a proper subset of Ω. $\Box$

Evaluation of our definitions is based on postulates from the next section, see Theorem 26.

IV. Postulates

Let a *static semantics* σ be a mapping from programs over a language $\mathcal{L}$ to sets of interpretations and *dynamic semantics* Σ be a mapping from sequences of programs (over the language $\mathcal{L}$) to sets of interpretations.[4]

Intuitively, the sequence of programs represents a series of updates. It is assumed that an update operation is inherent in Σ and the meaning of the resulting, updated program is expressed by Σ. Some conditions concerning the relation of both semantics are expressed by our postulates.

Only sequences $\langle P, U\rangle$ of two programs are considered in the postulates of this paper. Rules of U are more preferred than rules of P. A generalization to arbitrary sequences is discussed in Section VI.

We assume in this paper that $\Sigma(\langle P, U\rangle) = \emptyset$ iff P has no model or $\sigma(U) = \emptyset$. But we do not accept it as a postulate - other decisions can lead to some interesting conclusions.

Postulate 1 (Representation of update) *Let $\langle P, U\rangle$ be a sequence of programs over a language $\mathcal{L}$ and $\Sigma(\langle P, U\rangle) \neq \emptyset$. Then there is a non-empty program Q over the language $\mathcal{L}$ such that $\sigma(Q) = \Sigma(\langle P, U\rangle)$.*

The main idea behind Postulate 1 is that the result of each update should be expressed in the same language as the sequence of programs. We consider this as an important feature, which is not satisfied e.g. by some semantics', based

[4]In our approach, $\sigma = AS$ is the answer set semantics and $\Sigma = \Sigma_D$. But the postulates are intended for arbitrary σ and Σ.

on the causal rejection principle. This postulate corresponds to an implicit assumption behind the first AGM postulate for revision [8] – a revised belief set $K^*\alpha$ is a belief set, i.e., a set of deductively closed sentences of the given language. A similar attitude is behind the KM (Katsuno-Mendelzon, [10]) postulates and behind theories of belief base revision.

Observation 15 *Postulate 1 is satisfied by $\sigma = AS$ and $\Sigma = \Sigma_D$.*

Proof[5] using a construction of [6]: Q is the set of all rules r s.t. there is a dynamic answer set $\Delta^+ \cup \Delta^-$, $body(r) = \Delta^-$ and $head(r) \in \Delta^+$. □

Irrelevant updates.

A first idea concerning irrelevant updates could be as follows. If $\sigma(P \cup U) = \sigma(P)$, the update proposed by U could be considered as an irrelevant one. The following example shows that the case of nonmonotonic knowledge bases (and nonmonotonic assumptions) deserve a more subtle solution.

The updating program can generate a new reasonable answer set, even if $\sigma(P \cup U) = \sigma(P)$:

Example 16 Consider $\langle P, U\rangle$:

$$P = \{a \leftarrow not\ b \qquad\qquad U = \{\neg a \leftarrow b\}$$
$$b \leftarrow not\ a$$
$$a \leftarrow b\}$$

It holds that $AS(P \cup U) = AS(P) = \{\{a\}\}$. Hence,$U$ does not contribute to the meaning of $P \cup U$. But we argue that U specifies a relevant update.

There is a conflicting dependence of $\{a\}$ on the assumption $\Delta_1 = \{not\ a\}$ in P and a conflict $\{a, \neg a\}$ dependent on Δ_1 in $P \cup U$. An intuitive solution is to reject the (less preferred) rule $a \leftarrow b$. After that we can get a new (dynamic) answer set $\{\neg a, b\}$ of $\langle P, U\rangle$. An acceptable intuitive meaning of $\Sigma(\langle P, U\rangle)$ is $\{\{a\}, \{\neg a, b\}\}$. Notice that $R = (P \cup U) \setminus \{a \leftarrow b\}$ is a cautious solution of the conflict.

It can be said that U solves a conflict contained in consequences of $\{not\ a\}$, see Definition 18 below.

Hence, according to our opinion, the condition that U does not solve a conflict in $P \cup U$ should be added to the condition $\sigma(P) = \sigma(P \cup U)$ in order to define irrelevant updates. □

Relevant and irrelevant updates may be seemingly distinguished in terms of strong equivalence as follows: if P and $P \cup U$ are strongly equivalent, then the update of P by U is irrelevant, otherwise it is relevant. However, the next example shows that it is a weak distinguishing criterion.

Example 17 Let be

$$P = \{a \leftarrow not\ c \qquad\qquad U = \{\neg a \leftarrow not\ b\}$$
$$b \leftarrow a$$

P and $P \cup U$ are not strongly equivalent, see $R = \{\neg a \leftarrow b; c \leftarrow \neg a\}$ and it is intuitive to consider the update of P by U as an irrelevant one: $\Delta_1 = \{not\ b, not\ c\}$ is a proper superset of $\Delta_2 = \{not\ c\}$, while Δ_2 generates an answer set of P and $P \cup U$. Observe that $\{a, \neg a\} \subseteq \Delta_1^{\rightsquigarrow^{P \cup U}}$ and there is no cautious solution of that conflict. □

Finally, remind the notion of minimal active hypothesis, which is crucial for defining dynamic answer sets. Hence, a hypothesis, which is not a minimal active one, cannot generate a dynamic answer set and, consequently, such hypothesis cannot be a basis of relevant updates.

Definition 18 Let Δ and $\langle P, U\rangle$ be given, let Ω be a minimal active hypothesis supporting $\Delta^{\rightsquigarrow_{P \cup U}}$. Suppose that $\Delta^{\rightsquigarrow_{P \cup U}}$ contains a conflict C.

It is said that U *solves* the conflict C in $\Delta^{\rightsquigarrow_{P \cup U}}$ if there is $R \subseteq P \cup U, U \subseteq R$ s.t. $\Delta_{cf}^{\rightsquigarrow R}$ is a maximal conflict-free set of conclusions of Δ and R is a cautious solution of C w.r.t. Ω. □

The condition that R is a cautious solution of C is essential in this definition.

Postulate 2 (Irrelevant updates) [6] *If $\sigma(P \cup U) = \sigma(P) \neq \emptyset$ and U does not solve a conflict in $\Delta^{\rightsquigarrow_{P \cup U}}$, then $\Sigma(\langle P, U\rangle) = \sigma(P)$.*

Observation 19 *Postulate 2 is satisfied for $\sigma = AS$ and $\Sigma = \Sigma_D$.*

Proof sketch: If $AS(P) = AS(P \cup U)$, U does not solve a conflict and $(S^- \cup S^+) \in \Sigma_D(\langle P, U\rangle)$, then S^+ is a conflict-free set of conclusions of S^- only if S is an answer set of P. □

Dominance of New Information

A principle of Dominance of New Information is accepted usually in theories of update and belief revision.

Our goal is to adapt that idea to a description how a static semantics of an updating program should be preserved in a dynamic semantics of the updated program. Consider the next example.

Example 20 Let be $P = \{a \leftarrow\}$ and $U = \{b \leftarrow\}$. It is intuitive to accept $\sigma(U) = \{\{b\}\}$, $\sigma(P) = \{\{a\}\}$ and $\Sigma(\langle P, U\rangle) = \{\{a, b\}\}$. Of course, $\sigma(U) \subseteq \Sigma(\langle P, U\rangle)$ is a too strong requirement. □

Definition 21 Let two languages $\mathcal{L}_1$ and $\mathcal{L}_2$ be given and S be a set of literals over $\mathcal{L}_1 \cup \mathcal{L}_2$. Then $S \uparrow_{\mathcal{L}_1}$ is $\{l \in S \mid l \text{ is a literal of } \mathcal{L}_1\}$.

Postulate 3 (Dominance of new information) *Suppose that U is not an irrelevant update in $\langle P, U\rangle$. If $S' \in \sigma(U)$, then there is an $S \in \Sigma(\langle P, U\rangle)$ s.t. $S \uparrow_{\mathcal{L}_U} = S'$.*

Observation 22 *Postulate 3 is satisfied for $\sigma = AS$ and $\Sigma = \Sigma_D$.*

[5]Only sketchy outlines of some ideas are used instead of proofs of observations because of a limited space.

[6]This postulate replaces for non-monotonic knowledge bases – in a sense – the second postulate by Katsuno and Mendelzon, which can be considered as a postulate specifying irrelevant updates for monotonic knowledge bases.

An idea of a proof; Let Δ be a hypothesis and $\Delta^{\leadsto^U} \cup \Delta \in AS(U)$. Consider $\Delta^{\leadsto^{P\cup U}}$. If it contains a conflict, a solution contains all rules of U. $\Box$

Postulate 4 *If $\sigma(P \cup U) = \sigma(U)$, then $\Sigma(\langle P, U\rangle) = \sigma(U)$.*

Postulate 4 is symmetric, in a sense, to Postulate 2.

Observation 23 *Postulate 4 is satisfied by $\sigma = AS$ and $\Sigma = \Sigma_D$.*

Proof Sketch: If $S = \Delta^{\leadsto^U} \cup \Delta \in AS(P \cup U) = AS(U)$, then S is a dynamic view on $\langle P, U\rangle$. Let Ω be a minimal active hypothesis supporting $\Delta^{\leadsto^U}$. It follows from the minimality of $S \in AS(U)$, that no proper subset of Ω can be a minimal active hypothesis supporting a set of consequences $\Theta^{\leadsto^U}$ in another $\Theta^{\leadsto^U} \cup \Theta \in AS(P \cup U) = AS(U)$. Hence, $S \in \Sigma_D(\langle P, U\rangle)$. $\Box$

Postulate 5 (Satisfiability preservation) *Let $\sigma(P) \neq \emptyset$ and $\sigma(U) \neq \emptyset$. Then $\Sigma(\langle P, U\rangle) \neq \emptyset$.*

Postulate 5 corresponds to the third KM postulate [10] (if both ψ and μ are satisfiable then also $\psi \diamond \mu$ is satisfiable). We require a well defined meaning of the updated program, if both the original and the updating program are coherent (if there is a conflict between both programs then the dynamic semantics specifies a solution of the conflict).

Observation 24 *Postulate 5 is satisfied by $\sigma = AS$ and $\Sigma = \Sigma_D$.*

Proof idea: If $\sigma(P \cup U) = \emptyset$, then U solves conflicts. $\Box$

Postulate 6 (Update of programs over disjoint languages) *Let P and U be programs over disjoint languages. Then $\Sigma(\langle P, U\rangle) = \Sigma(\langle U, P\rangle) = \sigma(P \cup U)$.*

It is natural to require that meanings of programs over disjoint languages do not interfere. Postulate 6 is inspired by [7], [9].

Observation 25 *Postulate 6 is satisfied for $\sigma = AS$ and $\Sigma = \Sigma_D$.*

Proof sketch: Suppose that $AS(P \cup U) \neq \emptyset$ and $S = \Delta \cup \Delta^{\leadsto^{P\cup U}} \in AS(P \cup U)$. There are no conflicts in $P \cup U$, hence $S \in \Sigma_D(\langle P, U\rangle)$. Obviously, if $S \in \Sigma_D(\langle P, U\rangle)$, then $S \in AS(P \cup U)$. $\Box$

Theorem 26 *Dynamic semantics Σ_D from Definition 14 satisfies Postulates 1 - 6.*

Proof - A consequence of Observations. $\Box$

V. Discussion

First we discuss the relations of our postulates and postulates of [7], [9], expressed in terms of σ and Σ. After that is briefly discussed satisfaction of our postulates by some well known approaches to logic program updates.

Addition of tautologies: if U contains only tautologies then $\Sigma\langle P, U\rangle = \sigma(P)$. It is a consequence of Postulate 2. U does not contribute to the meaning of $P \cup U$: it does not solve conflicts in $\Delta^{\leadsto_{P\cup U}}$, where Δ is a minimal active hypothesis supporting $\Delta^{\leadsto^{P\cup U}}$.

Initialization $\Sigma(\langle \emptyset, U\rangle) = \sigma(U)$. A consequence of Postulate 4.

Idempotence $\Sigma(\langle P, P\rangle) = \sigma(P)$. A consequence of Postulate 2 (and also of 4).

Augmented Update: if $P \subseteq U$ then $\Sigma(\langle P, U\rangle) = \sigma(U)$. A consequence of Postulate 4.

Let us point out our postulates which essentially differ from postulates of [7], [9] and similar principles and are not satisfied by some prominent approaches to logic program updates. New and important, we believe, are Postulates 1, 2 and 5. According to our view, representation of a result of an update in the language of the given logic programs, prevention of irrelevant updates and satisfiability preservation are important features of logic program updates.

Now, we turn to the satisfaction of our postulates by some approaches to logic program updates. Only some basic observations are sketched because of limited space. We will return to the problem in a future paper.

a) Approaches based on the causal rejection principle.: Postulates 1 and 2 are not satisfied[7]. Remind Example 12 and consider its reformulation in the language of generalized logic programs: $P = \{a \leftarrow; b \leftarrow a\}, U = \{not\ a \leftarrow not\ b\}$. There are two dynamic stable models of that update, $\{a, b\}$ and $\emptyset$. There is no logic program with such stable models, hence the result of this update cannot be represented in the language of the original and updating program, see [12], Theorem 33. Moreover, this example shows that $\emptyset$ is a semantic result of an irrelevant update of P by U.

Example 2, case 2 shows that Postulate 5 can not be satisfied by an approach based on causal rejection principle - only conflicts between heads of rules are solved by that class of approaches.

b) Approach based on structural properties: We consider [9] – postulates of that paper and the presented semantics are proposed for the pairs of programs, too. But our arguments apply also to other papers.

Postulate 1 is not satisfied. An update program is over a language, containing new abducibles, not appearing in original and updating program. Similarly as for updates based on the causal rejection principle, Postulates 2 and 5 are not satisfied; the same examples can be used as above (and their translations into abductive programs).

[7]There are different semantics, based on the causal rejection principle. Our argumentation in this paragraph refers to [12], but it applies also to [4], therefore also to other semantics, less cautious than the well supported semantics

We do not accept *Strong consistency* of [9] (if $\sigma(P \cup U) \neq \emptyset$ then $\Sigma(\langle P, U\rangle) = \sigma(P \cup U)$). Remind Example 16. See also the next example.

Example 27 Let P be $\{\neg a \leftarrow \mathit{not}\ b\}$ and U be $\{a \leftarrow \mathit{not}\ b; b \leftarrow \mathit{not}\ a\}$.

$P \cup U$ has only one answer set $S = \{b\}$. However, U solves a conflict in $\{\mathit{not}\ b\}^{\rightsquigarrow_{P\cup U}}$. If we respect the preference of U over P then the dependence of $\neg a$ on $\mathit{not}\ b$ should be overridden by the dependence of a on $\mathit{not}\ b$ in the more preferred program. Therefore, $S' = \{a\}$ should represent an intended meaning of $\langle P, U\rangle$, too. Hence, $\Sigma(\langle P, U\rangle) = \{\{b\}, \{a\}\}$ is an adequate choice.

Conclusion: $\sigma(P \cup U) \neq \emptyset$, but $\sigma(P \cup U) \neq \Sigma(\langle P, U\rangle)$. □

Other works, e.g. [14], [19], [18], [5], will be discussed in a future, extended version of this paper.

VI. Conclusions

Main Contributions. We presented a set of postulates governing logic program updates. As important features of logic program updates are emphasized:

- representation of a result of an update in the language of the given logic programs,
- irrelevant updates should be prevented,
- and satisfiability of an original and an updating program should be preserved in the updated program.

Some well known postulates proposed by [7], [9] follow from our postulates.

A dynamic answer set semantics, which obeys Postulates 1 – 6, was introduced. The semantics follows the approach of [6] to stable model semantics by adding conflict solving and updating features to that approach.

Open Problems and Future Work. Because of a limited space we did not present a detailed study of main approaches to logic program updates from the viewpoint of our postulates. It will be discussed in a future paper. Similarly, we will study computational properties of dynamic answer set semantics in a future paper.

Our set of postulates deserves a future work. An extension of the set is possible and a comparison to all relevant postulates in the literature is necessary.

We believe that the postulates are relevant for more general case of non-monotonic knowledge bases (characterized by a semantic framework based on defeasible assumptions and dependencies). A formal elaboration of such a generalization of the presented approach is among our future goals.

Another kind of generalization is a straightforward one. $\Sigma(\langle P_1, \ldots, P_k\rangle$ can be defined in an analogical way as $\Sigma(\langle P, U\rangle)$, A representation of $\langle P_1, \ldots, P_n\rangle$ by a program Q can be required by a modified Postulate 1. If we add a postulate requiring associativity of Σ, then a proposition holds, saying that the meaning of a sequence of programs can be constructed iteratively, i.e. $\Sigma(\langle P_1, \ldots, P_k\rangle) = \Sigma(\langle Q, P_k\rangle)$, where $\sigma(Q) = \Sigma(\langle P_1, \ldots, P_{k-1}\rangle)$ and, consequently, each finite sequence of programs $\langle P_1, \ldots, P_k\rangle$, $k \geq 3$, can be replaced by a sequence of two programs $\langle P, U\rangle$. However, for that case Q should be expressed in a language with preferences on rules. We will devote to this problem a future paper. The problem of rejecting integrity constraints (together with an introduction of weak constraints) is also among the goals of a future research.

A generalization of the presented semantics to a multidimensional case is among our other goals. We plan also to include strong equivalence (see [15]) – both to postulates and to a dynamic semantics framework.

Updates of logic programs with consistency restoring rules [3] seems to be an interesting problem.

Acknowledgements. I am grateful to David Pearce for valuable comments and proposals. This paper was supported by the grant 1/0688/10 of VEGA.

References

[1] Leite,. J., Pereira, I.M.: Generalizing Updates: From Models to Programs. In LPKR(1997) 224-246

[2] Alferes, J.J., Leite, J.A., Pereira, L.M., Przymusinska, H., Przymusinski, T.C.: Dynamic logic programming. In: Procs. of KR'98. (1998) 98–109

[3] Balduccini,M., Gelfond, M.: Logic Programs with Consistency-Restoring Rules, International Symposium on Logical Formalization of Commonsense Reasoning, AAAI 2003 Spring Symposium Series, 2003, 9-18

[4] Banti, F., Alferes, J.J., Brogi, A., Hitzler, P.: The well supported semantics for multidimensional dynamic logic programs. LPNMR 2005, LNCS 3662, Springer, 356-368

[5] Delgrande, J., Schaub, T., Tompits, H.: A Preference-Based Framework for Updating Logic Programs. LPNMR 2007: 71-83

[6] Dimopoulos, Y., Torres, A.: Graph Theoretical Structures in Logic Programming and Default Theories. Theoretical Computer Science, 170 (1996), 209-244

[7] Eiter, T., Sabbatini, G., Fink, M., Tompits, H.: On properties of update sequences based on causal rejection. Theory and Practice of Logic Programming (2002) 711–767

[8] Gärdensfors, P., Makinson, D.: Revisions of Knowledge Systems using Epistemic Entrenchment. In: Theoretical Aspects of Rationality and Knowledge, 1988, 83-95

[9] Guadarrama, J. C., Dix, J., Osorio, M., Zacaras, F.: Updates in Answer Set Programming Based on Structural Properties. Proceedings of the 7th International Symposium on Logical Formalizations of Commonsense Reasoning; 2005; 213-219

[10] Katsuno, H., Mendelzon, A.O.: On the difference between updating a knowledge base and revising it. Proc. of KR'91

[11] Kruemplemann, P.: Towards dependency semantics for conflict handling in logic programs. 2009. Submitted for publication.

[12] Leite, J.A.: Evolving Knowledge Bases: Specification and Semantics. IOS Press (2003)

[13] Osorio, M., Zepeda, C.: Update Sequences Based on Minimal Generalized Pstable Models, Proc. of MICAI 2007, 283-293

[14] Sakama, C., Inoue, K.: An abductive framework for computing knowledge base updates. TPLP 3(6): 671-713 (2003)

[15] Slota, M., Leite, J.: On Semantic Update Operators for Answer-Set Programs, ECAI 2010

[16] Šefránek, J.: Irrelevant updates of nonmonotonic knowledge bases. JELIA 2006, 426-438.

[17] Zepeda, C., Carballido, J.L., Rossainz, M., Osorio, M.: Updates based on n ASP. Proc. of MICAI 22010.

[18] Zhang, Y., Foo, N.: A Unified Framework for Representing Logic Program Updates, AAAI 2005, 707-713

[19] Zhang, Y.: Logic Program-Based Updates. ACM Transactions on Computational Logic (TOCL), Volume 7 Issue 3, July 2006