A dynamic approach for combining abstract argumentation semantics

Jérémie Dauphin, Marcos Cramer, and Leendert van der Torre

Abstract Abstract argumentation semantics provide a direct relation from an argumentation framework to corresponding sets of acceptable arguments, or equivalently to labeling functions. Instead, we study step-wise update relations on argumentation frameworks whose fixpoints represent the labeling functions on the arguments. This dynamic generalization is used to study the combination of abstract argumentation semantics. We identify different orders of granularity between update relations with the same reachable fixpoints, and provide a formal definition of the most granular update relation for a given direct semantics. We focus also on a particular case where the combination of two update relations for different direct semantics leads to an update relation for a third direct semantics.

1 Introduction

Following the methodology in non-monotonic logic, logic programming and belief revision, formal argumentation theory defines a diversity of semantics. This diversity has the advantage that a user can select the semantics best fitting her application, but it leads also to various practical challenges. First of all, how to choose among the considerable number of semantics existing in the argumentation literature for a particular application? The behaviour of semantics on examples can already be insightful, and [3] addresses the need for more systematic study and comparison of semantics by presenting a classification of argumentation semantics based on a set of principles. However, what to do when neither semantics is perfect? In general, how do we know that the currently considered set of semantics is sufficient or complete? May there be a better semantics that has not been discovered yet? How to guide

Jérémie Dauphin
University of Luxembourg, Luxembourg; e-mail: jeremie.dauphin@uni.lu

Marcos Cramer
University of Luxembourg, Luxembourg; e-mail: marcos.cramer@uni.lu

Leendert van der Torre
University of Luxembourg, Luxembourg; e-mail: leon.vandertorre@uni.lu

the search for new and hopefully better argumentation semantics? In this paper, we propose a new approach: the combination of abstract argumentation semantics.

1. How to combine two abstract semantics to yield a third semantics?
2. How to obtain the complete semantics by combining the preferred and grounded semantics?

Concerning our first research question, there are various ways in which abstract argumentation semantics can be combined. For example, in multi-sorted argumentation [6, 1, 5], one part of the framework can be evaluated according to for example grounded semantics, whereas another part of the framework is evaluated according to the preferred semantics. Another approach manipulates directly the sets of extensions. For example, the grounded and preferred can be combined by simply returning both the grounded and preferred extensions. Neither of these two approaches is very satisfactory. For multi-sorted argumentation, we need to specify explicitly which semantics must be applied to which part of the framework. For the direct combination method, the approach seems too coarse grained and the number of ways to combine semantics seems relatively limited.

We therefore introduce a dynamic approach in this paper, which is based on the labeling approach to argumentation semantics, in which the three labels *in*, *out* and *undec* are used. In our dynamic approach, we define step-wise versions of standard semantics based on epistemic labellings, which associate with each argument a nonempty *set* of labels from $\{in, out, undecided\}$. Intuitively, the set represents uncertainty about the label. We start with labeling each argument of the framework with the set $\{in, out, undecided\}$. This represents that we do not know the labeling yet. Then in each step we refine the labels by removing some of the labels. Finally we end up with a single label for each argument, and thus with a standard labeling. To represent the possibility of multiple extensions, the steps are not deterministic. The steps are represented by an abstract *update relation*, which mathematically is simply a binary relation among epistemic labelings. Note that there are many distinct update relations representing the same standard semantics, and it is this additional expressive power that we will use when combining abstract argumentation semantics. The steps can be interpreted as moves in a dialogue, or as steps in an algorithm, or as learning a framework, or otherwise. Our dynamic semantic framework does not depend on such particular interpretations.

Concerning our second research question, it is well known that the grounded semantics returns the smallest complete extension, and that the preferred semantics returns maximal complete extensions. This suggests that by combining the grounded and preferred semantics, we can again recover all complete extensions. Note that there may be complete extensions that are neither minimal nor maximal, and that it is therefore non-trivial to recover exactly the complete extensions using only the grounded and the preferred semantics, without obtaining additional extensions and without loosing any complete extensions. Though the derivation of the complete semantics from the grounded and preferred semantics does not serve any practical purpose, it serves to show that the semantics framework has a considerable expressive power to combine abstract semantics. We therefore pursue this second question.

2 Preliminaries

An argumentation framework (AF) is a directed graph $\langle A, R \rangle\rangle$, where A is called the set of arguments, and R is called the attack relation. In this work, we only consider AFs. Standard argumentation semantics come in two variants. Extension-based semantics associates with each AF a set of extensions (sets of the arguments). Labelling-based semantics attribute to each argument the label *in*, *out* or *undecided*. The two approaches are inter-definable, in the sense that an argument is labeled *in* when it is in the extension, it is labeled *out* when it is not in the extension and there is an argument in the extension attacking it, and it is *undecided* otherwise. Our dynamic approach uses an epistemic labelling, which associates with each argument a nonempty *set* of labels. Intuitively, the set represents uncertainty about the label.

We assume familiarity with 3-labeling semantics of argumentation frameworks (AFs) as defined in [2]. Note that we will make use of the multi-labeling approach, where a set of labels is assigned to each argument. This set represents the possible labels for a given argument. The standard approach corresponds to the case where arguments are given singleton sets as labels.

We will also make use of the notions of *transitive closure* of a relation and *restriction* of a relation to a subset of its domain.

Definition 1. Let *rel* be a relation. We define the *transitive closure* of *rel* to be the smallest set rel^* such that $rel \subseteq rel^*$ and if $(a,b),(b,c) \in rel^*$, then $(a,c) \in rel^*$.

Definition 2. Let A, B be sets, $A' \subseteq A$ and R a relation from A to B. We define the *restriction* of R to A' to be:

$$R\downarrow_{A'} = \begin{cases} \{(a,b) \in R \mid a,b \in A'\} & \text{if } A = B \\ \{(a,b) \in R \mid a \in A'\} & \text{otherwise} \end{cases}$$

3 Update semantics

Standard labeling semantics provide a direct relation between an argumentation framework and a set of labeling functions, which attach to each argument exactly one label. We will now define update semantics, which formalize the idea that the final labelings can be determined in a step-wise fashion.

Notice that it makes little sense to separate the labeling function from the underlying framework, as the labeling is meaningless without it. We will hence consider pairs of argumentation framework and labeling functions, where frameworks with no labels correspond to a pair of said framework with a trivial labeling function which assigns to every argument the same initial value, which in our case of epistemic labeling will be the whole set of possible labels.

We define $\mathbb{L} = \{in, out, undec\}$ to be the set of possible *labels*.

Definition 3. We define a *labeled argumentation framework* (LAF) to be a pair $(\langle A, R \rangle, Lab)$ where $\langle A, R \rangle$ is a finite argumentation framework and *Lab* a function from A to $\mathscr{P}(\mathbb{L}) \setminus \{\emptyset\}$, called an *epistemic labeling*. Additionally, let $\mathbb{F}$ be the class of all labeled argumentation frameworks.

Observe that a labeling function cannot assign the empty set of labels to an argument, as the set of labels represents the possible final labels for that argument, and thus the empty set would mean that no label can be attached to it, which prevents us from having a final labeling for the framework. We might use the term *unlabeled*, by which we mean that all labels are still considered possible, and thus the argument is attached the entire set of possible labels by the multi-labeling function.

We then introduce the notions of *initial* and *final* labeled frameworks, which should correspond to the starting and end-points of a labeling process. In an initial LAF, every label is possible for each argument, while in a final LAF, every argument is assigned a singleton set of labels, representing the fact that a unique label has been selected.

Definition 4. Let $F = (\langle A,R \rangle, Lab)$ be a LAF. If for all $a \in A$, $Lab(a) \in \{\{in\},\{out\}, \{undec\}\}$, we say that F is *final*. If for all $a \in A$, $Lab(a) = \mathbb{L}$, we say that F is *initial*.

We now define a precision ordering on the LAFs based on the subset relation between the argument multi-labels, such that the final LAFs are the most precise and the initial LAFs are the least precise. Note however that only LAFs with the same underlying AF are comparable.

Definition 5. Let $F = (\langle A,R \rangle, Lab)$ and $F' = (\langle A',R' \rangle, Lab')$ be two labeled argumentation frameworks. We say that F is *at least as precise* as F' ($F \geq_p F'$), iff $\langle A,R \rangle = \langle A',R' \rangle$, and for all $a \in A$, $\emptyset \subset Lab(a) \subseteq Lab'(a)$. We say that F is *more precise* than F' ($F >_p F'$) iff $F \geq_p F'$ and $F \neq_p F'$.

We will now define the central notion of this paper, namely *update relations*, i.e. relations between LAFs which, starting from an initial LAF, monotonically increase precision, until a fixpoint is reached, at which point the LAF should be final and correspond to a desired output.

Definition 6. We say that $upd : \mathbb{F} \times \mathbb{F}$ is an *update relation* iff:
- for all $F' \in \mathbb{F}$ such that $upd(F,F')$, $F' \geq_p F$;
- if $upd(F,F)$, then F is final.

We now define correspondence between update relations and direct semantics that formalizes the idea that an update relation can be viewed as a step-wise procedure that gives rise to a certain direct semantics. For this we first need some auxiliary definitions.

Definition 7. Let *Rel* be a relation on $\mathbb{F}$ and F an LAF. We say that F is *reachable* in *Rel* iff there exists an initial LAF F_i such that there is a path in *Rel* from F_i to F. We say that F is a *reachable fixpoint* in *Rel* iff F is reachable in *Rel* and $(F,F) \in Rel$.

Definition 8. Given an AF $\langle A,R \rangle$ and a 3-labeling L of $\langle A,R \rangle$, define the epistemic labeling $T(L)$ by $T(L)(a) := \{L(a)\}$ for all $a \in A$.

Definition 9. Let *upd* be an update relation and *sem* a semantics. We say that *upd gives rise* to *sem* iff for each 3-labelling *Lab* of $\langle A,R \rangle$, $(\langle A,R \rangle, T(Lab))$ is a reachable fixpoint in *upd* iff *Lab* is a *sem* labeling of $\langle A,R \rangle$.

The following theorem, which easily follows from Definition 6, provides a simple way of combining two given update relations to yield a third update relation:

Theorem 1. *If upd_1 and upd_2 are update relations, than $upd_1 \cup upd_2$ is an update relation.*

In Section 4.1 we will present an example where combining two update relations with a union operation gives us not only the union of the final labelings reachable by either of them, but also additional labelings. This means that the semantics that $upd_1 \cup upd_2$ gives rise is not necessarily induced by the semantics that upd_1 and upd_2 separately give rise to.

We are now interested in the comparison of updates in terms of precision increase per step, i.e. in the granularity of update relations. The idea is that an update relation is more granular than another if it takes more steps to reach its final LAFs. First of all, notice that such a comparison only makes sense for updates which output the same final LAF, i.e. updates which give rise to the same semantics.

Definition 10. Let upd be an update relation. We define the *restriction of upd to relevant paths* ($\overline{upd}$) to be the set of pairs in upd that are in some upd-path from an initial to a final LAF.

Definition 11. Let upd_1 and upd_2 be two update relations. We say that upd_1 is *at least as fined-grained* as upd_2 ($upd_1 \geq_g upd_2$) iff $\overline{upd_1}^* \supseteq \overline{upd_2}$.

We then abstractly define the *most fine-grained* update relation for a given labeling semantics.

Definition 12. Let sem be a labeling semantics. We define mfg_{sem} to be the smallest update relation such that for all update relations upd that give rise to sem, we have $mfg_{sem} \geq_g upd$.

Lemma 1. *For every standard semantics, there exists a unique mfg_{sem}.*

Proof: Define mfg_{sem} as follows: $(F, F') \in mfg_{sem}$ iff either $F = F'$ is a sem labeling, or the following three properties are satisfied:

- $F' >_p F$;
- $\nexists F''$ such that $F' >_p F'' >_p F$;
- there exists a final F_f which is a sem labeling such that $F_f \geq_p F'$.

By definition, mfg_{sem} includes all possible links in any relevant path from an initial to a final LAF which encompasses a sem labeling. Hence, for any update relation upd which gives rise to sem, $\overline{mfg_{sem}}^* \supseteq \overline{upd}$. Also, mfg_{sem} includes by definition only pairs which are on a relevant path, as the first alternative adds the endpoints of these paths and the third item of the second alternative ensures that the pairs are on a relevant path. The first and second items of the second alternative ensure also that only the minimal amount of pairs are added, making mfg_{sem} as small as possible. □

4 Case analysis: Combining preferred and grounded

4.1 The algorithmic approach

Let us now have a look at an example of an update relation, which gives rise to the grounded labeling. This update relation first identifies the arguments which are only being attacked by arguments which are already labeled $\{out\}$, label them as $\{in\}$ and any argument they attack as $\{out\}$, and then repeat this process until no arguments can be further labeled, at which point it will label all remaining arguments as $\{undec\}$.

Definition 13. For any labeled argumentation framework $F = (\langle A,R\rangle, Lab)$, we define the set of *unattacked arguments* to be $unattacked(F) = \{a \in A \mid Lab(a) \supsetneq \{in\} \wedge \forall b \in A.((b,a) \in R \rightarrow Lab(b) = \{out\})\}$.

Definition 14. We define $step\_grnd : \mathbb{F} \times \mathbb{F}$ to be the relation such that $((\langle A,R\rangle, Lab), (\langle A,R\rangle, Lab')) \in step\_grnd$ iff one of the following conditions holds:

- $unattacked((\langle A,R\rangle, Lab)) \neq \emptyset$, and $(\langle A,R\rangle, Lab')$ is the least precise LAF that is more precise than $(\langle A,R\rangle, Lab)$ such that for all $a \in unattacked((\langle A,R\rangle, Lab))$, $Lab'(a) = \{in\}$ and for all $c \in A$ such that $(a,c) \in R$ and $out \in Lab(c)$, $Lab'(c) = \{out\}$.
- $unattacked((\langle A,R\rangle, Lab)) = \emptyset$, there is an $a \in A$ such that $Lab(a) \supsetneq \{undec\}$, and $(\langle A,R\rangle, Lab')$ is the least precise LAF that is more precise than $(\langle A,R\rangle, Lab)$ such that for all $a \in A$ such that $Lab(a) \supsetneq \{undec\}$, $Lab'(a) = \{undec\}$.
- $(\langle A,R\rangle, Lab) = (\langle A,R\rangle, Lab')$ is a final LAF.

Note that before labeling arguments out, we ensure that it is a possibility, e.g. by having the condition $out \in Lab(c)$ in the first item of Definition 14. While this requirement will straightforwardly be fulfilled in any reachable LAF, it is required to ensure that the increase in precision is satisfied even for those LAFs that are not reachable from an initial LAF.

The following lemma now easily follows from the above definition:

Lemma 2. *$step\_grnd$ is an update relation.*

The following theorem states that $step\_grnd$ does indeed have the intended property that it gives rise to the grounded labeling:

Theorem 2. *$step\_grnd$ gives rise to the grounded semantics.*

Proof sketch. One can easily see that whenever $step\_grnd$ changes the label of an argument a to $\{in\}, \{out\}$ or $\{undec\}$, argument a is legally labeled $\{in\}, \{out\}$ or $\{undec\}$ respectively. Thus the final labeling reachable in $step\_grnd$ is a complete labeling. To show that the final labeling reachable in $step\_grnd$ is the complete labeling that maximizes $undec$, suppose that there is some $A' \subseteq A$ and some complete labeling Lab of $\langle A,R\rangle$ such that for all $a \in A'$, $Lab(a) = undec$. It is now enough to show that $step\_grnd$ never labels any $a \in A'$ $\{in\}$ or $\{out\}$. Consider for a proof by contradiction the first step where $step\_grnd$ does label some $a \in A'$ $\{in\}$. Since a is legally labeled *undec* in *Lab*, some $a' \in A'$ must attack a, so by Definitions 13 and 14, a' must already be labeled $\{out\}$ in a previous step, which is a contradiction. □

Let us now examine $step\_pref$, a similar update relation which computes the preferred labelings. For this, we first define the notion of minimal non-trivial admissible sets of arguments.

Definition 15. Let $F = (\langle A, R \rangle, Lab)$ be a labeled argumentation framework. We define $min\_adm(F) \subseteq \mathscr{P}(A)$ to be the set of all minimal subsets S of A that satisfy the following conditions:

- $S \neq \emptyset$;
- for all $a \in S$, $Lab(a) \supsetneq \{in\}$;
- for all $a, b \in S$, $(a, b) \notin R$;
- for all $a \in S$ and $b \in A$ such that $Lab(b) \neq \{out\}$ and $(b, a) \in R$, there exists $a' \in S$ such that $(a', b) \in R$.

So the function $min\_adm(F)$ returns all minimal non-empty admissible sets of arguments whose label could still be changed to $\{in\}$. The update relation $step\_pref$ proceeds with a process similar to the one in the $step\_grnd$ update, iteratively labeling $\{in\}$ all arguments with all attackers $\{out\}$, and then labeling all arguments attacked by those as $\{out\}$. The difference lies in the case where $unattacked(F)$ is empty, where the preferred update relation looks for minimal non-trivial admissible sets, label them $\{in\}$ and arguments they attack $\{out\}$.

Definition 16. We define $step\_pref : \mathbb{F} \times \mathbb{F}$ to be the relation such that $((\langle A, R \rangle, Lab), (\langle A, R \rangle, Lab')) \in step\_pref$ iff one of the following conditions holds:

- $unattacked((\langle A, R \rangle, Lab)) \neq \emptyset$, and $(\langle A, R \rangle, Lab')$ is the least precise LAF that is more precise than $(\langle A, R \rangle, Lab)$ such that for all $a \in unattacked((\langle A, R \rangle, Lab))$, $Lab'(a) = \{in\}$ and for all $c \in A$ such that $(a, c) \in R$ and $out \in Lab(c)$, $Lab'(c) = \{out\}$.
- $unattacked((\langle A, R \rangle, Lab)) = \emptyset$, and for some $S \in min\_adm(F)$, $(\langle A, R \rangle, Lab')$ is the least precise LAF that is more precise than $(\langle A, R \rangle, Lab)$ such that for all $a \in S$, $Lab'(a) = \{in\}$ and for all $c \in A$ such that $(a, c) \in R$ and $out \in Lab(c)$, $Lab'(c) = \{out\}$.
- $unattacked((\langle A, R \rangle, Lab)) = min\_adm(F) = \emptyset$, and there is an $a \in A$ such that $Lab(a) \supsetneq \{undec\}$, and $(\langle A, R \rangle, Lab')$ is the least precise LAF that is more precise than $(\langle A, R \rangle, Lab)$ such that for all $a \in A$ such that $Lab(a) \supsetneq \{undec\}$, $Lab'(a) = \{undec\}$.
- $(\langle A, R \rangle, Lab) = (\langle A, R \rangle, Lab')$ is a final LAF.

The following lemma now easily follows from the above definition:

Lemma 3. *$step\_pref$ is an update relation.*

The following theorem, which can be proved in a similar way as Theorem 2, states that $step\_pref$ has the intended property that it gives rise to the preferred labeling:

Theorem 3. *$step\_pref$ gives rise to the preferred semantics.*

We now find the interesting result that combining these two update relations with a union operation gives us not only the union of the final labelings reachable by either of them, but also the complete labelings which are neither grounded nor preferred:

Theorem 4. *$step\_grnd \cup step\_pref$ gives rise to the complete semantics.*

Proof. This follows from Theorems 3 and 4 in [4].

Example 1. Let us examine the initial LAF $F = (\langle A, R\rangle, Lab)$ where $A = \{a, b, c, d\}$, $R = \{(a,b),(b,a),(b,c),(c,d),(d,c)\}$. Since $unattacked(F) = \emptyset$, $step\_grnd$ will send F to the fixpoint where all arguments are labeled $\{undec\}$. This is depicted in Fig. 1.

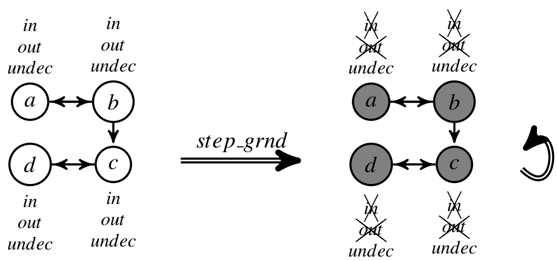

Fig. 1 Example path from the initial LAF F to the corresponding final LAF in $step\_grnd$.

Let us now consider the same LAF F under the $step\_pref$ update relation this time. Again, $unattacked(F) = \emptyset$, but $min\_adm(F) = \{\{a\},\{b\},\{d\}\}$. The relation hence branches out in three paths. Let us focus the path with $\{a\}$. So the relation $step\_pref$ sends F to the LAF F_{pref1} where a is $\{in\}$ and b is $\{out\}$, as depicted in Fig. 2. $unattacked(F_{pref1}) = \emptyset$, but $min\_adm(F_{pref1}) = \{\{c\},\{d\}\}$, which gives us once again two possible directions in which to branch out. We will examine the one which selects $\{c\}$. This then gives us the final fixpoint $F_{pref2} = (\langle A, R\rangle, Lab_{pref2})$, where $Lab_{pref2}(a) = Lab_{pref2}(c) = \{in\}$ and $Lab_{pref1}(b) = Lab_{pref1}(d) = \{out\}$.

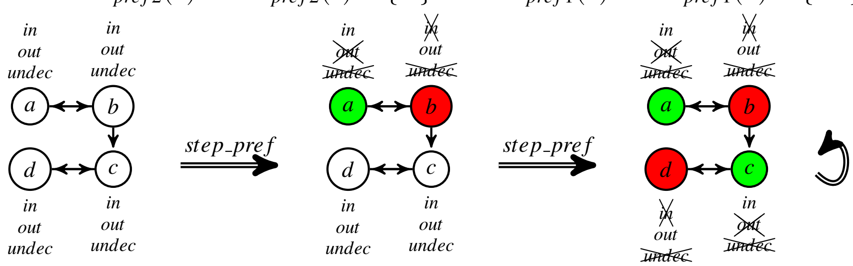

Fig. 2 Example path from the initial LAF F to one of the corresponding final LAFs in $step\_pref$.

We now consider the union of both relations. We can first send F to F_{pref1} using the same step from $step\_pref$ as above. However this time we can apply $step\_grnd$ to F_{pref1}, and since $unattacked(F_{pref1} = \emptyset$, the remaining arguments c and d are assigned the $\{undec\}$ label, sending F_{pref1} to the fixpoint F_{comp}, where a is $\{in\}$, b is $\{out\}$ and c,d are $\{undec\}$. Notice that F_{comp} corresponds to a complete labeling of F which is neither preferred nor grounded. This situation is depicted in Fig. 3.

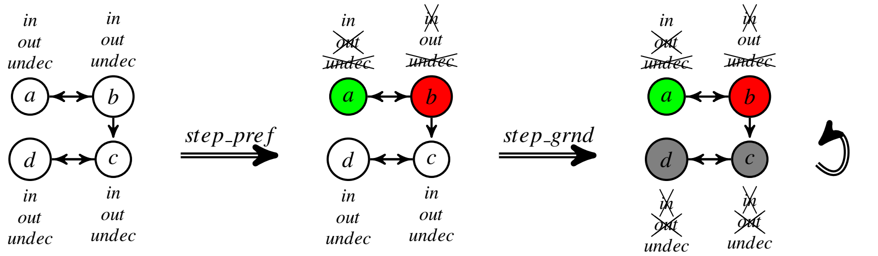

Fig. 3 Example path from the initial LAF F to one of the corresponding final LAFs in $step\_grnd \cup step\_pref$ which neither update can reach by itself.

4.2 The semantic approach

We will focus on the most fine-grained update relations for the preferred and grounded semantics, and aim at combining them in order to obtain an update relation for the complete semantics again. However, if we were to attempt to combine mfg_{pref} and mfg_{grnd} by simply taking their union, as we have done in the algorithmic approach, it follows from their definition that we would simply obtain as reachable fixpoints the labelings which are either preferred or grounded. The main issue is that mfg_{pref} and mfg_{grnd} are not applicable to LAFs which do not agree with some final LAF of that semantics. Hence, once neither mfg_{pref} nor mfg_{grnd} allow us to get closer to a desired complete labeling, we will focus on a particular sub-framework and draw analogies with another framework which also contains that sub-framework and where it behaves similarly as in the original LAF. If a set of conditions are met, we will allow for a step made in such a parallel framework to be imported into the original LAF. The conditions are there to ensure that the two frameworks agree on the behavior of the sub-framework. We split the original framework into three parts, based on sets of arguments: S, the arguments we will focus on; I, a set of arguments which already have a maximally precise label, i.e. a singleton, and separate the set S from the rest of the framework; and lastly $A \setminus (S \cup I)$, the rest of the framework, on which the two frameworks may differ.

Definition 17. Let $F = (\langle A, R \rangle, Lab)$ be a LAF and $S \subseteq A$. We define the *sub-framework* of F generated by S to be $Sub(F,S) = (\langle S, R\downarrow_S \rangle, Lab\downarrow_S)$.

Definition 18. Let upd_1 and upd_2 be two update relations. We define the *combination* of upd_1 and upd_2 ($upd_1 \uplus upd_2$) as the smallest relation such that:

1. $upd_1 \uplus upd_2 \supseteq upd_1$;
2. $upd_1 \uplus upd_2 \supseteq upd_2$;
3. $((\langle A,R\rangle, Lab),(\langle A,R\rangle, Lab')) \in upd_1 \uplus upd_2$ if there exist disjoint sets $S, I \subseteq A$ s.t.:

   a. $Lab' \downarrow_S \neq Lab \downarrow_S$
   b. there is an LAF $F_2 = (\langle A_2, R_2\rangle, Lab_2)$ and an LAF $F_2' = (\langle A_2, R_2\rangle, Lab_2')$ s.t. $(F_2, F_2') \in upd_1 \cup upd_2$ and $Sub(F_2, S \cup I) = Sub(F, S \cup I)$;
   c. $\forall s \in S$, $\forall a \in A \setminus (I \cup S)$, $(s,a),(a,s) \notin R, R_2$;
   d. $\forall a \in I$, $Lab(a) = Lab_2(a)$ is a singleton;
   e. $\forall a \notin S$, $Lab(a) = Lab'(a)$;
   f. $Sub(F, A \setminus S)$ is reachable by $upd_1 \uplus upd_2$;
   g. $Lab' \downarrow_S = Lab_2' \downarrow_S$.

4. if F is final and reachable by $upd_1 \uplus upd_2$, then $(F,F) \in upd_1 \uplus upd_2$.

Theorem 5. *$mfg_{pref} \uplus mfg_{grnd}$ gives rise to the complete semantics.*

Proof sketch. 1. Every complete labeling is reachable: Let AF be an argumentation framework, and let L be the complete labeling of AF that we want to reach. First do preferred update steps to ensure that every argument labeled $\{in\}$ or $\{out\}$ in L is labeled $\{in\}$ or $\{out\}$ respectively. Note that at this point, we want to update the labeling further in such a way that all argument currently labeled $\{in, out, undec\}$ are finally labeled $\{undec\}$. Define S to be the set of all arguments currently labeled $\{in, out, undec\}$, and define I to be the set of all arguments labeled $\{out\}$. It can be easily checked that S does not attack anything in $A \setminus (S \cup I)$, as otherwise there would be an argument labeled *undec* in L that attacks an argument labeled *in* in L. Also, similar reasoning about L shows that all the $\{out\}$-labeled argument are legally *out* in $A \setminus I$. So S and I satisfy criteria d, e and f of Definition 18. Now consider the argumentation framework AF' which is like AF only that all attacks from $\{out\}$-labeled arguments to $\{in\}$-labeled arguments are removed. Now it is easy to see that all $\{in, out, undec\}$-labeled arguments are part of a cycle and can therefore be labeled just $\{undec\}$ by a series of grounded update steps in AF' that can be taken over to AF.

2. Every reachable fixpoint is complete: Let $F = (\langle A,R\rangle, Lab)$ be a reachable LAF in $mfg_{pref} \uplus mfg_{grnd}$. We show by induction that there exists a final complete LAF which is at least as precise as F. Suppose there exists a final complete LAF $F_f = (\langle A,R\rangle, Lab_f)$ which is at least as precise as F, and $(F,F') \in mfg_{pref} \uplus mfg_{grnd}$. We distinguish three cases:
   $(F,F') \in mfg_{pref}$. Then, by the definition of mfg_{pref}, there exists a final LAF which represents a preferred labeling of $\langle A,R\rangle$ and is at least as precise as F'. Since preferred labelings are also complete, we are done.
   $(F,F') \in mfg_{grnd}$. Similarly to the case above, it follows from the definition of mfg_{grnd} that there exists a complete final LAF which is at least as precise as F'.
   $(F,F') \notin mfg_{pref} \cup mfg_{grnd}$. Then, it must be that there exist two disjoint sub-frameworks F_S and F_I of F, as well as $(F_2, F_2') \in mfg_{pref} \cup mfg_{grnd}$ satisfying the conditions of Def. 18, third item. So there exists a final LAF $F_{f2} =$

$(\langle A_2, R_2 \rangle, Lab_{f2})$ which is complete and at least as precise as F_2'. We construct the final LAF $F_c = (\langle A, R \rangle, Lab_c)$ where $\forall a \in A \setminus S, Lab_f(a) = Lab_c(a)$ and $\forall a \in S, Lab_f(a) = Lab_{f2}(a)$. Now one can easily show that all arguments are legally labeled in F_c and so it is a complete labeling. Hence, there exists a final complete LAF which is at least as precise as F'. □

Example 2. Let us consider the same LAF F as in Example 1. We can apply steps from mfg_{grnd} to reach the LAF $F' = (\langle A, R \rangle, Lab')$, where $Lab'(a) = \{in, undec\}$, $Lab'(b) = \{out, undec\}$ and $Lab'(c) = Lab'(d) = \{undec\}$, as depicted in Fig. 4, since F' is less precise than the final grounded labeling for F where all arguments are $\{undec\}$. From here, we can set $S = \{a, b\}$ and $I = \{c\}$, and observe that in the framework $F_2 = (\langle A, (R \cup \{d, d\}) \setminus \{c, d\} \rangle, Lab')$, there exists a more precise final preferred labeling where a is $\{in\}$ and b is $\{out\}$, as depicted in Fig. 5. F_2 satisfies the conditions for the third item of Def. 18 and we can thus import these steps and apply them to F', giving us the final complete labeling where a is $\{in\}$, b is $\{out\}$ and c, d are $\{undec\}$, as shown in Fig. 6.

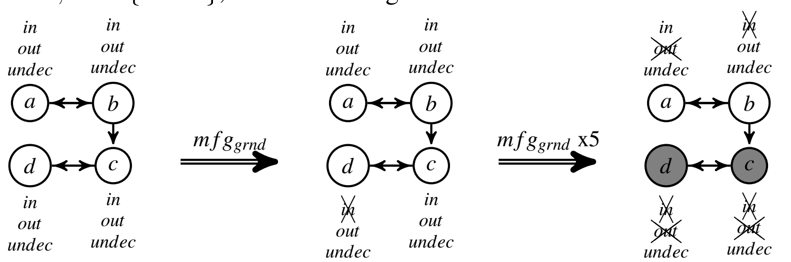

Fig. 4 Example path from the initial LAF F to an intermediary LAF F' in $mfg_{pref} \uplus mfg_{grnd}$.

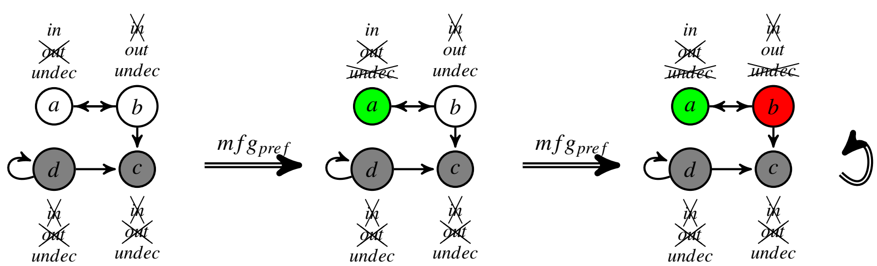

Fig. 5 Example path on a parallel F_2 framework with $S = \{a, b\}$ and $I = \{c\}$, where mfg_{pref} is applicable.

5 Conclusion and future work

In this paper we introduce a dynamic approach to combine two abstract semantics to yield a third one. In particular, we provide a formal environment for the analysis of step-wise relations between labeled framework with an increase in the label precision, whose reachable fixpoints correspond to some standard direct semantics.

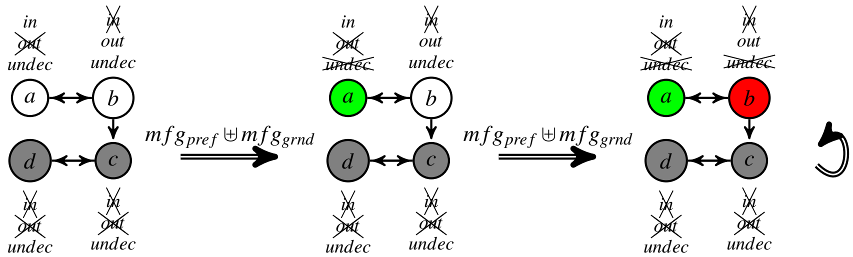

Fig. 6 Importing the steps made in Fig. 5 into F' allows us to reach a complete labeling which is neither grounded nor preferred.

We define and discuss two approaches to combining two given update semantics to yield a third update semantics, and then examine how these two approaches allow us to obtain update relations for the complete labeling by combining update relations for the preferred and grounded labelings.

Our paper gives rise to various topics for further research. Concerning the combination of abstract argumentation semantics, many questions remain. Though we introduced our update relations to combine abstract argumentation semantics, we believe that this dynamic semantics framework can be used for other applications as well. Most importantly, one of the main challenges in formal argumentation is the gap between graph based semantics and dialogue theory. Our more dynamic semantics framework may be used to decrease or even close the gap. In particular, in dialogue each statement may increase the knowledge and thus the set of arguments of participants. This is also related to the formalisation of learning in the context of formal argumentation. Moreover, an important approach in argumentation semantics is the SCC recursive scheme. This scheme can be represented naturally using update relations. Various algorithms have been proposed for abstract argumentation semantics, and these algorithmic approaches may be expressed naturally using update relations. Finally, the principle based analysis of abstract argumentation semantics can be extended to the more fine grained update relations.

References

1. Ryuta Arisaka, Ken Satoh, and Leendert van der Torre. Anything you say may be used against you in a court of law. In *Artificial Intelligence and the Complexity of Legal Systems (AICOL)*. Springer, 2018.
2. Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.
3. Pietro Baroni and Massimiliano Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171(10-15):675–700, 2007.
4. Jeremie Dauphin and Claudia Schulz. Arg Teach – A Learning Tool for Argumentation Theory. In *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*, pages 776–783. IEEE, 2014.
5. Massimiliano Giacomin. Handling Heterogeneous Disagreements Through Abstract Argumentation. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 3–11. Springer, 2017.
6. Tjitze Rienstra, Alan Perotti, Serena Villata, Dov M Gabbay, and Leendert van der Torre. Multi-sorted argumentation. In *International Workshop on Theorie and Applications of Formal Argumentation*, pages 215–231. Springer, 2011.