A Unifying Four-State Labelling Semantics for Bridging Abstract Argumentation Frameworks and Belief Revision*

Stefano Bistarelli[1[0000−0001−7411−9678]] and Carlo Taticchi[1[0000−0003−1260−4672]]

University of Perugia, Perugia, Italy
[firstname.lastname]@unipg.it

Abstract. In many formalisms extending Dung's Abstract Argumentation Frameworks (AFs), arguments are not always "present". In timed AFs, for instance, arguments are only available in precise intervals of time, as they can appear and disappear in an intermittent manner; in incomplete AFs, both attacks and arguments can be absent; in constellation probabilistic AFs (attacks and) arguments have a probability to be present or not, and possible worlds are generated for the computation of the semantics. We review current approaches and propose a four-state labelling semantics to take in account such absent/unknown state of an argument. The four labels we use can be traced to the states a belief can assume, allowing us to also define operations related to belief manipulation, like expansion contraction and revision. We also discuss how labels/states of arguments in an AFs can be modified by using belief revision operations.

Keywords: Argumentation Theory · Four-State Labelling · Non-Monotonic Reasoning · Belief Revision · AGM.

1 Introduction

A labelling for AFs has been proposed by Caminada [11] to cope with the issue of reinstatement, namely the phenomenon for which defended arguments can be considered accepted. Such a labelling consists of a function assigning three different labels (`IN`, `OUT` and `UNDEC`) to arguments of a framework according to a set of rules. Only `IN` and `OUT` arguments can be directly labelled, while the `UNDEC` label is assigned to arguments which can be neither `IN` nor `OUT`. A distinction between arguments to be ignored and arguments whose acceptability cannot be established is made in [17] trough a four-state labelling obtained starting from two labels (+ and −) that can be assigned to arguments. Assigning both labels corresponds to identify an undecided argument, while not assigning any label means that the argument will be ignored. In this setting, the authors introduce partial (not considering arguments with a +/− label) and total[1] labellings. However, they only consider total ones for defining complete semantics. In other words, no argument in a complete labelling can be left unspecified. The notion of UNDEC argument is also revised in [5], where the authors provide an approach which explicitly expresses the reason why acceptability cannot be decided: as in [17], a distinction is made between arguments we "don't care" about and those to whom we "do not know" what label to assign. However, no modifications on the rules to assign IN/OUT labels are proposed. In [18], an OFF label is introduced, alongside IN, OUT and UNDEC, to model incomplete AFs. In such kind of AFs, part of the information can be excluded from the computational process which leads to the selection of accepted arguments. The OFF label denotes, in particular, arguments that we do not want to consider. Four labels are also used in [2] to define a labelling semantics that allows conflicts among accepted arguments. In addition to the classical IN and OUT labels, the author introduces a BOTH label for arguments which could be both accepted and rejected, and a NONE indicating, instead, lack of information.

In this paper, we provide a unifying representation for (temporarily) excluded arguments or for arguments we want to ignore. We use a partial labelling with four labels to identify the possible states of arguments, namely IN for accepted, OUT for rejected, DK for arguments we don't know how to label, and DC for arguments we don't care about (because not adopted in an AF or just ignored by the user). The introduced four-state labelling can be mapped with belief states in AGM Theory [1], where accepted/rejected beliefs corresponds to IN/OUT arguments, the undetermined state coincides with having assigned a DC label, and the notion of inconsistency boils down to the DK label. Such a correspondence allows the use of AFs as a communication mean between intelligent agents involved in complex forms of interaction. In particular, acceptance states can be used to reason about shared knowledge in order to pursue different goals. For instance, agents involved in a debate affirm some belief and defends it from the attacks of other parts; negotiating agents need to find a common agreement that is beneficial to all; an agent with the goal of persuading its opponents has to both defend its position from the attacks of the other agents and defeat all the arguments against its proposal. Operations needed for the implementation of such kind of interactions must be able to modify the knowledge base of the involved agents. We also discuss how labels/states of arguments in an AFs can be modified by using belief revision operations.

2 Argumentation Theory and Labellings

In this section we recall the formal definition of AF and the related semantic [16], together with the notion of labelling and labelling-based semantics introduced in the literature.

[1] In the original work, total labellings are called complete. Here we use a different term in order not to raise ambiguity with the complete semantics.

Definition 1 (Abstract Argumentation Framework). *Let $\mathcal{U}$ be the set of all available arguments*[2]*, which we refer to as the "universe". An Abstract Argumentation Framework is a pair $\langle \mathcal{A}, \mathcal{R} \rangle$ where $\mathcal{A} \subseteq \mathcal{U}$ is a set of adopted arguments and $\mathcal{R}$ is a binary relation on $\mathcal{A}$.*

Consider two arguments a, b belonging to an AF. We denote with $(a, b) \in \mathcal{R}$ (or simply $a \rightarrow b$) an attack from a to b; we can also say that b is defeated by a. We define the sets of arguments that attack (and that are attacked by) another argument as follows.

Definition 2 (Attacks). *Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an AF, $a \in \mathcal{A}$ and $A \subseteq \mathcal{A}$. We define the sets $a^+ = \{b \in \mathcal{A} \mid a \rightarrow b\}$, $a^- = \{b \in \mathcal{A} \mid b \rightarrow a\}$, $A^+ = \cup\{a^+ \mid a \in A\}$ and $A^- = \cup\{a^- \mid a \in A\}$.*

In order for b to be acceptable, we require that every argument that defeats b is defeated in turn by some other argument of the AF. More formally, we have the following definition.

Definition 3 (Acceptable argument). *Given an AF $\langle \mathcal{A}, \mathcal{R} \rangle$, an argument $a \in \mathcal{A}$ is acceptable with respect to $D \subseteq \mathcal{A}$ if and only if $\forall b \in \mathcal{A}$ such that $b \in a^-$ (b is attacking a) $\exists d \in D$ such that $d \in b^-$ (d is attacking b) and we say that a is **defended** by D.*

Using the notion of defence as a criterion for distinguishing acceptable arguments in the framework, one can further refine the set of selected "good" arguments through semantics.

Definition 4 (Extension-based semantics). *Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an AF. A set $E \subseteq \mathcal{A}$ is conflict-free if and only if there are no $a, b \in E$ such that $(a, b) \in \mathcal{R}$. A conflict-free subset E is then*

- *admissible, if each $a \in E$ is defended by E;*
- *complete, if it is admissible and $\forall a \in \mathcal{A}$ defended by E, $a \in E$;*
- *stable, if $E \cup E^+ = \mathcal{A}$;*
- *preferred, if it is admissible and it is maximal (with respect to set inclusion);*
- *grounded, if it is complete and it is minimal (with respect to set inclusion).*

In Figure 1, we show a framework F for which we compute the set of extensions $S_\sigma(F)$, where σ is a semantics among conflict-free, admissible, complete, stable, preferred and grounded semantics (abbreviated with cf, adm, com, stb, prf and gde). We have: $S_{cf}(F) = \{\{\}, \{a\}, \{b\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}, \{b, d\}\}$, $S_{adm}(F) = \{\{\}, \{a\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}\}$, $S_{com}(F) = \{\{a\}, \{a, c\}, \{a, d\}\}$, $S_{prf}(F) = \{\{a, c\}, \{a, d\}\}$, $S_{stb}(F) = \{\{a, d\}\}$, and $S_{gde}(F) = \{\{a\}\}$.

The work in [11] introduces the notion of reinstatement labelling, partitioning arguments of an AF into three subsets, each representing a different degree of acceptance. Below, we report the labelling function and the characterisation for the various semantics.

[2] The set $\mathcal{U}$ is not present in the original definition; we introduce it to model arguments not adopted in $\mathcal{A}$, that could be added with dynamic operations [7, 8, 10, 14].

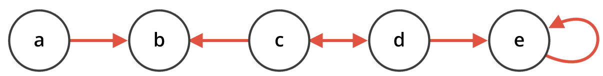

Fig. 1. Example of abstract argumentation framework.

Definition 5 (IN-OUT-UNDEC labelling for AFs [11, Definition 5]). *Let $F = \langle \mathcal{A}, \mathcal{R} \rangle$. An IN-OUT-UNDEC labelling L of F is a total function $L : \mathcal{A} \to \{\texttt{IN}, \texttt{OUT}, \texttt{UNDEC}\}$ satisfying the following rules $\forall a \in \mathcal{A}$:*

- $L(a) = \texttt{IN} \iff \forall b \in \mathcal{A} \mid (b, a) \in \mathcal{R}.L(b) = \texttt{OUT}$ *and*
- $L(a) = \texttt{OUT} \iff \exists b \in \mathcal{A} \mid (b, a) \in \mathcal{R} \wedge L(b) = \texttt{IN}$

In other words, an argument a is labelled IN when all its attackers are labelled OUT, and it is labelled OUT when at least an IN argument attacks it. In all other cases, a is labelled UNDEC. In Figure 2 we show an example of IN-OUT-UNDEC labelling on an AF in which arguments a and c highlighted in green are IN, red ones (b and d) are OUT, and the the yellow argument e (that attacks itself) is UNDEC.

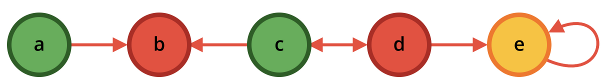

Fig. 2. Example of IN-OUT-UNDEC labelling.

Given an IN-OUT-UNDEC labelling L, it is possible to identify a correspondence between sets of IN arguments and extensions of the semantics given in Definition 4. A *labelling-based* semantics [3, 4] associates an AF with a subset of all the possible labellings of a certain semantics. The labelling of Definition 5 coincides with a complete extension, while other semantics can be obtained by introducing additional conditions [11].

The definition for an admissible labelling given in [12] allows arguments only attacked by OUT to be left UNDEC. We show an example in Figure 3.

Definition 6 (Admissible IN-OUT-UNDEC labelling for AFs [12, Definition 4]). *Let $F = \langle \mathcal{A}, \mathcal{R} \rangle$. An admissible IN-OUT-UNDEC labelling L_{adm} of F is a total function $L_{adm} : \mathcal{A} \to \{\texttt{IN}, \texttt{OUT}, \texttt{UNDEC}\}$ satisfying the following rules $\forall a \in \mathcal{A}$:*

- $L(a) = \texttt{IN} \implies \forall b \in \mathcal{A} \mid (b, a) \in \mathcal{R}.L(b) = \texttt{OUT}$ *and*
- $L(a) = \texttt{OUT} \iff \exists b \in \mathcal{A} \mid (b, a) \in \mathcal{R} \wedge L(b) = \texttt{IN}$

Different definitions of labellings, as for instance those given in [13, 17], allow accepted arguments to attack both rejected and undecided ($L(a) = \texttt{OUT} \implies \exists b \in \mathcal{A} \mid (b, a) \in \mathcal{R} \land L(b) = \texttt{IN}$). However, nothing changes in terms of extensions, since the set of accepted arguments remains the same.

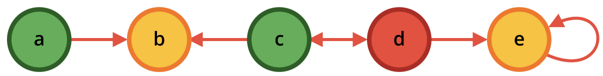

Fig. 3. Example of admissible IN-OUT-UNDEC labelling.

The authors of [17] propose a labelling in which arguments can be assigned up to two labels among $+$ and $-$. The combination of these labels results in four possible acceptance states for arguments of an AF.

Definition 7 (+-– labelling for AFs [17, Definition 3]). *Let $F = \langle \mathcal{A}, \mathcal{R} \rangle$. A +-– labelling M of F is a total function $M : \mathcal{A} \to 2^{\{+,-\}}$ satisfying the following rules $\forall a \in \mathcal{A}$:*

- $- \in M(a) \implies \exists b \in \mathcal{A} \mid (b, a) \in \mathcal{R} \land + \in M(b)$
- $+ \in M(a) \implies \forall b \in \mathcal{A} \mid (b, a) \in \mathcal{R}. - \in M(b) \land$
  $\forall c \in \mathcal{A} \mid (a, c) \in \mathcal{R}. - \in M(c)$

Arguments only detaining a $+$ (or $-$) label are accepted (rejected, respectively), those with both labels are undecided, and those with no label are just ignored. In Figure 4, grey arguments have an empty label $\emptyset$, while yellow ones have both $+$ and $-$.

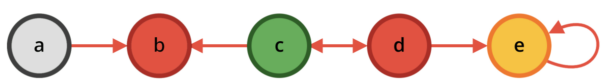

Fig. 4. Example of +-– labelling.

Comparing this labelling with the IN-OUT-UNDEC one, we have that, for any argument $a \in \mathcal{A}$, $M(a) = \{+\} \implies L(a) = \texttt{IN}$, $M(a) = \{-\} \implies L(a) = \texttt{OUT}$ and $M(a) = \{+, -\} \implies L(a) = \texttt{UNDEC}$. Since the implications only hold for one direction, there is no correspondence between +-– labelling and classical semantics: for instance, while arguments labelled $\{+\}$ by M will always be labelled IN by L, the vice versa is not true, meaning an argument a can exist for which $L(a) = \texttt{IN}$ and $M(a) = \{\}$. Note that, in both Definition 5 and 7, if an accepted argument a attacks another argument b, then b must be rejected.

To give the possibility of ignoring particular arguments without losing the link with extension-based semantics, the authors of [18] use, instead, an OFF label for arguments that must not be evaluated when computing acceptance (we show in Figure 5 an example of labelling where b is an OFF argument). Only the grounded labelling is taken into account, starting from a grounded IN-OUT-UNDEC labelling L_{gde}, as we report in the following.

Definition 8 (Grounded IN-OUT-UNDEC-OFF labelling for AFs [18, Definition 2.16]). *Let $F = \langle \mathcal{A}, \mathcal{R} \rangle$ and $G = \langle \mathcal{A}', \mathcal{R}' \rangle$ with $\mathcal{A}' \subseteq \mathcal{A}$ and $\mathcal{R}' \subseteq \mathcal{R}$. A grounded IN-OUT-UNDEC-OFF labelling N_{gde} with respect to $\mathcal{A}'$ is a total function $N_{gde} : \mathcal{A} \rightarrow \{$IN, OUT, UNDEC, OFF$\}$ such that:*

- $\forall a \in \mathcal{A} \setminus \mathcal{A}'. N_{gde}(a) = \text{OFF}$
- $\forall a \in \mathcal{A}'. N_{gde}(a) = L_{gde}(a)$

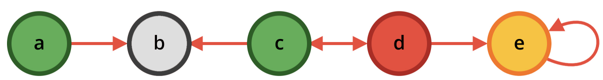

Fig. 5. Example of grounded IN-OUT-UNDEC-OFF labelling.

A different split for UNDEC arguments is proposed in [2], following the intuition that any argument allowing some positive interpretation should be accepted, regardless of possible negative interpretations.

Definition 9 (IN-OUT-BOTH-NONE labelling for AFs [2, Definition 9]). *Let $F = \langle \mathcal{A}, \mathcal{R} \rangle$ and $E \subseteq \mathcal{A}$. An IN-OUT-BOTH-NONE labelling with respect to E is a total function $O_E : \mathcal{A} \rightarrow \{$IN, OUT, BOTH, NONE$\}$ satisfying $\forall a \in \mathcal{A}$:*

- $O_E(a) = \text{IN} \iff a \in E \land a \notin E^+$
- $O_E(a) = \text{OUT} \iff a \notin E \land a \in E^+$
- $O_E(a) = \text{BOTH} \iff a \in E \land a \in E^+$
- $O_E(a) = \text{NONE} \iff a \notin E \land a \notin E^+$

Moreover, we say that O_E is BOTH-free when $E \cap E^+ = \emptyset$.

Argument a in Figure 6 is labelled NONE, while e is labelled BOTH. Definitions for p-admissible and p-complete labellings are also given in [2], which are based on Definition 9 and are used to identify paraconsistent semantics. Paraconsistent labellings describe the role of arguments in a framework, rather than justifying their acceptability. For instance a p-admissible labelling[3] allows OUT arguments to be attacked by IN and/or BOTH. According to [2, Propositions 30 and 33], a BOTH-free IN-OUT-BOTH-NONE labelling O_E is a p-admissible (p-complete, respectively) labelling for F if and only if E is an admissible (complete, respectively) extension of F.

[3] The p- stands for paraconsistent.

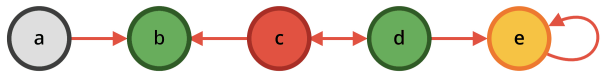

Fig. 6. Example of IN-OUT-BOTH-NONE labelling with respect to $E = \{b, d, e\}$.

3 A Unifying Four-State Labelling Semantics

The labellings discussed in the previous section have both pros and cons that vary according to the point of view. Four-state labellings [2, 17, 18] are more informative than the IN-OUT-UNDEC one [11, 12] (that does not include an $\emptyset$/OFF/NONE label), but in general there is no direct connection between +-−, IN-OUT-UNDEC-OFF, IN-OUT-BOTH-NONE labellings and extension-based semantics. For instance, an IN-OUT-UNDEC-OFF labelling identifies the grounded extension, but does not address the other semantics, while a p-admissible (p-complete) IN-OUT-BOTH-NONE labelling corresponds to an admissible (complete) extension only if it is BOTH-free.

On the other hand, an IN-OUT-UNDEC labelling can always be mapped into a set of accepted arguments, but it does not allow to leave unlabelled arguments that we do not want to consider in computing acceptability, and forces all arguments that are neither IN nor OUT to be labelled UNDEC. Consequently, when considering IN-OUT-UNDEC labellings to inspect AFs, the information brought by the UNDEC label can be misleading. Also, since any IN-OUT-UNDEC labelling corresponds to a complete extension, it cannot identify conflict-free sets. To overcome these inconveniences, we propose a four-state labelling which considers not only complete, but also admissible and conflict-free sets of arguments, and that provides a unifying representation for the various approaches proposed in the literature.

Definition 10 (Four-state labelling-based semantics). *Let $\mathcal{U}$ be a universe of arguments and $F = \langle \mathcal{A}, \mathcal{R} \rangle$ with $\mathcal{A} \subseteq \mathcal{U}$. A four-state labelling L^F of F is a partial[4] function $L^F : \mathcal{U} \rightharpoonup \{IN, OUT, DK, DC\}$ such that $\forall a \in \mathcal{U} \setminus \mathcal{A}. L(a) = \uparrow$. Superscript F will be omitted when clear from the context. We say that:*

- *L is a conflict-free labelling when*
  - $L(a) = IN \implies \forall b \in a^-. L(b) \neq IN$ *and*
  - $L(a) = OUT \implies \exists b \in a^- \mid L(b) = IN$
- *L is an admissible labelling when*
  - $L(a) = IN \implies \forall b \in a^-. L(b) = OUT$ *and*
  - $L(a) = OUT \iff \exists b \in a^- \mid L(b) = IN$
- *L is a complete labelling when*
  - $L(a) = IN \iff \forall b \in a^-. L(b) \in \{OUT, DC\}$ *and*
  - $L(a) = OUT \iff \exists b \in a^- \mid L(b) = IN$

[4] The labelling function is not defined for not adopted arguments.

- *L is a stable labelling when*
  - *L is a complete labelling and*
  - $\mathcal{A}\downarrow_{DK} = \emptyset$
- *L is a preferred labelling when*
  - *L is an admissible labelling and*
  - $\mathcal{A}\downarrow_{IN}$ *is maximal among all the admissible labellings*
- *L is a grounded labelling when*
  - *L is a complete labelling and*
  - $\mathcal{A}\downarrow_{IN}$ *is minimal among all the complete labellings*

We denote with L_σ a labelling L satisfying the above conditions for a given semantics σ, with $\mathcal{L}$ the set of all possible labellings, and with $\mathcal{L}_\sigma$ the set of all possible labellings satisfying conditions for σ. For any $A \subseteq \mathcal{A}$, $R \subseteq \mathcal{R}$ and $l \subseteq \{\texttt{IN}, \texttt{OUT}, \texttt{DK}, \texttt{DC}\}$, we use $A\downarrow_l = \{a \in A \mid L(a) \in l\}$ and $R\downarrow_l = \{(a, b) \in R \mid L(a) \in l \wedge L(b) \in l\}$ to restrict to arguments and relations only involving certain (sets of) labels.

The labelling of an AF contains information about the acceptability of the arguments in the framework (according to the various Dung's semantics) and can be used by intelligent agents to represent the state of their beliefs. Each different label can be traced to a particular meaning. For instance, DC stands for "don't care" [17] and identifies arguments that are not interesting for the agents. Arguments in $\mathcal{U} \setminus \mathcal{A}$, that are only part of the universe, but not of the AF, are not labelled. Accepted and rejected arguments (labelled as IN and OUT, respectively), allow agents to discern true beliefs from the false ones. At last, DK could be both accepted and rejected, meaning that agents cannot decide about the acceptability of a belief ("don't know", indeed).

Since arguments in $\mathcal{U} \setminus \mathcal{A}$ do not constitute an actual part of the AF, they are not labelled, as we consider them not adopted. We, instead, label DC any argument in $\mathcal{A}$ we don't care about. Notice that arguments we do not adopt and arguments we do not care about are very different: not adopted arguments are involved in no attack, as they do not even belong to the considered AF; don't care arguments, instead, can be involved in attacks and we also consider them for deciding the label of the attacked arguments. In particular, concerning the complete four-state labelling, an argument a that is only attacked by ignored arguments (labelled DC) is accepted and thus labelled IN, as happens in [18], providing an optimistic interpretation of the DC label[5].

We now show the correspondence between four-state labellings satisfying restrictions given in the definition above and the extensions of a certain semantics.

Theorem 1. *A four-state labelling* L^F *of* $F = \langle \mathcal{A}, \mathcal{R} \rangle$ *is a conflict-free labelling if and only if* $\mathcal{A}\downarrow_{IN}$ *is a conflict-free extension of* F. *Moreover,* L^F *is an admissible (respectively complete, stable, preferred, grounded) labelling if and only if* $\mathcal{A}\downarrow_{IN}$ *is an admissible (respectively complete, stable, preferred, grounded) extension of* $F' = \langle \mathcal{A}\downarrow_{IN,OUT,DK}, \mathcal{R}\downarrow_{IN,OUT,DK} \rangle$.

[5] A pessimistic interpretation labelling as OUT arguments attacked by DC could also be introduced.

The four-state labelling can also be mapped into the labellings presented in the previous section, hence providing a unifying representation for argument states.

Theorem 2. *A four-state labelling L is an* IN-OUT-UNDEC *labelling (see Definition 5) if and only if L is complete,* $\mathcal{A}\downarrow_{DC} = \emptyset$ *and* $\mathcal{A} = \mathcal{U}$*, with the mapping among labels* $DK \equiv UNDEC$*.*

Proof of Theorem 2 directly follows from Definitions 5 and 10. Moreover, by using the conditions of Table 1, four-state labellings can be traced to stable (respectively preferred, grounded) IN-OUT-UNDEC labellings.

Corollary 1. *A four-state labelling L is a stable (preferred or grounded, respectively)* IN-OUT-UNDEC *labelling if and only if L is stable (preferred or grounded, respectively),* $\mathcal{A}\downarrow_{DC} = \emptyset$*,* $\mathcal{A} = \mathcal{U}$ *and* $\mathcal{A}\downarrow_{DK} = \emptyset$ *(*$\mathcal{A}\downarrow_{IN}$ *is maximal or* $\mathcal{A}\downarrow_{IN}$ *is minimal, respectively), with the mapping among labels* $DK \equiv UNDEC$*.*

Corollary 1 can be proved considering that L is an IN-OUT-UNDEC labelling (according to Theorem 2) satisfying the restrictions of Table 1 for stable (preferred or grounded, respectively) labellings. Finally, we compare four-state and admissible IN-OUT-UNDEC labellings.

Theorem 3. *A four-state labelling L is an admissible* IN-OUT-UNDEC *labelling (see Definition 6) if and only if L is admissible,* $\mathcal{A}\downarrow_{DC} = \emptyset$ *and* $\mathcal{A} = \mathcal{U}$*, with the mapping among labels* $DK \equiv UNDEC$*.*

Proof of Theorem 3 directly follows from Definitions 6 and 10. When considering +-− labelling, we have to keep in mind that none, one or two labels can be assigned to each argument.

Theorem 4. *A four-state labelling L is a +-− labelling (see Definition 7) if and only if L is admissible and* $\mathcal{A} = \mathcal{U}$*, with the mapping among labels* $IN \equiv +$*,* $OUT \equiv -$*,* $DK \equiv \{+, -\}$ *and* $DC \equiv \emptyset$*.*

IN-OUT-UNDEC-OFF labellings, then, allow arguments to be excluded from the computation of the acceptability, similarly to how not adopted arguments can be ignored in a four-state labelling.

Theorem 5. *A four-state labelling L is a grounded* IN-OUT-UNDEC-OFF *labelling with respect to* $\mathcal{A}'$ *(see Definition 8) if and only if L is grounded,* $\mathcal{A}\downarrow_{DC} = \emptyset$*, and* $\mathcal{A}' = \mathcal{U} \setminus \mathcal{A}$*, with the mapping among labels* $\uparrow \equiv OFF$ *and* $DK \equiv UNDEC$*.*

Theorems 4 and 5 can be proved directly from Definitions 7, 8 and 10.

Notice that, in general, a direct mapping from IN-OUT-BOTH-NONE labelling to our four-state labelling does not exist. Assume NONE $\equiv$ DC and BOTH $\equiv$ DK, and consider the p-admissible labelling of Figure 7 (left) with respect to $E = \{a, b, d\}$. The OUT argument c is only attacked by argument b (labelled BOTH), and the represented labelling is not an admissible four-state labelling since, according to Definition 10, OUT arguments must be attacked by at least one IN. A four-state labelling, then, is not guaranteed to be an admissible IN-OUT-BOTH-NONE labelling. On the other side, the admissible four-state labelling of Figure 7 (right) is not p-admissible, since according to [2, Definition 11] NONE arguments cannot have an attacker labelled BOTH.

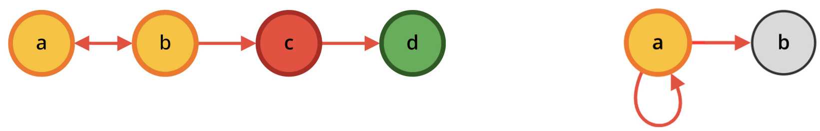

Fig. 7. Examples of a p-admissible labelling (on the left) and admissible four-state labelling (on the right).

Consider now the labelling of Figure 8 (left). It is a complete four-state labelling, but not a p-complete labelling, since the IN argument b is not attacked by any OUT. Finally, Figure 8 (right) shows a p-complete labelling that is not a complete four-state labelling. Indeed, the OUT argument e is not attacked by any IN.

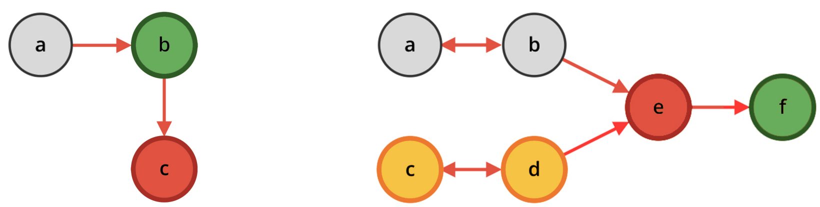

Fig. 8. Examples of a complete four-state labelling (on the left) and a p-complete labelling (on the right).

4 Revision of Labels in AFs

The four-state labelling introduced in the previous section share similarities with the AGM framework [1], and in particular with the states that can be associated to information in a knowledge base. The mapping between argument labels and belief states allow for using argumentation semantics as a reasoning engine. These states can be sorted according to the amount of information they hold by using a partial order relation $\prec_{kb}$. Starting from $\uparrow$, representing the absence from the knowledge base, and arriving to DK, which is the label with the greatest amount of information, we have $\uparrow \prec_{kb}$ DC $\prec_{kb}$ IN/OUT $\prec_{kb}$ DK.

The AGM framework [1] provides an approach to the problem of revising knowledge basis by using theories (deductively closed sets of formulae) to represent the beliefs of the agents. A formula α in a given theory can have different statuses for an agent, according to its knowledge base K: if the agent can deduce α from its beliefs, we say that α is *accepted*; if the agent can deduce the negation of α, then we say that α is *rejected*; otherwise, the agent cannot deduce anything and α is *undetermined*. The correspondence between accepted/rejected beliefs and IN/OUT arguments in a labelling is straightforward. Since the undetermined status represents the absence of a piece of information (nothing can be deduced in favour of either accepting or rejecting a belief) it can be mapped into both the label DC and the absence of the label itself ($L(a) = \uparrow$ when $a \in \mathcal{U} \setminus \mathcal{A}$) . Finally, the DK label is assigned to arguments that are both IN and OUT, boiling down to the notion of inconsistency in AGM. Arguments for which the labelling is not defined (i.e., those in $\mathcal{U} \setminus \mathcal{A}$), play a fundamental role in identifying new arguments that agents can bring to the debate to defend (or strengthen) their position. The status of a belief can be changed through some operations (namely expansion, contraction and revision) on the knowledge base, as depicted in Figure 9.

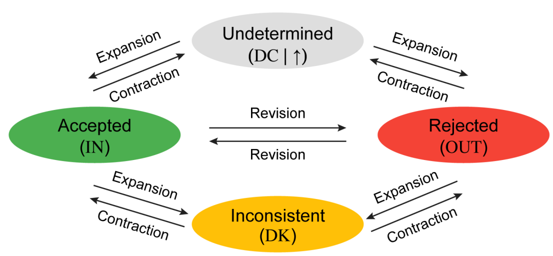

Fig. 9. Transitions between AGM beliefs states.

An expansion basically brings new pieces of information to the base, allowing for undetermined belief to become either accepted or refused. A contraction, on the contrary, reduces the information an agent can rely on in making its deduction. A revision, then, makes acceptable belief refused and vice-versa. The AGM framework also defines three sets of rationality postulates (one for each operation) that any good operator should satisfy.

AGM operators provide building blocks for realizing complex interaction processes between agents. As for knowledge basis in belief revision, AFs can undergo changes that modify the structure of the framework itself, either integrating new information (and so increasing the arguments and the attacks in the AF) or discarding previously available knowledge. Agents using AFs as the mean for exchanging and inferring information have to rely on operations able to modify such AFs. Besides considering the mere structural changes, also modifications on the semantics level need to be addressed by the operations executed by the agents. In the following, we define an *argument expansion* operator for AFs, that complies with classical operators of AGM. Notice that changes to the knowledge base we are interested in modelling are restricted to a single argument at a time, miming the typical argument interaction in dynamic AFs. In this paper we only provide definition and postulates for the expansion operator.

Definition 11 (Single labelling argument expansion). *Let $F = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF on the universe $\mathcal{U}$, σ a semantics, L_σ^F a given labelling function, and $a \in \mathcal{U}$ an argument. A single labelling argument expansion with respect to a labelling L_σ^F consists of a function $\oplus_{L_\sigma^F} : AF \times \mathcal{U} \rightarrow AF$ that computes a new framework $F' = F \oplus_{L_\sigma^F} a$ such that $L_\sigma^{F'}(a) \succcurlyeq_{kb} L_\sigma^F(a)$.*

4.1 AGM-Style Postulates

In devising our expansion operator, we reinterpret the AGM expansion operator [1]. We can prove that our operator (Definition 11) satisfies the AGM expansion postulates [1]. Given two AFs F and G with arguments on $\mathcal{U}$, a semantics σ, a labelling function L_σ, and an argument $a \in \mathcal{U}$, we say that $G \subseteq_{L_\sigma}^a F$ when $L_\sigma^G(a) \preccurlyeq_{kb} L_\sigma^F(a)$. Postulates for single labelling argument expansion can be formulated as follows:

1. $F \oplus_{L_\sigma^F} a$ is an AF
2. given $F' = F \oplus_{L_\sigma^F} a$, $L_\sigma^{F'}(a) \succ_{kb} \uparrow$
3. given $F' = F \oplus_{L_\sigma^F} a$, $L_\sigma^{F'}(a) \succcurlyeq_{kb} L_\sigma^F(a)$
4. if $L_\sigma^F(a) = \texttt{DK}$, then $F \oplus_{L_\sigma^F} a = F$
5. if $G \subseteq_{L_\sigma}^a F$, then $G \oplus_{L_\sigma^G} a \subseteq_{L_\sigma}^a F \oplus_{L_\sigma^F} a$
6. given $F' = F \oplus_{L_\sigma^F} a$, if $L_\sigma^F(a) = \uparrow$, then $L_\sigma^{F'}(a) = \texttt{IN}/\texttt{OUT}$, and if $L_\sigma^F(a) = \texttt{IN}/\texttt{OUT}$, then $L_\sigma^{F'}(a) = \texttt{DK}$

5 Conclusion and Future Work

We defined a four-state labelling semantics for AFs that allows for establishing acceptability of arguments on a finer grain with respect to other approaches presented in the literature. We use four labels: IN, OUT, DK and DC, the last one denoting arguments we want to ignore. We showed the connection between our labelling and the AGM framework, introducing operators for expansion, contraction and revision of arguments states. AGM operators have already been studied from the point of view of their implementation in work as [6, 15], especially with regard to enforcement. However, differently from the previous literature, where extensions are considered, we take into account single arguments.

In our setting, arguments only attacked by DC arguments are always labelled IN. As future work, we want to consider a pessimistic interpretation for ignored arguments: since a DC-labelled argument a could be (re)considered into the AF, thus gaining an IN, OUT or DK label, arguments only attacked by a could be labelled OUT in turn. We also plan to extend the four-state labelling to weighted AFs [9].

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. J. Symb. Log. **50**(2), 510–530 (1985). https://doi.org/10.2307/2274239
2. Arieli, O.: On the acceptance of loops in argumentation frameworks. J. Log. Comput. **26**(4), 1203–1234 (2016). https://doi.org/10.1093/logcom/exu009
3. Baroni, P., Caminada, M., Giacomin, M.: An introduction to argumentation semantics. Knowl. Eng Rev. **26**(4), 365–410 (2011). https://doi.org/10.1017/S0269888911000166
4. Baroni, P., Caminada, M., Giacomin, M.: Abstract argumentation frameworks and their semantics. Handbook of formal argumentation **1**, 157–234 (2018)
5. Baroni, P., Giacomin, M., Liao, B.S.: I don't care, I don't know ... I know too much! on incompleteness and undecidedness in abstract argumentation. In: Eiter, T., Strass, H., Truszczynski, M., Woltran, S. (eds.) Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation - Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday. Lecture Notes in Computer Science, vol. 9060, pp. 265–280. Springer (2015). https://doi.org/10.1007/978-3-319-14726-0_18
6. Baumann, R., Brewka, G.: Expanding argumentation frameworks: Enforcing and monotonicity results. In: Baroni, P., Cerutti, F., Giacomin, M., Simari, G.R. (eds.) Computational Models of Argument: Proceedings of COMMA 2010, Desenzano del Garda, Italy, September 8-10, 2010. Frontiers in Artificial Intelligence and Applications, vol. 216, pp. 75–86. IOS Press (2010). https://doi.org/10.3233/978-1-60750-619-5-75
7. Bistarelli, S., Taticchi, C.: A concurrent language for argumentation. In: Fazzinga, B., Furfaro, F., Parisi, F. (eds.) Proceedings of the Workshop on Advances In Argumentation In Artificial Intelligence 2020 co-located with the 19th International Conference of the Italian Association for Artificial Intelligence (AIxIA 2020), Online, November 25-26, 2020. CEUR Workshop Proceedings, vol. 2777, pp. 75–89. CEUR-WS.org (2020), http://ceur-ws.org/Vol-2777/paper67.pdf
8. Bistarelli, S., Taticchi, C.: A concurrent language for argumentation: Preliminary notes. In: de Boer, F.S., Mauro, J. (eds.) Recent Developments in the Design and Implementation of Programming Languages, Gabbrielli's Festschrift, November 27, 2020, Bologna, Italy. OASIcs, vol. 86, pp. 9:1–9:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). https://doi.org/10.4230/OASIcs.Gabbrielli.9
9. Bistarelli, S., Taticchi, C.: A labelling semantics for weighted argumentation frameworks. In: Calimeri, F., Perri, S., Zumpano, E. (eds.) Proceedings of the 35th Italian Conference on Computational Logic - CILC 2020, Rende, Italy, October 13-15, 2020. CEUR Workshop Proceedings, vol. 2710, pp. 263–277. CEUR-WS.org (2020), http://ceur-ws.org/Vol-2710/paper17.pdf
10. Budán, M.C., Lucero, M.J.G., Chesñevar, C.I., Simari, G.R.: Modeling time and valuation in structured argumentation frameworks. Inf. Sci. **290**, 22–44 (2015). https://doi.org/10.1016/j.ins.2014.07.056
11. Caminada, M.: On the Issue of Reinstatement in Argumentation. In: Logics in Artificial Intelligence, 10th European Conference, JELIA 2006, Liverpool, UK, September 13-15, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4160, pp. 111–123. Springer (2006). https://doi.org/10.1007/11853886_11
12. Caminada, M.: Strong Admissibility Revisited. In: Computational Models of Argument - Proceedings of COMMA 2014, Atholl Palace Hotel, Scottish Highlands, UK,

September 9-12, 2014. Frontiers in Artificial Intelligence and Applications, vol. 266, pp. 197–208. IOS Press (2014). https://doi.org/10.3233/978-1-61499-436-7-197
13. Caminada, M.W.A., Gabbay, D.M.: A Logical Account of Formal Argumentation. Studia Logica **93**(2-3), 109–145 (2009). https://doi.org/10.1007/s11225-009-9218-x
14. Cobo, M.L., Martínez, D.C., Simari, G.R.: Acceptability in timed frameworks with intermittent arguments. In: Iliadis, L.S., Maglogiannis, I., Papadopoulos, H. (eds.) Artificial Intelligence Applications and Innovations - 12th INNS EANN-SIG International Conference, EANN 2011 and 7th IFIP WG 12.5 International Conference, AIAI 2011, Corfu, Greece, September 15-18, 2011, Proceedings, Part II. IFIP Advances in Information and Communication Technology, vol. 364, pp. 202–211. Springer (2011). https://doi.org/10.1007/978-3-642-23960-1_25
15. Coste-Marquis, S., Konieczny, S., Mailly, J., Marquis, P.: Extension enforcement in abstract argumentation as an optimization problem. In: Yang, Q., Wooldridge, M.J. (eds.) Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015. pp. 2876–2882. AAAI Press (2015), http://ijcai.org/Abstract/15/407
16. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-Person games. Artif. Intell. **77**(2), 321–357 (Sep 1995). https://doi.org/10.1016/0004-3702(94)00041-X
17. Jakobovits, H., Vermeir, D.: Robust Semantics for Argumentation Frameworks. J Log Comput **9**(2), 215–261 (1999). https://doi.org/10.1093/logcom/9.2.215
18. Riveret, R., Oren, N., Sartor, G.: A probabilistic deontic argumentation framework. Int. J. Approx. Reason. **126**, 249–271 (2020). https://doi.org/10.1016/j.ijar.2020.08.012