

Enhancing Statement Evaluation in Argumentation via Multi-labelling Systems

Pietro Baroni

*DII, University of Brescia
Brescia, Italy*

PIETRO.BARONI@UNIBS.IT

Régis Riveret

*Data61, CSIRO
Brisbane, Australia*

REGIS.RIVERET@DATA61.CSIRO.AU

Abstract

In computational models of argumentation, the justification of statements has drawn less attention than the construction and justification of arguments. As a consequence, significant losses of sensitivity and expressiveness in the treatment of statement statuses can be incurred by otherwise appealing formalisms. In order to reappraise statement statuses and, more generally, to support a uniform modelling of different phases of the argumentation process we introduce multi-labelling systems, a generic formalism devoted to represent reasoning processes consisting of a sequence of labelling stages. In this context, two families of multi-labelling systems, called *argument-focused* and *statement-focused* approach, are identified and compared. Then they are shown to be able to encompass several prominent literature proposals as special cases, thereby enabling a systematic comparison evidencing their merits and limits. Further, we show that the proposed model supports tunability of statement justification by specifying a few alternative statement justification labellings, and we illustrate how they can be seamlessly integrated into different formalisms.

1. Introduction

A full formalism for argument-based reasoning is required to cover various aspects of the argumentation process like argument construction, argument assessment and statement assessment. It can be observed, however, that while generally argument construction and assessment have been investigated in great detail in the literature, statement statuses have often been treated as a byproduct of the previous phases and their assessment has drawn less attention.

As a consequence, losses of expressiveness and sensitivity can be incurred in the treatment of statement statuses by otherwise appealing formalisms, possibly leading to puzzling differences among formalisms and their outcomes, even in very simple common sense cases. This can occur in particular in some basic situations, such as two mutually attacking arguments, an argument attacking another one without being counterattacked, or the absence of any argument about a given statement. An instance of these three situations is provided in Example 1.1 below.

Example 1.1 (Adapted from Baroni, Governatori, Lam, & Riveret, 2016a; Baroni, Governatori, & Riveret, 2016b). *Suppose that Dr. Smith says to you: ‘Given your clinical data I conclude you are affected by disease D1’. Suppose then that another equally competent physician Dr. Jones says to you: ‘Given your clinical data I conclude you are not affected by disease D1’. Your view on the justification of the statements $s_1 = ‘I am affected by disease D1’$ and $\neg s_1 = ‘I am not affected by disease D1’$ may become quite uncertain.*

In a different situation, at home, you use an off-the-shelf test kit suggesting you have caught disease D2. You then undertake a serious and reliable clinical test, which excludes disease D2. Would you consider the same status for the statement $s_2 = 'I \text{ am affected by disease D2}'$ and the statement s_1 ?

In addition to statements which are supported by some evidence (i.e. those which can be regarded intuitively as the conclusion of some arguments, like those exemplified above), one may consider also statements which are not supported by any evidence given the currently available information (i.e. those which are not the conclusion of any argument). Consider for instance the statement $s_3 = 'I \text{ am affected by D3}'$, where D3 is a poorly studied and initially asymptomatic disease you only know by name. It should be possible to ascribe a justification status (intuitively representing a situation of full ignorance) also to such a statement and it seems reasonable and useful that such justification status is distinct with respect to those of s_1 and s_2 . Actually, such a distinction may be decisive.

Distinctions between statement justification statuses can be certainly achieved by some argumentation formalisms, but surprisingly enough, there are well-known argumentation formalisms where not all the desirable distinctions are captured (depending on the argumentation semantics adopted). For instance, there are formalisms equating the justification status of s_3 with the one of s_2 , or with that of s_1 and $\neg s_1$. There are also situations (in particular under specific choices of the applied argumentation semantics) where the justification status of s_2 is equated with those of s_1 and $\neg s_1$. As to our knowledge, these differences in the treatment of statement justification statuses have received limited attention in the literature, and a systematic formal framework to analyse and compare different formalisms in this respect, also supporting the introduction of possible fixes where needed, is lacking. \square

To fill the gap mentioned above and to support the reappraisal of the evaluation of statement statuses, we propose in this paper the general model of multi-labelling systems¹, i.e. systems based on multiple stages of labellings, and we illustrate its ability to support both the analysis and further development of different argument-based formalisms, in particular by establishing a set of general formal properties useful for their characterisation and comparison.

1.1 Contribution.

This paper first introduces the novel model of multi-labelling systems (MLSs) and defines some fundamental properties therein. To demonstrate the applicability of the proposed model, we identify two main classes of MLSs for argument-based reasoning, namely the argument-focused approach and the statement-focused approach. We compare these two approaches, and then show that several existing formalisms can be regarded as instances of MLSs under one (or possibly both) of the two approaches. The investigation leads then us to propose diverse statement labellings that can be ‘plugged’ into existing argument-based approaches, thus allowing a variety of tunable attitudes towards statement justification, all ensuring a uniform treatment of the Example 1.1 across different formalisms.

The main contribution lies in a general framework, lacking in the previous literature, encompassing phases of argument and statement labellings in a unitary context. The formalism supports

1. The ideas underlying multi-labelling systems and their application to argumentation have been preliminarily presented in (Baroni et al., 2016a, 2016b). The model presented in this paper has been redefined from scratch to achieve a full generalisation of these ideas and a systematic applicability to a variety of argumentation formalisms.

the identification of the two main classes of MLSs for argumentation mentioned above, of which we provide, as a further contribution, a formal comparison of expressiveness. The reconciliation of statement justification of different formalisms at various levels of details through pluggable labellings demonstrates the applicability of the framework and its ability to support the systematic study of novel solutions. Given their illustrative purpose, each of the proposed labellings is relatively simple and in principle could have been conceived independently of the formal framework we propose. However the fact that, simple as they are, these labellings have not been previously introduced in the literature, while their analysis is straightforward in our framework shows its utility in the systematic design and comparison of diverse statement justification approaches.

1.2 Outline.

In Section 2, our investigation is further motivated. We present in Section 3 the model of MLSs for argumentation and some general properties useful to characterize its instances. We analyse in Section 4 the application of MLSs to argumentation by characterising an argument-focused approach and a statement-focused approach. In Section 5 we compare the expressiveness of the two approaches, and in Section 6 we specialise some properties of MLSs to them. To demonstrate the applicability of the proposed model, we show in Section 7 that several literature proposals can be reconstructed as instances of it, and that this reconstruction supports their analysis and comparison. Sections 8, 9, and 10 illustrate how argumentation formalisms can be tuned to support a variety of statement justification labellings by exploiting the model of MLSs. Section 11 discusses some relationships with existing literature and Section 12 concludes.

2. Motivations

Motivations to reappraise statement assessment with multi-labelling systems are partitioned into motivations for the reappraisal and motivations to do it with multi-labelling systems.

2.1 Motivations to Reappraise Statement Assessment

The reappraisal of statement assessment is motivated by the observation that, in studies of formal argumentation, there is an unbalanced treatment between the construction of arguments and the assessment of their statuses on one hand, and the assessment of statement statuses on the other hand.

At a general level, various models of structured argumentation and argument-related logic frameworks for defeasible reasoning (see e.g. Modgil & Prakken, 2014; Toni, 2014; García & Simari, 2014) can be cast into a generic process consisting of three main phases:

- construction of arguments;
- assessment of argument statuses;
- assessment of statement statuses.

A major line of research on argument assessment is based on Dung abstract argumentation frameworks and their semantics (Dung, 1995) which are employed by several formalisms, like *ASPIC*⁺ (Modgil & Prakken, 2014) and *ABA* (Toni, 2014), while other formalisms like *DeLP* (García & Simari, 2014) do not rely on Dung's system and include their own methods of evaluation. Given this diversity, different assessment criteria corresponding to different attitudes may be adopted,

even in the context of the same formalism. In this sense, argument assessment is generally seen as a pluggable and tunable component, and this has given rise to a large corpus of studies on different argumentation semantics and on their principle-based evaluation and comparison. In contrast, statement assessment has mostly been treated as a direct byproduct of argument assessment. Typically formalisms include their own specific definition of statement assessment. A possible variety of approaches is not considered, it is not seen as a pluggable or tunable component, there is no systematic investigation of general properties and relevant comparisons. Hence, the reappraisal of statement assessment is motivated by this evident unbalance and by the goal of extending the rich expressiveness and adaptability of argumentation formalisms to this phase too.

2.2 Motivations to Multi-labelling Systems

To achieve the reappraisal of statement assessment, we have devised MLSs, whose main features result from three main motivations.

A first motivation originates from the desire to reflect common stages in argumentation models which can be found in the literature. Statement assessment can be reappraised in various ways, including those not featuring a direct correspondence with stages of natural argumentation. For example, Brewka and Woltran (2010) proposed a powerful formalism, called Abstract Dialectical Frameworks, which can be used to capture arbitrary relationships in a network of statements through a variety of statement acceptance conditions. However, such a framework and similar proposals do not directly address the articulation of the process into common stages such as those pertaining to argument construction, argument assessment and statement assessment. MLSs aim at reappraising statement assessment by reflecting the common stages found in structured argumentation models.

Accordingly, as a second motivation, the proposed model aims at capturing formalisms found in the literature as instances of MLSs. In this sense, the formalism of MLSs can be seen as a sort of meta-model² of argumentation, supporting a systematic analysis and comparison of different formalisms.

A third motivation of the multi-labelling model comes from the intention to support the design of ‘argument-based software systems’ based on well-established software engineering principles. As MLSs support a rigorous and formal separation of concerns corresponding to different stages of the argumentation process, one may decompose an argument-based software system into well-defined independent modules, for example by developing a module for each labelling stage. The modules can be then tested in isolation before their integration, leading to a better maintainability. As each module can be tuned, for instance to implement a different type of labelling, this modular design favours reusability of the different modules. Any modules may be plugged into other systems thanks to abstract interfaces of the different argumentation stages, towards higher interoperability and thus (re)usability. By doing so, a multi-labelling software system could also be reused to reflect different argumentation systems without major changes to the conceptual integrity of the underlying software architecture.

To sum up, the reappraisal of statement assessment is motivated by the evidenced unbalance of treatment between argument construction and assessment on the one hand, and statement assessment on the other hand. To achieve this reappraisal, we propose the general model of MLSs, which supports an explicit articulation of ordinary stages found in argumentation formalisms, and which is

2. This is different from the so called meta-argumentation models where, for example, one may advance arguments about arguments.

tailored to account for different argument-based approaches, while keeping in mind well-established principles of software engineering.

3. Multi-labelling Systems

MLSs provide a generic model for any form of reasoning or evaluation process which can be conceived as a sequence of stages, where the result of each stage consists in one or more labellings of a set of objects. The set of labelled objects may vary through the stages, the general idea being that, starting from an initial stage, the outcome of each stage is derived from the outcome of the previous stage on the basis of some relationships between the sets of labelled objects considered in the two stages. Formally, multi-labelling systems can be understood as finite compositions of labelling generators yielding sequences of ‘labellings collections’. Such compositions are defined in this section at an abstract general level, and will be applied in later sections to the case of argumentation systems.

3.1 Basic Concepts

For the sake of generality, our definition of MLSs resorts to multisets rather than ordinary sets, i.e. collections admitting repetitions of their elements or, in other words, where elements have a multiplicity.

Definition 3.1 (Multiset). *A multiset is a pair $\langle S, m \rangle$ where S is a set and $m : S \rightarrow \mathbb{N}^{\geq 1}$ is a function where for every $s \in S$, $m(s)$ is the multiplicity of s .*

Notation 3.1. *An element of a multiset may be denoted as a pair (s, k) where $s \in S$ and $k \in \mathbb{N}^{\geq 1}$ is its multiplicity. Given a multiset $\widehat{S} = \langle S, m \rangle$, we may write $s \in^m \widehat{S}$ iff $s \in S$.*

Definition 3.2 (Multisubset). *Given a set S , a multiset $\langle S', m \rangle$ is a multisubset of S iff $S' \subseteq S$.*

Notation 3.2. *The set of all the multisubsets of a set S is denoted as $\Omega^*(S)$.*

Clearly, ordinary sets are multisets where the multiplicity of every element is 1. Concerning ordinary sets, we use the following notation for power sets.

Notation 3.3. *Given a set S , the powerset of S is denoted $\text{pow}(S)$, i.e. $\text{pow}(S) = \{S' \mid S' \subseteq S\}$.*

Multi-labelling systems are based on the notion of labelling.

Definition 3.3 (Labelling). *Given a set of labels Λ and a set S , a Λ -labelling L of S is a possibly partial³ function $L : S \rightarrow \Lambda$.*

Notation 3.4. *The set of all possible Λ -labellings of a set S is denoted $\mathfrak{L}(S, \Lambda)$, and the set of all possible multisubsets of Λ -labellings of a set S is denoted $\mathfrak{L}^*(S, \Lambda)$, i.e. $\mathfrak{L}^*(S, \Lambda) = \Omega^*(\mathfrak{L}(S, \Lambda))$. A generic unspecified labelling will be denoted by an italic symbol L , whereas an upright symbol L indicates a specific labelling (similar typographic arrangements hold for other constructs).*

3. All the functions in this paper are possibly partial, unless differently specified.

Example 3.1. *To illustrate the basic concepts of MLSs in general, we provide an example not regarding argumentation systems. Consider a set of three students $S_0 = \{\text{John, Kris, Laura}\}$ and assume that each member of an evaluation committee assigns a label of merit to each student from the set of labels $\Lambda_0 = \{A, B, C, D\}$. If the committee is composed of four members, then four (non necessarily distinct) Λ_0 -labellings of S_0 are produced to represent their evaluations, e.g.*

$$\begin{aligned} L_0^1 &= \{(\text{John}, C), (\text{Kris}, A), (\text{Laura}, B)\}, \\ L_0^2 &= \{(\text{John}, C), (\text{Kris}, B), (\text{Laura}, B)\}, \\ L_0^3 &= \{(\text{John}, C), (\text{Kris}, A), (\text{Laura}, A)\}, \\ L_0^4 &= L_0^3. \end{aligned}$$

Then in general the labellings produced by the committee members give rise to a multisubset $\widehat{\mathcal{L}}$ in $\mathfrak{L}^(S_0, \Lambda_0)$. In the example, we get $\widehat{\mathcal{L}} = \{(L_0^1, 1), (L_0^2, 1), (L_0^3, 2)\}$. \square*

A significant class of computational activities (including some related to argument-based reasoning) can be represented as a sequence of labellings produced at different stages starting from an initial one. Moving across the stages, the labellings produced at one stage are used as input to produce new labellings at the next stage, where the labels, their meaning and/or the labelled elements change. At an abstract level, the transition from a stage to another stage is captured by the notion of labelling generator.

Definition 3.4 (Labelling generator). *Given two sets S_1 and S_2 , a Λ_2 -labelling generator for S_2 based on the pair (S_1, Λ_1) is a function $gen : \mathfrak{L}^*(S_1, \Lambda_1) \rightarrow \mathfrak{L}^*(S_2, \Lambda_2)$.*

Example 3.2 (continues Example 3.1). *Suppose that the evaluation of students is subsequently exploited to rank their schools, such that, given the set of schools $S_1 = \{\text{School1, School2, School3}\}$, every school can be labelled with the set of labels $\Lambda_1 = \{\text{Excel}(\text{lent}), \text{Good}, \text{Med}(\text{iocre}), \text{Poor}\}$. The evaluation can be performed with a Λ_1 -labelling generator for S_1 based on the pair (S_0, Λ_0) . In turn, suppose that these schools are either public (Pub) or private (Priv), and that one wants to decide for each of these sectors whether it should be collectively subject to an investigation (Invest) or not (NoInvest), depending on the evaluations obtained by the schools in the sector. Letting $S_2 = \{\text{Pub, Priv}\}$ and $\Lambda_2 = \{\text{Invest, NoInvest}\}$, a Λ_2 -labelling generator for S_2 based on the pair (S_1, Λ_1) is needed to cover this subsequent evaluation stage. \square*

The elements introduced above provide the basis for defining the general notion of MLSs, namely a formal setting to capture the production of a cascading sequence of labellings starting from an initial one.

Definition 3.5 (Multi-labelling system). *A multi-labelling system is a finite composition of labelling generators $gen_m \circ gen_{m-1} \circ \dots \circ gen_1$, where for $1 \leq i \leq m$, $gen_i : \mathfrak{L}^*(S_{i-1}, \Lambda_{i-1}) \rightarrow \mathfrak{L}^*(S_i, \Lambda_i)$.*

Notation 3.5. *In order to make apparent the set of elements and labels in consideration, a MLS may be denoted as a sequence of stages (Ξ_0, \dots, Ξ_m) , where:*

- $\Xi_0 = \langle S_0, \Lambda_0 \rangle$ is a pair including a set S_0 (of elements to be labelled) and a set Λ_0 of labels;
- $\Xi_i = \langle S_i, \Lambda_i, gen_i \rangle$ (for $0 < i \leq m$) where S_i is a set, Λ_i is a set of labels and gen_i is Λ_i -labelling generator for S_i based on the pair (S_{i-1}, Λ_{i-1}) .

The idea is that, given a multiset of Λ_0 -labellings of S_0 as starting point, say $\widehat{\mathcal{L}}_0 \in \mathfrak{L}^*(S_0, \Lambda_0)$, defining a MLS corresponds to deriving a sequence of multisets of labellings $\widehat{\mathcal{L}}_i \in \mathfrak{L}^*(S_i, \Lambda_i)$ ($1 \leq i \leq m$) where $\widehat{\mathcal{L}}_i = gen_i(\widehat{\mathcal{L}}_{i-1})$.

3.2 Monolabelling and Polylabelling Generators

Having laid down a general notion of MLSs, we now introduce two specific families of labelling generators, namely polylabelling and monolabelling generators, which will be useful for our purpose of capturing argumentation formalisms. Both generators share the idea that labels are projected from a stage to the subsequent stage on the basis of some ‘influence’, and then the projected labels are synthesised into a new label. For this reason, the generators share the basic formal elements of influences, n -influences, label projections, and n -synthesizers; as defined next.

In general, each element of the set S_i to be labelled at stage i can be influenced by some elements belonging to the set S_{i-1} at the previous stage. Thus, for each element s of S_i , a set of influencing elements in S_{i-1} can be identified. We call this simple correspondence *influence*.

Definition 3.6 (Influence). *Given two sets S and U , an influence of U on S is a function $fl : S \rightarrow \text{pow}(U)$.*

The idea is that the generation of Λ_i -labellings of S_i from Λ_{i-1} -labellings of S_{i-1} is based, for each element s of S_i , on the elements of S_{i-1} influencing it, namely $fl(s)$. In the simplest case, a single influence of S_{i-1} on S_i is sufficient, and in the case where $S_{i-1} = S_i$ the identity function id can be used as a trivial form of influence. In general, one can consider an arbitrary number of influences between two subsequent stages, leading us to define n -influences.

Definition 3.7 (N -influence). *Given two sets S and U and an integer $n \geq 1$, a n -influence of U on S is a tuple $\langle fl_1, \dots, fl_n \rangle$ where for all $1 \leq i \leq n$ fl_i is an influence of U on S .*

Example 3.3 (continues Example 3.1). *To evaluate the schools on the basis of their students, we may distinguish those students (called full students for brevity) who did their education entirely in the school (intuitively they should count more in the evaluation of the school) from those (called partial students) who did their education only partially in the school. Hence, two distinct influences of the set of students $\{\text{John, Kris, Laura}\}$ on $\{\text{School1, School2, School3}\}$ are needed: one for full students and one for partial students. The relations between schools and their students can be thus represented by a 2-influence $\text{Stud} = \langle \text{fullStuds, partStuds} \rangle$ where fullStuds and partStuds identify full and partial students respectively. For instance, if John had his entire education at School1, while Kris attended both School1 and School2, and Laura attended both School2 and School3, then the influences fullStuds and partStuds can be represented as follows (with the obvious meaning):*

	School1	School2	School3
fullStuds(\cdot)	{John}	\emptyset	\emptyset
partStuds(\cdot)	{Kris}	{Kris, Laura}	{Laura}

In another stage assessing public and private schools, one can set down an influence of schools in $\{\text{School1, School2, School3}\}$ on the evaluation of different types of schools $\{\text{Pub, Priv}\}$, classifying schools as Pub or Priv. Suppose that the schools School1 and School2 are public, while School3 is private. This corresponds to the following 1-influence $\text{Schools} = \langle \text{schools} \rangle$:

	Pub	Priv
schools(\cdot)	{School1, School2}	{School3}

□

Given an influence fl belonging to some n -influence, the labels of influencing elements can be taken into account by means of their projections through fl on the influenced elements.

Definition 3.8 (Label projection). *Given a set $\mathfrak{L}^*(S_1, \Lambda_1)$, a set S_2 , and an influence fl of S_1 on S_2 , the projection of a multiset $\widehat{\mathfrak{L}} \in \mathfrak{L}^*(S_1, \Lambda_1)$ on S_2 based on fl is a function $\text{prj}[fl] : \mathfrak{L}^*(S_1, \Lambda_1) \times S_2 \rightarrow \text{pow}(\Lambda_1)$ such that for every $\widehat{\mathfrak{L}} \in \mathfrak{L}^*(S_1, \Lambda_1)$ and for every $s_2 \in S_2$:*

$$\text{prj}[fl](\widehat{\mathfrak{L}}, s_2) = \bigcup_{L \in {}^m \widehat{\mathfrak{L}}} \{L(s_1) \mid s_1 \in fl(s_2)\}.$$

In words, each element s_2 of S_2 receives, according to $\widehat{\mathfrak{L}}$, the set of labels ‘coming’ from the elements of the influence $fl(s_2)$. Note that projections are defined as regular sets as this is sufficient for the purposes of this paper. The study of more general versions is left to future work.

Example 3.4 (continues Example 3.1). *One can think of a context where each labelling of S_1 is projected separately (i.e. as a singleton) on the elements of S_2 . Continuing our example, for the singletons $\{L_0^1\}$, $\{L_0^2\}$ and $\{L_0^3\}$ of the set of students, and the functions fullStuds and partStuds from schools to students, we get the following projections:*

	School1	School2	School3
$\text{prj}[\text{fullStuds}](\{L_0^1\}, \cdot)$	{C}	\emptyset	\emptyset
$\text{prj}[\text{partStuds}](\{L_0^1\}, \cdot)$	{A}	{A, B}	{B}
$\text{prj}[\text{fullStuds}](\{L_0^2\}, \cdot)$	{C}	\emptyset	\emptyset
$\text{prj}[\text{partStuds}](\{L_0^2\}, \cdot)$	{B}	{B}	{B}
$\text{prj}[\text{fullStuds}](\{L_0^3\}, \cdot)$	{C}	\emptyset	\emptyset
$\text{prj}[\text{partStuds}](\{L_0^3\}, \cdot)$	{A}	{A}	{A}

In another context, the evaluation of full or partial students can be performed collectively, i.e. the labellings in $\widehat{\mathfrak{L}}$ are collectively projected from students to schools through the influences fullStuds and partStuds , as follows:

	School1	School2	School3
$\text{prj}[\text{fullStuds}](\widehat{\mathfrak{L}}, \cdot)$	{C}	\emptyset	\emptyset
$\text{prj}[\text{partStuds}](\widehat{\mathfrak{L}}, \cdot)$	{A, B}	{A, B}	{A, B}

□

As to the last ingredient needed to define polylabelling and monolabelling generators, we introduce synthesis operators of n -tuples of sets of labels (n -synthesizers for short). The idea is that the labels coming from a previous stage are synthesised into a label for the subsequent stage. In other words, a n -synthesizer takes in input a n -tuple of sets of labels (each of these sets being a subset of a given set of labels Λ_1) and converts them into a single label taken from a set of labels Λ_2 .

Definition 3.9 (N -synthesizer). *Given two sets of labels Λ_1 and Λ_2 and an integer $n \geq 1$, a n -synthesizer from Λ_1 to Λ_2 is a function $\text{syn} : (\text{pow}(\Lambda_1))^n \rightarrow \Lambda_2$.*

Notation 3.6. *The domain of definition of a n -synthesizer syn is denoted $\text{dom}(\text{syn})$; it is the set of elements of $(\text{pow}(\Lambda_1))^n$ for which the function syn is defined. For $1 \leq i \leq n$ and $\Lambda \subseteq \Lambda_1$ we say that $\Lambda \in \text{dom}^i(\text{syn})$ iff $\exists \Lambda'_1, \dots, \Lambda'_{i-1}, \Lambda'_{i+1}, \dots, \Lambda'_n \subseteq \Lambda_1$ such that $(\Lambda'_1, \dots, \Lambda'_{i-1}, \Lambda, \Lambda'_{i+1}, \dots, \Lambda'_n) \in \text{dom}(\text{syn})$.*

Intuitively, the n -tuple of sets of labels comes in some way from a previous stage through a n -influence, then a n -synthesizer is applied to obtain a resulting synthetic label from the set Λ_2 in the next stage. In this paper, we will be interested in 1-synthesizers, namely functions $syn : \text{pow}(\Lambda_1) \rightarrow \Lambda_2$; and in 2-synthesizers, namely functions $syn : \text{pow}(\Lambda_1) \times \text{pow}(\Lambda_1) \rightarrow \Lambda_2$.

As anticipated, n -influences, label projections and n -synthesizers are key elements in the definition of two main types of labelling generators. While being quite generic, these families are oriented to reflect some typical features of the argumentation formalisms we aim at capturing.

Definition 3.10 (Monolabelling and polylabelling generators). *Given two sets S_1 and S_2 , with relevant sets of labels Λ_1 and Λ_2 , a n -influence $Fl = \langle fl_1, \dots, fl_n \rangle$ of S_1 on S_2 and a n -synthesizer syn from Λ_1 to Λ_2 , the monolabelling and polylabelling Λ_2 -generators for S_2 based on Fl and syn are defined as follows:*

- the monolabelling Λ_2 -generator is such that for each $\widehat{\mathcal{L}} \in \mathfrak{L}^*(S_1, \Lambda_1)$:

$$\text{mgen}[Fl, syn](\widehat{\mathcal{L}}) = \{L \mid \forall s \in S_2, L(s) = syn(\text{prj}[fl_1](\widehat{\mathcal{L}}, s), \dots, \text{prj}[fl_n](\widehat{\mathcal{L}}, s))\};$$

- the polylabelling Λ_2 -generator is such that for each $\widehat{\mathcal{L}} \in \mathfrak{L}^*(S_1, \Lambda_1)$:

$$\text{pgen}[Fl, syn](\widehat{\mathcal{L}}) = \bigcup_{L' \in \widehat{\mathcal{L}}} \{L \mid \forall s \in S_2, L(s) = syn(\text{prj}[fl_1](\{L'\}, s), \dots, \text{prj}[fl_n](\{L'\}, s))\}.$$

In words, a monolabelling generator produces a singleton $\{L\}$ where for each element s of S_2 , $L(s)$ is obtained by applying syn to the projections of $\widehat{\mathcal{L}}$ based on the influences fl_1, \dots, fl_n in the n -influence Fl . A polylabelling generator in general produces instead multiple labellings such that each labelling L in $\text{pgen}[Fl, syn](\widehat{\mathcal{L}})$ corresponds to a labelling L' in $\widehat{\mathcal{L}}$. More precisely, L is obtained from L' by applying syn to the projections of $\{L'\}$ based on the influences fl_1, \dots, fl_n in the n -influence Fl .

Example 3.5 (continues Example 3.1). *Suppose the following (quite simplified) criterion to rank schools from the evaluation of their students: a school is Excellent if all its full students and all partial students got a mark A; a school is Good if all its full students got a mark A, and all its partial students a mark A or B, a school is Mediocre if all its full students got a mark A or B, and all its partial students got A or B or C, a school is poor otherwise. This criterion can be captured by a 2-synthesizer syn_1 from $\Lambda_0 = \{A, B, C, D\}$ to $\Lambda_1 = \{\text{Excel}, \text{Good}, \text{Med}, \text{Poor}\}$ which, assuming that the first argument of the 2-synthesizer concerns full students while the second one concerns partial students, can be defined for all $U, V \in \text{pow}(\Lambda_0)$ as follows:*

- $syn_1(U, V) = \text{Excel}$ iff $U \subseteq {}^4\{A\}$ and $V \subseteq \{A\}$;
- $syn_1(U, V) = \text{Good}$ iff $syn_1(U, V) \neq \text{Excel}$ and $U \subseteq \{A\}$ and $V \subseteq \{A, B\}$;
- $syn_1(U, V) = \text{Med}$ iff $syn_1(U, V) \neq \text{Excel}$ and $syn_1(U, V) \neq \text{Good}$ and $U \subseteq \{A, B\}$ and $V \subseteq \{A, B, C\}$;
- $syn_1(U, V) = \text{Poor}$ otherwise.

Then on the basis of syn_1 and the 2-influence $\text{Stud} = \langle \text{fullStuds}, \text{partStuds} \rangle$ as defined previously, by applying a polylabelling generator, the set $\widehat{\mathcal{L}} = \{(L_0^1, 1), (L_0^2, 1), (L_0^3, 2)\}$ gives rise to the set of labellings of the schools $\text{pgen}[\text{Stud}, syn_1](\widehat{\mathcal{L}}) = \{L_1^1, L_1^2\}$ where, from the projections of L_0^1 , we get:

4. The use of \subseteq covers the case of an empty set of students.

$$\begin{aligned} L_1^1(\text{School1}) &= \text{syn}_1(\{C\}, \{A\}) & (= \text{Poor}), \\ L_1^1(\text{School2}) &= \text{syn}_1(\emptyset, \{A, B\}) & (= \text{Good}), \\ L_1^1(\text{School3}) &= \text{syn}_1(\emptyset, \{B\}) & (= \text{Good}); \end{aligned}$$

and from the projections of L_0^2 we get the same labelling L_1^1 , though in a different way:

$$\begin{aligned} L_1^1(\text{School1}) &= \text{syn}_1(\{C\}, \{B\}) & (= \text{Poor}), \\ L_1^1(\text{School2}) &= \text{syn}_1(\emptyset, \{B\}) & (= \text{Good}), \\ L_1^1(\text{School3}) &= \text{syn}_1(\emptyset, \{B\}) & (= \text{Good}); \end{aligned}$$

and from the projections of L_0^3 we get:

$$\begin{aligned} L_1^2(\text{School1}) &= \text{syn}_1(\{C\}, \{A\}) & (= \text{Poor}), \\ L_1^2(\text{School2}) &= \text{syn}_1(\emptyset, \{A\}) & (= \text{Excel}), \\ L_1^2(\text{School3}) &= \text{syn}_1(\emptyset, \{A\}) & (= \text{Excel}). \end{aligned}$$

Alternatively, considering a monolabelling generator, we get $\text{mgen}[\text{Stud}, \text{syn}_1](\widehat{\mathcal{L}}) = \{L_1\}$ where from the collective projections of the labellings in $\widehat{\mathcal{L}}$ we get:

$$\begin{aligned} L_1(\text{School1}) &= \text{syn}_1(\{C\}, \{A, B\}) & (= \text{Poor}), \\ L_1(\text{School2}) &= \text{syn}_1(\emptyset, \{A, B\}) & (= \text{Good}), \\ L_1(\text{School3}) &= \text{syn}_1(\emptyset, \{A, B\}) & (= \text{Good}). \end{aligned}$$

Starting from the evaluations of students in $\widehat{\mathcal{L}}$, two distinct outcomes concerning schools have been produced, namely $\text{pgen}[\text{Stud}, \text{syn}_1](\widehat{\mathcal{L}})$ and $\text{mgen}[\text{Stud}, \text{syn}_1](\widehat{\mathcal{L}})$. The evaluations of schools can be then employed to assess the sectors of private and public schools, possibly leading to an investigation or not on the sectors, based on the set of labels $\Lambda_2 = \{\text{Invest}, \text{NoInvest}\}$ where *Invest* holds for ‘Investigation’ and *NoInvest* marks ‘No investigation’. We have already defined a 1-influence Schools between the two sectors and schools, so we need a 1-synthesizer from Λ_1 to Λ_2 . Suppose the following 1-synthesizer syn_2 is applied, where an investigation is performed if schools are not excellent or good, defined for each $S \in \text{pow}(\Lambda_1)$ as:

- $\text{syn}_2(S) = \text{NoInvest}$ iff $S = \emptyset$ or $S = \{\text{Excel}\}$ or $S = \{\text{Excel}, \text{Good}\}$;
- $\text{syn}_2(S) = \text{Invest}$ otherwise.

Based on Schools and syn_2 , a monolabelling generator can then be applied to derive an assessment of public and private schools from $\text{pgen}[\text{Stud}, \text{syn}_1](\widehat{\mathcal{L}}) = \{L_1^1, L_1^2\}$ and $\text{mgen}[\text{Stud}, \text{syn}_1](\widehat{\mathcal{L}}) = \{L_1\}$. In particular, letting $\text{mgen}[\text{Schools}, \text{syn}_2](\{L_1^1, L_1^2\}) = \{L_2\}$, we get:

$$\begin{aligned} L_2(\text{Pub}) &= \text{syn}_2(\{\text{Poor}, \text{Good}, \text{Excel}\}) & (= \text{Invest}), \\ L_2(\text{Priv}) &= \text{syn}_2(\{\text{Good}, \text{Excel}\}) & (= \text{NoInvest}), \end{aligned}$$

and, letting $\text{mgen}[\text{Schools}, \text{syn}_2](\{L_1\}) = \{L_2'\}$, we obtain:

$$\begin{aligned} L_2'(\text{Pub}) &= \text{syn}_2(\{\text{Poor}, \text{Good}\}) & (= \text{Invest}), \\ L_2'(\text{Priv}) &= \text{syn}_2(\{\text{Good}\}) & (= \text{Invest}). \end{aligned}$$

Altogether, through the sequence of examples we have presented two distinct multi-labelling systems for the students-school-sector evaluation pipeline. In particular letting:

- $\Xi_0 = \langle \{\text{John}, \text{Kris}, \text{Laura}\}, \{A, B, C, D\} \rangle$,
- $\Xi_1 = \langle \{\text{School1}, \text{School2}, \text{School3}\}, \{\text{Excel}, \text{Good}, \text{Med}, \text{Poor}\}, \text{pgen}[\text{Stud}, \text{syn}_1] \rangle$,
- $\Xi_1' = \langle \{\text{School1}, \text{School2}, \text{School3}\}, \{\text{Excel}, \text{Good}, \text{Med}, \text{Poor}\}, \text{mgen}[\text{Stud}, \text{syn}_1] \rangle$,

- $\Xi_2 = (\{\text{Pub}, \text{Priv}\}, \{\text{Invest}, \text{NoInvest}\}, \text{mgen}[\text{Schools}, \text{syn}_2])$,

we have illustrated both the multi-labelling system Ξ_0, Ξ_1, Ξ_2 (with final outcome L_2) and the multi-labelling system Ξ_0, Ξ'_1, Ξ_2 (with final outcome L'_2).

As a final remark, besides illustrating the various components of MLSs, the examples have also evidenced their intrinsic modularity, since a different MLS can be obtained from another one by replacing one stage with another one (in the example Ξ_1 with Ξ'_1) provided that their ‘interface’ with the previous and following stage remains the same. \square

3.3 Some Properties of MLSs

We will use MLSs to analyse and compare actual argumentation formalisms on a common ground consisting of abstract general properties. In particular we consider the notions of ‘coverage’, ‘indistinguishability’, and ‘refinement’.

The first property concerns labelling generators and requires that all output labellings are total.

Definition 3.11 (Coverage). *A labelling generator $\text{gen} : \mathfrak{L}^*(S_1, \Lambda_1) \rightarrow \mathfrak{L}^*(S_2, \Lambda_2)$ satisfies the property of coverage iff for every $\widehat{S}_1 \in \text{dom}(\text{gen})$ every labelling $L \in \text{gen}(\widehat{S}_1)$ is total.*

The second property concerns n -synthesizers and has to do with (in)distinguishability. In a few words, two sets of labels are indistinguishable at position i if, when present in the i -th argument of the synthesizer they give rise to the same result. In other words, a n -synthesizer is able to distinguish sets of labels at position i if there is at least a case where these sets of labels, as inputs at position i , make some difference. In that regard, we define two notions, namely, strong and weak i -indistinguishability.

Strong syn - i -indistinguishability accounts for the indistinguishability of sets of labels together with other labels, whereas weak syn - i -indistinguishability consists just in giving the same outcome when taken in isolation.

Definition 3.12 (Strong syn - i -indistinguishability). *Given a n -synthesizer $\text{syn} : (\text{pow}(\Lambda_1))^n \rightarrow \Lambda_2$, and $1 \leq i \leq n$, two sets of labels $\Lambda^a, \Lambda^b \subseteq \Lambda_1$ are strongly syn - i -indistinguishable, denoted as $\Lambda^a \equiv [i, \text{syn}]\Lambda^b$, iff for every sequence $(\Lambda^1, \dots, \Lambda^n) \in (\text{pow}(\Lambda_1))^n$ such that $(\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda^a, \Lambda^{i+1}, \dots, \Lambda^n), (\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda^b, \Lambda^{i+1}, \dots, \Lambda^n) \in \text{dom}(\text{syn})$ it holds that $\text{syn}(\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda^a, \Lambda^{i+1}, \dots, \Lambda^n) = \text{syn}(\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda^b, \Lambda^{i+1}, \dots, \Lambda^n)$.*

Definition 3.13 (Weak syn - i -indistinguishability). *Given a n -synthesizer $\text{syn} : (\text{pow}(\Lambda_1))^n \rightarrow \Lambda_2$, and $1 \leq i \leq n$, two sets of labels $\Lambda^a, \Lambda^b \subseteq \Lambda_1$ are weakly syn - i -indistinguishable, denoted as $\Lambda^a \sim [i, \text{syn}]\Lambda^b$, iff for every sequence $(\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^{i+1}, \dots, \Lambda^n) \in (\text{pow}(\Lambda_1))^{n-1}$ such that $(\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^a, \Lambda^{i+1}, \dots, \Lambda^n), (\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^b, \Lambda^{i+1}, \dots, \Lambda^n) \in \text{dom}(\text{syn})$ it holds that $\text{syn}(\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^a, \Lambda^{i+1}, \dots, \Lambda^n) = \text{syn}(\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^b, \Lambda^{i+1}, \dots, \Lambda^n)$.*

Example 3.6 (continues Example 3.1). *Given the 1-synthesizer syn_2 in Example 3.5, it can be seen that $\forall \Lambda' \subseteq \{\text{Excel}, \text{Good}, \text{Med}, \text{Poor}\}$ such that $\Lambda' \cup \{\text{Med}\}, \Lambda' \cup \{\text{Poor}\} \in \text{dom}(\text{syn}_2)$, it holds that $\text{syn}_2(\Lambda' \cup \{\text{Med}\}) = \text{syn}_2(\Lambda' \cup \{\text{Poor}\})$. Therefore, we have that $\{\text{Med}\} \equiv [1, \text{syn}_2]\{\text{Poor}\}$. \square*

It is easy to see that the relation of strong syn - i -indistinguishability is reflexive and symmetric. In general it may not be transitive. For example, for a 1-synthesizer, suppose $\Lambda_1^a \equiv [1, \text{syn}]\Lambda_1^b$ and $\Lambda_1^b \equiv [1, \text{syn}]\Lambda_1^c$, it may be the case that $\exists \Lambda' \subseteq \Lambda_1$ such that $\Lambda' \cup \Lambda_1^b \notin \text{dom}(\text{syn})$ while

$\Lambda' \cup \Lambda_1^a, \Lambda' \cup \Lambda_1^c \in \text{dom}(\text{syn})$ and $\text{syn}(\Lambda' \cup \Lambda_1^a) \neq \text{syn}(\Lambda' \cup \Lambda_1^c)$. Transitivity holds if such a case does not occur, as formalized by the following proposition whose proof is omitted.

Proposition 3.1. *Given a n -synthesizer $\text{syn} : (\text{pow}(\Lambda_1))^n \rightarrow \Lambda_2$, and three sets of labels $\Lambda_1^a, \Lambda_1^b, \Lambda_1^c \subseteq \Lambda_1$, if*

- $\Lambda_1^a \equiv [i, \text{syn}]\Lambda_1^b$ and
- $\Lambda_1^b \equiv [i, \text{syn}]\Lambda_1^c$ and
- $\{(\Lambda^1, \dots, \Lambda^n) \in (\text{pow}(\Lambda_1))^n \mid (\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda^a, \Lambda^{i+1}, \dots, \Lambda^n), (\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda^c, \Lambda^{i+1}, \dots, \Lambda^n) \in \text{dom}(\text{syn})\} \subseteq \{(\Lambda^1, \dots, \Lambda^n) \in (\text{pow}(\Lambda_1))^n \mid (\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda^b, \Lambda^{i+1}, \dots, \Lambda^n) \in \text{dom}(\text{syn})\}$

then $\Lambda_1^a \equiv [i, \text{syn}]\Lambda_1^c$.

Completely analogous considerations apply to weak syn - i -indistinguishability.

A basic property of the strong syn -indistinguishability relation will be useful in the following to simplify the analysis of actual formalisms: if a set Λ_1^a is strongly syn - i -indistinguishable from Λ_1^b then it is also strongly syn - i -indistinguishable from the union of Λ_1^b and any of the subsets of Λ_1^a (and hence in particular from $\Lambda_1^b \cup \Lambda_1^a$).

Proposition 3.2. *Given a n -synthesizer $\text{syn} : (\text{pow}(\Lambda_1))^n \rightarrow \Lambda_2$, and three sets of labels $\Lambda_1^a, \Lambda_1^b, \Lambda_1^c \subseteq \Lambda_1$ such that $\Lambda_1^c \subseteq \Lambda_1^a$ and $\Lambda^a \equiv [i, \text{syn}]\Lambda^b$, it holds that $\Lambda^a \equiv [i, \text{syn}]\Lambda^b \cup \Lambda_1^c$.*

Proof. From $\Lambda^a \equiv [i, \text{syn}]\Lambda^b$ and the facts that $(\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda_1^a \cup \Lambda_1^c, \Lambda^{i+1}, \dots, \Lambda^n) = (\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda_1^a, \Lambda^{i+1}, \dots, \Lambda^n) \in \text{dom}(\text{syn})$ and $(\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda_1^b \cup \Lambda_1^c, \Lambda^{i+1}, \dots, \Lambda^n) \in \text{dom}(\text{syn})$ it follows that $\text{syn}(\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda_1^a, \Lambda^{i+1}, \dots, \Lambda^n) = \text{syn}(\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda_1^a \cup \Lambda_1^c, \Lambda^{i+1}, \dots, \Lambda^n) = \text{syn}(\Lambda^1, \dots, \Lambda^{i-1}, \Lambda^i \cup \Lambda_1^b \cup \Lambda_1^c, \Lambda^{i+1}, \dots, \Lambda^n)$ as desired. \square

Based on strong indistinguishability, it is possible to define also a notion of (in)sensitivity of a n -synthesizer to one of its parameters. The idea is that syn is i -insensitive if the actual value of the i -th parameter does not affect in any case the label returned by syn .

Definition 3.14 (i -(in)sensitivity). *Letting $1 \leq i \leq n$, a n -synthesizer $\text{syn} : (\text{pow}(\Lambda_1))^n \rightarrow \Lambda_2$ is:*

- i -insensitive iff for all $\Lambda^a, \Lambda^b \subseteq \Lambda_1$ it holds that Λ^a and Λ^b are strongly syn - i -indistinguishable;
- i -sensitive otherwise.

Finally, we are also interested in comparing different labellings and the n -synthesizer which generates them in terms of refinement. Intuitively, a labelling is more refined than another one if the former provides a more articulated representation of every label encompassed by the latter. Then a n -synthesizer can be said more refined than another one if it produces more refined labellings when employed in a labelling generator. Formally, the definition of refinement for labellings and n -synthesizers can be based on the definition of refinement for partitions (Ellerman, 2010).

Definition 3.15 (Partition). *A partition π on a set U is a set $\{B\}_{B \in \pi}$ of nonempty subsets (or ‘blocks’) $B \subseteq U$ that are disjoint and whose union is U .*

Every total function $f : D \rightarrow C$ defines a partition of its domain of definition D , denoted π_f . Each block of this partition is a set $B_f(c) = \{d \in D \mid f(d) = c\}$, and corresponds to an element $c \in C$ such that $c = f(d)$ for some $d \in D$. In particular, a total Λ -labelling L of a set S defines a partition π_L of S , where each block consists of the elements of S having the same label. In the following, for the sake of conciseness, $\lambda(L)$ denotes the block $B_L(\lambda) = \{u \mid L(u) = \lambda\}$.

Definition 3.16 (Distinction). *A pair $(u, u') \in U \times U$ is a distinction of the partition π if there are distinct subsets $B, B' \in \pi$ with $u \in B$ and $u' \in B'$.*

The set of distinctions of a partition π , called its distinction set and denoted $\text{dit}(\pi) \subseteq U \times U$, is the partition seen as a relation:

$$\text{dit}(\pi) = \bigcup_{B, B' \in \pi, B \neq B'} B \times B' \quad (1)$$

A refinement can be then defined in terms of inclusion of distinctions.

Definition 3.17 (Refinement). *Given two total functions $f_1 : D \rightarrow C_1$ and $f_2 : D \rightarrow C_2$, f_2 refines (or is a refinement of) f_1 iff $\text{dit}(\pi_{f_1}) \subseteq \text{dit}(\pi_{f_2})$.*

Thus f_2 refines f_1 if it allows to make more (or at least the same) distinctions with respect to the elements of their common domain. Equivalently, we can say that f_2 refines f_1 iff for every $c_2 \in C_2$, there is $c_1 \in C_1$ such that $B_{f_2}(c_2) \subseteq B_{f_1}(c_1)$. In words, for each c_2 in the codomain of f_2 there is c_1 in the codomain of f_1 , denoted in the following as $\text{coa}(c_2)$, such that c_2 is returned by f_2 in a subset of the cases where c_1 is returned by f_1 . Conversely, for each c_1 in the codomain of f_1 , there is a subset $\text{ref}(c_1) \triangleq \{c_2 \in C_2 \mid B_{f_2}(c_2) \subseteq B_{f_1}(c_1)\}$ such that c_1 is returned by f_1 in all and only the cases where f_2 returns one of the elements of $\text{ref}(c_1)$. Thus the elements of $\text{ref}(c_1)$ can be regarded as refining c_1 in the output of f_2 with respect to f_1 . Note that $\text{ref}(c_1) = \{c_2 \in C_2 \mid c_1 = \text{coa}(c_2)\}$.

In the case of (total) labellings, we can say that a Λ_2 -labelling refines a Λ_1 -labelling iff for every label $\lambda_2 \in \Lambda_2$, there is a label $\lambda_1 \in \Lambda_1$ such that $\lambda_2(L_2) \subseteq \lambda_1(L_1)$. An analogous observation applies to (total) n -synthesizers.

It is useful to note that, in the context of labelling generators, the refinement relation between n -synthesizers induces a refinement relation between the generated labellings for both monolabelling and polylabelling generators.

Proposition 3.3. *Given two sets S_1 and S_2 , a n -influence $Fl = \langle fl_1, \dots, fl_n \rangle$ of S_1 on S_2 , a n -synthesizer syn_1 from Λ to Λ_1 , and an n -synthesizer syn_2 from Λ to Λ_2 , such that syn_2 refines syn_1 , let $\text{mgen}[Fl, syn_1]$ be the monolabelling Λ_1 -labelling generator based on Fl and syn_1 and $\text{mgen}[Fl, syn_2]$ be the monolabelling Λ_2 -labelling generator based on Fl and syn_2 . For every $\widehat{\mathcal{L}} \in \mathcal{L}^*(S_1, \Lambda_1)$, letting $\text{mgen}[Fl, syn_1](\widehat{\mathcal{L}}) = \{L_1\}$, and $\text{mgen}[Fl, syn_2](\widehat{\mathcal{L}}) = \{L_2\}$, it holds that L_2 is a refinement of L_1 .*

Proof. For every $s \in S_2$ we have that $L_1(s) = syn_1(\text{prj}[fl_1](\widehat{\mathcal{L}}, s), \dots, \text{prj}[fl_n](\widehat{\mathcal{L}}, s))$ and $L_2(s) = syn_2(\text{prj}[fl_1](\widehat{\mathcal{L}}, s), \dots, \text{prj}[fl_n](\widehat{\mathcal{L}}, s))$. From syn_2 being a refinement of syn_1 it follows that for every $\lambda_2 \in \Lambda_2$, there is $\lambda_1 = \text{coa}(\lambda_2) \in \Lambda_1$, such that $B_{syn_2}(\lambda_2) \subseteq B_{syn_1}(\lambda_1)$. It follows that, for every $s \in S_2$, if $L_2(s) = \lambda_2$ for some $\lambda_2 \in \Lambda_2$ then $L_1(s) = \text{coa}(\lambda_2)$. Then for every $\lambda_2 \in \Lambda_2$ there is $\lambda_1 \in \Lambda_1$ such that $\lambda_2(L_2) \subseteq \lambda_1(L_1)$ and the conclusion follows. \square

Proposition 3.4. *Given two sets S_1 and S_2 , a n -influence $Fl = \langle fl_1, \dots, fl_n \rangle$ of S_1 on S_2 , a n -synthesizer syn_1 from Λ to Λ_1 , and a n -synthesizer syn_2 from Λ to Λ_2 , such that syn_2 refines syn_1 , let $\text{pgen}[Fl, syn_1]$ be the polylabelling Λ_1 -labelling generator based on Fl and syn_1 and $\text{pgen}[Fl, syn_2]$ be the polylabelling Λ_2 -labelling generator based on Fl and syn_2 . For every $\widehat{\mathcal{L}} \in \mathcal{L}^*(S_1, \Lambda_1)$, letting $\text{pgen}[Fl, syn_1](\widehat{\mathcal{L}}) = \mathcal{L}_1$, and $\text{pgen}[Fl, syn_2](\widehat{\mathcal{L}}) = \mathcal{L}_2$, it holds that for every $L_2 \in \mathcal{L}_2$ there is $L_1 \in \mathcal{L}_1$ such that L_2 is a refinement of L_1 .*

Proof. For every labelling $L_2 \in \mathfrak{L}_2$ there exists a labelling $L' \in^m \widehat{\mathfrak{L}} : \forall s \in S_2, L_2(s) = \text{syn}_2(\text{prj}[fl_1](\{L'\}, s), \dots, \text{prj}[fl_n](\{L'\}, s))$. Then there exists also $L_1 \in \mathfrak{L}_1$ such that $\forall s \in S_2, L_1(s) = \text{syn}_1(\text{prj}[fl_1](\{L'\}, s), \dots, \text{prj}[fl_n](\{L'\}, s))$. The fact that L_2 is a refinement of L_1 can then be proved following the same line of the proof of Proposition 3.3. \square

Example 3.7 (continues Example 3.1). *Instead of the 1-synthesizer syn_2 as defined in Example 3.5, suppose the following operator syn'_2 is applied such that a deep investigation (DeepInvest) is performed if a school is poorly rated, defined for every $S \in \text{pow}(\Lambda_1)$ as:*

- $\text{syn}'_2(S) = \text{NoInvest}$ iff $S = \emptyset$ or $S = \{\text{Excel}\}$ or $S = \{\text{Excel}, \text{Good}\}$;
- $\text{syn}'_2(S) = \text{DeepInvest}$ iff $\text{Poor} \in S$;
- $\text{syn}'_2(S) = \text{RegInvest}$ otherwise.

Letting $\{L'_2\} = \text{mgen}[\text{Schools}, \text{syn}'_2](\{L_1^1, L_1^2\})$ we get:

$$\begin{aligned} L'_2(\text{Pub}) &= \text{syn}'_2(\{\text{Poor}, \text{Good}, \text{Excel}\}) && (= \text{DeepInvest}), \\ L'_2(\text{Priv}) &= \text{syn}'_2(\{\text{Good}, \text{Excel}\}) && (= \text{NoInvest}), \end{aligned}$$

and letting $\{L''_2\} = \text{mgen}[\text{Schools}, \text{syn}'_2](\{L_1\})$ we get:

$$\begin{aligned} L''_2(\text{Pub}) &= \text{syn}'_2(\{\text{Poor}, \text{Good}\}) && (= \text{DeepInvest}), \\ L''_2(\text{Priv}) &= \text{syn}'_2(\{\text{Good}\}) && (= \text{RegInvest}). \end{aligned}$$

Clearly, syn'_2 refines syn_2 , with $\text{ref}(\text{Invest}) = \{\text{RegInvest}, \text{DeepInvest}\}$ and $\text{ref}(\text{NoInvest}) = \{\text{NoInvest}\}$. Hence, if the last stage $\Xi_2 = \langle \{\text{Pub}, \text{Priv}\}, \{\text{Invest}, \text{NoInvest}\}, \text{mgen}[\text{Schools}, \text{syn}_2] \rangle$ is replaced with $\Xi'_2 = \langle \{\text{Pub}, \text{Priv}\}, \{\text{RegInvest}, \text{DeepInvest}, \text{NoInvest}\}, \text{mgen}[\text{Schools}, \text{syn}'_2] \rangle$, then every final labelling produced by Ξ'_2 is a refinement of the final labelling produced by Ξ_2 for the same input. \square

4. Modeling the Evaluation of Arguments and Statements with Multi-labelling Systems

In this section, we discuss how multi-labelling systems can describe different forms of evaluation of arguments and statements in the context of argumentation processes. In very general terms, we assume that the argumentation process consists in three main phases:

1. building arguments which support conclusions expressed in a given language;
2. evaluating the status of the arguments;
3. evaluating the status of the statements of the language, given the status of the arguments.

Describing the process backwards, the final goal is the evaluation of statements, which may concern what to believe (as, in the leading example, assessing whether I should believe I have a certain disease) and/or what to do (which would occur in the example if there were rules indicating which therapy to undertake for a given disease). The evaluation of statements results from the evaluation of arguments, which are assumed to be the essential entities representing the reasoning or dialectical activities of one or more agents in a given context. Clearly, arguments (and typically their relationships) must be formed by the involved agent(s) before they can be assessed.

Multi-labelling systems provide a general modelling tool based on labellings for the phases 2 and 3, taking for granted that a set of arguments has been produced in phase 1. The general idea is that the evaluation activities carried out in the phases 2 and 3 are represented as a multi-labelling system (Ξ_0, \dots, Ξ_m) where:

- in the first stage $\Xi_0 = \langle S_0, \Lambda_0 \rangle$, S_0 is the set of arguments, and Λ_0 is the set of labels assigned to arguments in the first evaluation step concerning argument acceptance. As commonly set up in many argumentation formalisms, we assume that the outcome of the argument acceptance evaluation consists of a set of Λ_0 -labellings of S_0 ;
- in the last stage $\Xi_m = \langle S_m, \Lambda_m, gen_m \rangle$, S_m is the set of language statements whose status has to be evaluated using the set of labels Λ_m , this final evaluation being generated using the generator gen_m ;
- the stages Ξ_1, \dots, Ξ_{m-1} capture the intermediate evaluation steps leading from the initial assessment of argument acceptance to the final evaluation of statements carried out in stage Ξ_m .

In this context, we identify two main approaches: the *argument-focused* approach and the *statement-focused* approach. These two approaches differ in the way the initial assessment of argument acceptance is used in subsequent stages (see Figure 1).

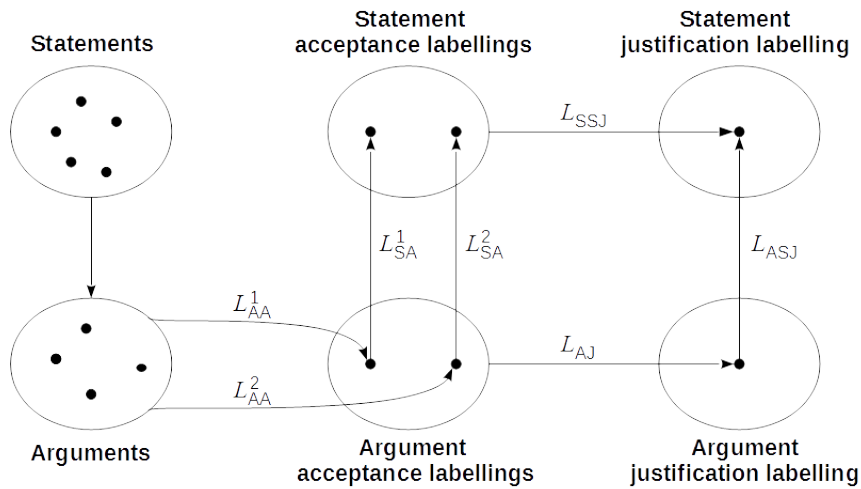


Figure 1: Overview of multi-labelling systems for argumentation (AF and SF approaches).

In the *argument-focused* (AF) approach, argument acceptance gives rise to argument justification at a second stage, from which statement justification is derived at a third stage. In the *statement-focused* (SF) approach, argument acceptance is projected on statements, giving rise to statement acceptance at a second stage, from which statement justification is again derived at a third stage. The description and formal definitions of these different stages are provided in the sequel, preceded by some basic concepts.

4.1 Argument Production

The first phase of the process regards the production of a set of arguments \mathcal{A} whose structure and mutual relationships are left unspecified. Argument production being taken for granted, the only relevant property for our purposes is that each argument $A \in \mathcal{A}$ has a conclusion, denoted as $con(A)$, belonging to a language \mathcal{L} , which is a set of statements.

Definition 4.1 (Language). *A language \mathcal{L} is a set of statements.*

We do not make any assumptions on the set of arguments, while we assume that the language is equipped with a contrariness relation. In its simplest form, the contrariness relation corresponds to the traditional notion of negation but other more general forms of contrariness have been considered in the literature (Modgil & Prakken, 2013; Baroni, Giacomin, & Liao, 2015). To encompass this wider view, we assume a contrariness relation cnt , allowing the existence of multiple (or no) contraries for each statement, and a larger compatibility with a variety of argumentation formalisms.

Definition 4.2 (Contrariness). *Given a language \mathcal{L} , a contrariness relation is a function $\text{cnt} : \mathcal{L} \rightarrow \text{pow}(\mathcal{L})$ identifying for each statement φ a set $\text{cnt}(\varphi)$ of statements, called the contraries of φ .*

The language can be closed under negation (i.e. every negation of its statements are included in the language), but in general, this constraint is not necessary.

Example 4.1 (continues Example 1.1). *For the sake of conciseness, we do not define a full language for representing all the details of our example (e.g. we do not consider statements concerning the ‘clinical pictures’) as this would make the presentation longer while being irrelevant to the contribution of the paper. Thus we focus on a restricted set of statements which is sufficient for illustrating our proposal: the language \mathcal{L} for our medical enquiry is the set of statements s_1 , $\neg s_1$, s_2 and its negation denoted $\neg s_2$, and s_3 (we do not include $\neg s_3$ in the language for the sake of conciseness, because it would not introduce any significant difference).*

$$\mathcal{L} = \{s_1, \neg s_1, s_2, \neg s_2, s_3\}.$$

The contrariness relation over this language is as follows:

$$\text{cnt}(s_1) = \neg s_1, \quad \text{cnt}(\neg s_1) = s_1, \quad \text{cnt}(s_2) = \{\neg s_2\}, \quad \text{cnt}(\neg s_2) = \{s_2\}, \quad \text{cnt}(s_3) = \emptyset.$$

□

The outcomes of the argument production stage can be summarised in an abstract form as an argument-conclusion structure.

Definition 4.3 (Argument-conclusion structure). *An argument-conclusion structure (ACS) is a triple $\langle \mathcal{L}, \mathcal{A}, \text{con} \rangle$ where \mathcal{L} is a language, \mathcal{A} is a finite set of arguments and $\text{con} : \mathcal{A} \rightarrow \mathcal{L}$ is a relation associating every argument with its conclusion.*

Argument-conclusion structures differ from common argumentation frameworks (Dung, 1995; Amgoud, Cayrol, Lagasque-Schiex, & Livet, 2008), because they do not cater for attacks or other structural relations over arguments while they include the notion of conclusion which is abstracted away in those frameworks.

Example 4.2 (continues Example 1.1). *Let us model our medical enquiry with an ACS $\text{AC}_1 = \langle \mathcal{L}, \mathcal{A}, \text{con} \rangle$ where:*

- the language is as in Example 4.1:

$$\mathcal{L} = \{s_1, \neg s_1, s_2, \neg s_2, s_3\};$$

- the set of arguments \mathcal{A} is such that every statement of the language \mathcal{L} is the conclusion of one and only one argument, except for statement s_3 which has no supporting argument:

$$\mathcal{A} = \{A_1, A_2, A_3, A_4\};$$

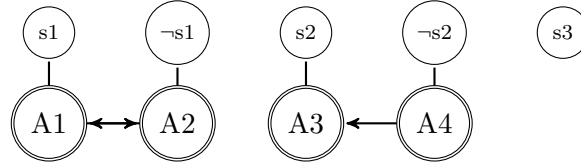


Figure 2: Argumentation framework superposed with its ACS.

- the relation con associating every argument with its conclusion is such that:

$$\text{con}(A1) = s1, \quad \text{con}(A2) = \neg s1, \quad \text{con}(A3) = s2, \quad \text{con}(A4) = \neg s2.$$

Note that, again for the sake of conciseness, we abstract away the underlying argument construction process. Intuitively each argument is built using as premises some evidences (e.g. the opinion of a doctor or the outcome of a test) and applying some rules (e.g. that if a doctor formulates a diagnosis about the presence/absence of a disease this provides a reason to believe that the disease is present/absent). A detailed modelling would also possibly involve some ancillary arguments (e.g. in some argumentation formalisms evidences are regarded as arguments themselves) which we again leave implicit since they would lengthen the presentation and possibly involve some formalism-specific issues without playing any actual role in the discussion.

As to attacks, they are not captured by ACSs but will play a role when we discuss the behaviour of formalisms from the literature. In that regard, we remark again that the example could be formalised in slightly different ways in the context of different formalisms, but, leaving these differences apart, it is meant to capture some common basic patterns: two mutually attacking arguments, one argument defeating another one, and a statement not involved in the current argumentation process (and therefore not being the conclusion of any argument). Accordingly, we assume that arguments A1 and A2 attack each other, and that attacks can be determined also by preferences (as commonly adopted in structured argumentation formalisms) and argument A4 is preferred to A3, and consequently that argument A4 attacks argument A3 but not vice versa. Eventually, the corresponding argumentation framework can be superposed with its ACS. as illustrated in Figure 2, where directed arrows represent attacks.

□

Note that some elements of \mathcal{L} may not play the role of conclusions, e.g. if \mathcal{L} encompasses negation as failure.

In general each statement $\varphi \in \mathcal{L}$ is supported by a (possibly empty) set of arguments denoted as $\text{sup}(\varphi)$. This notion can obviously be extended to sets of statements as in the following definition.

Definition 4.4 (Supporting arguments). *Given an ACS $\langle \mathcal{L}, \mathcal{A}, \text{con} \rangle$ and a set $\Phi \subseteq \mathcal{L}$, the set of supporting arguments of Φ is defined as*

$$\text{sup}(\Phi) \triangleq \{A \in \mathcal{A} \mid \text{con}(A) \in \Phi\}.$$

Summing up, at this stage, arguments are produced and related to statements of the considered language to build ACSs. The next stage concerns the labelling of arguments to reflect their acceptance statuses.

4.2 Argument Acceptance

Assuming that an ACS $\langle \mathcal{L}, \mathcal{A}, \text{con} \rangle$ is given, the first stage, which is common to the *argument-focused* approach and the *statement-focused* approach, concerns the acceptance evaluation of the arguments in \mathcal{A} using a set of labels Λ_{AA} . Each label in Λ_{AA} represents an individual argument acceptance status and a labelling L_{AA} altogether represents a ‘reasonable’ viewpoint (in general among many possible ones) about the acceptance of the arguments in \mathcal{A} .

Thus, at this stage, a set of argument acceptance labellings has to be generated by using a set of labels Λ_{AA} : we assume that there is a mechanism that produces this set of acceptance labellings of arguments, without specifying any detail of how it works and of how it relies on argument structure and/or their relationships.

Definition 4.5 (Argument acceptance labelling and evaluation). *Given an ACS $\mathcal{AC} = \langle \mathcal{L}, \mathcal{A}, \text{con} \rangle$ and a set of acceptance labels Λ_{AA} , an argument acceptance Λ_{AA} -labelling for \mathcal{A} is a Λ_{AA} -labelling of \mathcal{A} . A Λ_{AA} -acceptance evaluation for \mathcal{A} , denoted as $\mathfrak{L}_{AA}(\mathcal{A})$, is a set of argument acceptance Λ_{AA} -labellings for \mathcal{AC} .*

Example 4.3 (continues Example 1.1). *As argument acceptance labellings, we may consider Dung style complete $\{\text{IN}, \text{OUT}, \text{UN}\}$ -labellings (a brief recall of the essentials of Dung’s theory is given in Appendix A.), such that:*

- *an argument is labelled IN iff all its attacking arguments are OUT;*
- *an argument is labelled OUT iff there exists an attacking argument labelled IN.*

Assuming that arguments A1 and A2 attack each other, and that argument A4 attacks argument A3, as illustrated in Figure 2, we have three complete $\{\text{IN}, \text{OUT}, \text{UN}\}$ -labellings:

	A1	A2	A3	A4
$L_{AA}^1(\cdot)$	IN	OUT	OUT	IN
$L_{AA}^2(\cdot)$	OUT	IN	OUT	IN
$L_{AA}^3(\cdot)$	UN	UN	OUT	IN

The $\{\text{IN}, \text{OUT}, \text{UN}\}$ -acceptance evaluation is thus $\mathfrak{L}_{AA} = \{L_{AA}^1, L_{AA}^2, L_{AA}^3\}$.

□

In the MLS model, the first stage is defined by the pair $\Xi_0 = \langle \mathcal{A}, \Lambda_{AA} \rangle$ which indicates that for every possible acceptance evaluation for $\mathfrak{L}_{AA}(\mathcal{A})$ it holds that $\mathfrak{L}_{AA}(\mathcal{A}) \subseteq \mathfrak{L}(\mathcal{A}, \Lambda_{AA})$. Different ways of using the set of acceptance labellings $\mathfrak{L}_{AA}(\mathcal{A})$ give rise to two alternatives for the subsequent stages. In a nutshell, in the *argument-focused* approach, the set of acceptance labelling is projected on arguments and then synthesised, giving rise to an argument justification stage, while in the *statement-focused* approach, the focus is transferred from arguments to their conclusions, giving rise to a statement acceptance stage.

4.3 Argument-Focused (AF) Approach

In the AF approach, acceptance labellings are projected on arguments and then synthesised, thus yielding an argument justification stage, eventually leading to a statement justification stage.

ARGUMENT JUSTIFICATION

The assessment of argument justification consists in assigning to each argument A a justification label, taken from a set Λ_{AJ} of argument justification labels, based on the acceptance labels assigned to

A by the elements of $\mathcal{L}_{AA}(\mathcal{A})$. In words, the justification label of A can be regarded as a synthesis (or aggregation) of the acceptance labels of A itself. This is captured by a stage $\Xi_1^{AF} = \langle \mathcal{A}, \Lambda_{AJ}, \text{mgen}_{AJ} \rangle$ where $\text{mgen}_{AJ} = \text{mgen}[\text{ID}, \text{syn}_{AJ}]$ is a monolabelling generator based on the identity 1-influence $\text{ID} \triangleq \langle \text{id} \rangle$ and on a 1-synthesizer syn_{AJ} from Λ_{AA} to Λ_{AJ} .

Example 4.4 (continues Example 1.1). *Assume that $\Lambda_{AJ} = \{\text{SKJ}, \text{CRJ}, \text{NOJ}\}$ and the 1-synthesizer syn_{AJ} from $\Lambda_{AA} = \{\text{IN}, \text{OUT}, \text{UN}\}$ to Λ_{AJ} is defined, for every $S \in \text{pow}(\Lambda_{AA})$, as follows:*

- $\text{syn}_{AJ}(S) = \text{SKJ}$ iff $S = \{\text{IN}\}$,
- $\text{syn}_{AJ}(S) = \text{CRJ}$ iff $S \supseteq \{\text{IN}\}$,
- $\text{syn}_{AJ}(S) = \text{NOJ}$ otherwise.

Then, applying Definition 3.10, and letting $\{L_{AJ}\} = \text{mgen}_{AJ}(\mathcal{L}_{AA})$, we get for each argument A $L_{AJ}(A) = \text{syn}_{AJ}(\text{prj}[\text{id}](\mathcal{L}_{AA}, A))$. The global projection of \mathcal{L}_{AA} on every argument is as follows:

	A1	A2	A3	A4
$\text{prj}[\text{id}](\mathcal{L}_{AA}, \cdot)$	{IN, OUT, UN}	{IN, OUT, UN}	{OUT}	{IN}

Applying syn_{AJ} it then follows that:

	A1	A2	A3	A4
$L_{AJ}(\cdot)$	CRJ	CRJ	NOJ	SKJ

□

AF STATEMENT JUSTIFICATION.

The subsequent and last stage in the AF approach deals with the justification status of statements. We assume that the justification status of statements is represented by a labelling L_{ASJ} , based on a set of statement justification labels Λ_{SJ} , which is derived from the argument justification labelling produced in the previous stage. In particular, we assume that, for a statement φ , both the arguments whose conclusion is φ and the arguments whose conclusion is a contrary of φ play a role in determining the justification of φ . To capture that a statement is influenced by these two sets of arguments, we introduce the 2-influence of \mathcal{A} on \mathcal{L} defined as: $\text{SC} = \langle \text{supp}, \text{cntr} \rangle$ where, for each statement φ , $\text{supp}(\varphi) = \text{sup}(\{\varphi\})$ and $\text{cntr}(\varphi) = \text{sup}(\text{cnt}(\varphi))$. Then we can define the last stage of the AF approach: $\Xi_2^{AF} = \langle \mathcal{L}, \Lambda_{ASJ}, \text{mgen}_{ASJ} \rangle$ where $\text{mgen}_{ASJ} = \text{mgen}[\text{SC}, \text{syn}_{ASJ}]$ is a monolabelling generator based on SC and on a 2-synthesizer syn_{ASJ} from Λ_{AJ} to Λ_{ASJ} .

Example 4.5 (continues Example 1.1). *Applying Definition 3.10 and letting $\{L_{ASJ}\} = \text{mgen}_{ASJ}(\{L_{AJ}\})$, for each statement φ we get:*

$$L_{ASJ}(\varphi) = \text{syn}_{ASJ}(\text{prj}[\text{supp}](\{L_{AJ}\}, \varphi), \text{prj}[\text{cntr}](\{L_{AJ}\}, \varphi)).$$

As to the projections of labellings from supporting arguments to statements through influence supp we get:

	s1	\neg s1	s2	\neg s2	s3
$\text{prj}[\text{supp}](\{L_{AJ}\}, \cdot)$	{CRJ}	{CRJ}	{NOJ}	{SKJ}	∅

As to the projections of labellings from contrary-supporting arguments to statements through influence cntr we get:

	s1	\neg s1	s2	\neg s2	s3
prj[ctr]({L _{AJ} }, ·)	{CRJ}	{CRJ}	{SKJ}	{NOJ}	∅

As to the statement justification labels and the relevant 2-synthesizer, we can consider here two variants, both very simple and ignoring the contrary-supporting arguments (as we will see this choice is not uncommon in the literature). First, assuming $\Lambda_{ASJ} = \{\text{inj}, \text{noj}\}$, the 2-synthesizer syn_{ASJ} from Λ_{AJ} to Λ_{SJ} can be defined for $S_1, S_2 \in \text{pow}(\Lambda_{AJ})$ as follows:

- $\text{syn}_{ASJ}(S_1, S_2) = \text{inj}$ iff $\text{SKJ} \in S_1$;
- $\text{syn}_{ASJ}(S_1, S_2) = \text{noj}$ otherwise.

According to this specification, we get the following $\{\text{inj}, \text{noj}\}$ -labelling as a result for statement justification:

	s1	\neg s1	s2	\neg s2	s3
$L_{ASJ}(\cdot)$	noj	noj	noj	inj	noj

As variant, if $\Lambda'_{ASJ} = \{\text{skj}, \text{crj}, \text{noj}\}$ and the 2-synthesizer syn'_{ASJ} from Λ_{AJ} to Λ_{SJ} is defined for $S_1, S_2 \in \text{pow}(\Lambda_{AJ})$ as follows

- $\text{syn}'_{ASJ}(S_1, S_2) = \text{skj}$ iff $\text{SKJ} \in S_1$;
- $\text{syn}'_{ASJ}(S_1, S_2) = \text{crj}$ iff $\text{CRJ} \in S_1$ and $\text{SKJ} \notin S_1$;
- $\text{syn}'_{ASJ}(S_1, S_2) = \text{noj}$ otherwise;

then we get the following $\{\text{skj}, \text{crj}, \text{noj}\}$ -labelling L'_{ASJ} for statement justification:

	s1	\neg s1	s2	\neg s2	s3
$L'_{ASJ}(\cdot)$	crj	crj	noj	skj	noj

In this example, syn'_{ASJ} refines syn_{ASJ} , and hence, in particular, the $\{\text{skj}, \text{crj}, \text{noj}\}$ -labelling L'_{ASJ} refines the $\{\text{inj}, \text{noj}\}$ -labelling L_{ASJ} . □

4.4 Statement-Focused (SF) Approach

In the SF approach, every argument acceptance labelling results into a statement acceptance labelling (the statement acceptance stage), and then statement acceptance labellings are projected on statements to obtain the justification labellings of statements.

STATEMENT ACCEPTANCE.

At this stage each argument labelling belonging to $\mathcal{L}_{AA}(\mathcal{AC})$ generates a statement labelling. The idea is that, to generate an acceptance label for a statement φ in the context of each labelling in $\mathcal{L}_{AA}(\mathcal{AC})$, the justification labels of the (generally many) arguments whose conclusion is φ and the justification labels of the (generally many) arguments whose conclusion is a contrary of φ are taken into account. This is captured by a stage $\Xi_1^{\text{SF}} = \langle \mathcal{L}, \Lambda_{SA}, \text{pgen}_{SA} \rangle$ where $\text{pgen}_{SA} = \text{pgen}[\text{SC}, \text{syn}_{SA}]$ is a polylabelling generator based on the same 2-influence SC defined in the previous section and on a 2-synthesizer syn_{SA} from Λ_{AA} to Λ_{SA} .

Example 4.6 (continues Example 1.1). *Given the set of argument labellings $\mathcal{L}_{AA} = \{L_{AA}^1, L_{AA}^2, L_{AA}^3\}$, the labelling generator pgen_{SA} yields a set $\{L_{SA}^1, L_{SA}^2, L_{SA}^3\}$ such that for every statement φ :*

$$L_{SA}^i(\varphi) = \text{syn}_{SA}(\text{prj}[\text{supp}]({L_{AA}^i}, \varphi), \text{prj}[\text{ctr}]({L_{AA}^i}, \varphi)).$$

Let us first examine $\text{prj}[\text{supp}](L_{AA}^i, \varphi)$ and $\text{prj}[\text{cntr}](L_{AA}^i, \varphi)$ for all statements for the three labellings and then illustrate an example of 2-synthesizer syn_{SA} . We have the following projections:

	s1	\neg s1	s2	\neg s2	s3
$\text{prj}[\text{supp}]({L_{AA}^1}, \cdot)$	{IN}	{OUT}	{OUT}	{IN}	\emptyset
$\text{prj}[\text{cntr}]({L_{AA}^1}, \cdot)$	{OUT}	{IN}	{IN}	{OUT}	\emptyset
$\text{prj}[\text{supp}]({L_{AA}^2}, \cdot)$	{OUT}	{IN}	{OUT}	{IN}	\emptyset
$\text{prj}[\text{cntr}]({L_{AA}^2}, \cdot)$	{IN}	{OUT}	{IN}	{OUT}	\emptyset
$\text{prj}[\text{supp}]({L_{AA}^3}, \cdot)$	{UN}	{UN}	{OUT}	{IN}	\emptyset
$\text{prj}[\text{cntr}]({L_{AA}^3}, \cdot)$	{UN}	{UN}	{IN}	{OUT}	\emptyset

Let us now assume that $\Lambda_{SA} = \{\text{in}, \text{no}\}$ and introduce the (again rather simple and ignoring contraries) 2-synthesizer syn_{SA} from Λ_{AA} to Λ_{SA} defined for $S_1, S_2 \in \text{pow}(\Lambda_{AA})$ as follows:

- $\text{syn}_{SA}(S_1, S_2) = \text{in}$ iff $\text{IN} \in S_1$;
- $\text{syn}_{SA}(S_1, S_2) = \text{no}$ otherwise.

According to the above definitions, we obtain the following acceptance statement labellings:

	s1	\neg s1	s2	\neg s2	s3
$L_{SA}^1(\cdot)$	in	no	no	in	no
$L_{SA}^2(\cdot)$	no	in	no	in	no
$L_{SA}^3(\cdot)$	no	no	no	in	no

and thus:

$$\text{pgen}_{SA}(\mathfrak{L}_{AA}) = \{L_{SA}^1, L_{SA}^2, L_{SA}^3\}.$$

□

SF STATEMENT JUSTIFICATION

In the last stage of the SF approach, the idea is that a single justification labelling for statements, based on a set of statement justification labels Λ_{SJ} , is derived from the set of (in general many) statement acceptance labellings produced in the previous step. In particular, to derive the justification label for each statement φ , the acceptance labels assigned to φ itself and to its contraries are taken into account. To capture that a statement is influenced both by itself and its contraries, we introduce the 2-influence of \mathcal{L} on \mathcal{L} , defined as: $\text{IC} = \langle \text{id}, \text{cnt} \rangle$. Then we can define the last stage of the SF approach: $\Xi_2^{\text{SF}} = \langle \mathcal{L}, \Lambda_{SSJ}, \text{mgen}_{SSJ} \rangle$ where $\text{mgen}_{SSJ} = \text{mgen}[\text{IC}, \text{syn}_{SSJ}]$ is a monolabelling generator based on IC and on a 2-synthesizer syn_{SSJ} from Λ_{SA} to Λ_{SSJ} .

Example 4.7 (continues Example 1.1). Let $\mathfrak{L}_{SA} = \text{pgen}_{SA}(\mathfrak{L}_{AA})$ be the set of statement acceptance labellings produced in the previous stage. Applying Definition 3.10 and letting $\{L_{SSJ}\} = \text{mgen}_{SSJ}(\mathfrak{L}_{SA})$ for each statement φ we get:

$$L_{SSJ}(\varphi) = \text{syn}_{SSJ}(\text{prj}[\text{id}](\mathfrak{L}_{SA}, \varphi), \text{prj}[\text{cnt}](\mathfrak{L}_{SA}, \varphi)).$$

The projections of \mathfrak{L}_{SA} on each statement through the identity influence id and the contrary influence cnt are as follows:

	s1	¬s1	s2	¬s2	s3
prj[id](\mathfrak{L}_{SA}, \cdot)	{in, no}	{in, no}	{no}	{in}	{no}
prj[cnt](\mathfrak{L}_{SA}, \cdot)	{in, no}	{in, no}	{in}	{no}	\emptyset

Then, assuming $\Lambda_{SSJ} = \{\text{skj}, \text{crj}, \text{noj}\}$, a 2-synthesizer syn_{SSJ} from Λ_{SA} to Λ_{SSJ} can be defined for $S_1, S_2 \in \text{pow}(\Lambda_{SA})$ as follows:

- $\text{syn}_{SSJ}(S_1, S_2) = \text{skj}$ if $S_1 = \{\text{in}\}$;
- $\text{syn}_{SSJ}(S_1, S_2) = \text{crj}$ iff $S_1 \supseteq \{\text{in}\}$;
- $\text{syn}_{SSJ}(S_1, S_2) = \text{noj}$ otherwise.

Accordingly, we have:

	s1	¬s1	s2	¬s2	s3
$L_{SSJ}(\cdot)$	crj	crj	noj	skj	noj

□

To recap, we have identified two families of multi-labelling systems (MLSs) for argumentation, namely argument-focused (AF) MLSs and statement-focused (SF) MLSs. Both AF and SF MLSs are based on an ACS $\langle \mathcal{L}, \mathcal{A}, \text{con} \rangle$, and on a set of acceptance labels Λ_{AA} for arguments, giving rise to the initial stage $\Xi_0 = \langle \mathcal{A}, \Lambda_{AA} \rangle$. Then a generic AF MLS consists of the sequence $(\Xi_0, \Xi_1^{\text{AF}}, \Xi_2^{\text{AF}})$, such that $\Xi_1^{\text{AF}} = \langle \mathcal{A}, \Lambda_{AJ}, \text{mgen}_{AJ} \rangle$ and $\Xi_2^{\text{AF}} = \langle \mathcal{L}, \Lambda_{ASJ}, \text{mgen}_{ASJ} \rangle$, and, referring to Definition 3.5, it can be simply defined as the composition $\text{mgen}_{ASJ} \circ \text{mgen}_{AJ}$. A generic SF MLS consists of the sequence $(\Xi_0, \Xi_1^{\text{SF}}, \Xi_2^{\text{SF}})$, such that $\Xi_1^{\text{SF}} = \langle \mathcal{L}, \Lambda_{SA}, \text{pgen}_{SA} \rangle$ and $\Xi_2^{\text{SF}} = \langle \mathcal{L}, \Lambda_{SSJ}, \text{mgen}_{SSJ} \rangle$, and, referring to Definition 3.5, it can be simply defined as the composition $\text{mgen}_{SSJ} \circ \text{pgen}_{SA}$.

5. Comparing AF and SF Approaches

In the AF approach, the outcomes of the second stage (i.e. the acceptance labellings of arguments) are first projected and synthesised on the argument themselves, giving rise to argument justification. Then argument justification outcomes are transferred to statements and synthesised in turn, taking contraries into account, to get statement justification. On the other hand in the SF approach the outcomes of the argument acceptance stage are immediately transferred and synthesised on statements, giving rise to statement acceptance. Then the acceptance outcomes of a statement and of its contraries are taken into account to derive statement justification.

One may wonder whether, under the assumptions we made, the two approaches feature the same expressiveness, i.e. whether any statement justification labelling produced by an AF MLS can be obtained by a corresponding SF MLS and vice versa.

In general, it turns out that the AF and SF approaches are incomparable in terms of expressiveness. This can be shown by using some simple examples with the set of argument acceptance labels $\Lambda_{AA} = \{\text{IN}, \text{OUT}\}$.

5.1 A Distinction Expressible only by the SF Approach

C1. Consider a first case C1 where there are (possibly among others) two arguments A and B such that, for some statement φ , $\text{sup}(\{\varphi\}) = \{A, B\}$ (i.e. they have the same conclusion φ and no other arguments conclude φ). For simplicity, let us also assume that $\text{sup}(\text{cnt}(\varphi)) = \emptyset$. Suppose that the outcome of the argument acceptance stage consists of two labellings, i.e. $\mathfrak{L}_{AA}(\mathcal{A}) = \{L_{AA}^1, L_{AA}^2\}$ such that $L_{AA}^1(A) = \text{IN}$ and $L_{AA}^1(B) = \text{OUT}$, while $L_{AA}^2(A) = \text{OUT}$ and $L_{AA}^2(B) = \text{IN}$.

- In the AF approach, at the argument justification stage, and whatever adopted 1-synthesizer syn_{AJ} , we get

$$\begin{aligned} L_{AJ}(A) &= syn_{AJ}(prj[id](\mathfrak{L}_{AA}, A)) & L_{AJ}(B) &= syn_{AJ}(prj[id](\mathfrak{L}_{AA}, B)) \\ &= syn_{AJ}(\{IN, OUT\}) & &= syn_{AJ}(\{IN, OUT\}) \\ &= \lambda & &= \lambda \end{aligned}$$

for some $\lambda \in \Lambda_{AJ}$. At the statement justification stage, we have $prj[supp](\{L_{AJ}\}, \varphi) = \{\lambda\}$ and $prj[ctr](\{L_{AJ}\}, \varphi) = \emptyset$. Then, whatever 2-synthesizer syn_{ASJ} is adopted, $L_{ASJ}(A)$ functionally depends on the pair $(\{\lambda\}, \emptyset)$ i.e. $L_{ASJ}(\varphi) = syn_{ASJ}(\{\lambda\}, \emptyset)$.

- In the SF approach, at the statement acceptance stage, we get

$$\begin{aligned} L_{SA}^1(\varphi) &= syn_{SA}(prj[supp](\{L_{AA}^1\}, \varphi), \emptyset) & L_{SA}^2(\varphi) &= syn_{SA}(prj[supp](\{L_{AA}^2\}, \varphi), \emptyset) \\ &= syn_{SA}(\{IN, OUT\}, \emptyset) & &= syn_{SA}(\{IN, OUT\}, \emptyset) \\ &= \lambda^0 & &= \lambda^0 \end{aligned}$$

for some $\lambda^0 \in \Lambda_{SA}$. At the statement justification stage, $prj[id](\mathfrak{L}_{SA}, \varphi) = \{\lambda^0\}$ and thus $L_{SSJ}(\varphi) = syn_{SSJ}(\{\lambda^0\}, \emptyset)$.

C2. Consider a second case C2 where there is a single argument A with conclusion φ , i.e. $sup(\{\varphi\}) = \{A\}$, and assume again that $sup(cnt(\varphi)) = \emptyset$ and that the outcome of the argument acceptance stage consists of two labellings, i.e. $\mathfrak{L}_{AA}(\mathcal{AC}) = \{L_{AA}^1, L_{AA}^2\}$ such that $L_{AA}^1(A) = IN$ while $L_{AA}^2(A) = OUT$.

- In the AF approach, at the argument justification stage, as in the case C1,

$$\begin{aligned} L_{AJ}(A) &= syn_{AJ}(prj[id](\mathfrak{L}_{AA}, A)) \\ &= syn_{AJ}(\{IN, OUT\}) \\ &= \lambda \end{aligned}$$

Hence, at the statement justification stage, we get $prj[supp](\{L_{AJ}\}, \varphi) = \{\lambda\}$ and $prj[ctr](\{L_{AJ}\}, \varphi) = \emptyset$ from which $L_{ASJ}(\varphi) = syn_{ASJ}(\{\lambda\}, \emptyset)$ must be the same as in case C1.

- In the SF approach, at the statement acceptance stage, we get

$$\begin{aligned} L_{SA}^1(\varphi) &= syn_{SA}(prj[supp](\{L_{AA}^1\}, \varphi), \emptyset) & L_{SA}^2(\varphi) &= syn_{SA}(prj[supp](\{L_{AA}^2\}, \varphi), \emptyset) \\ &= syn_{SA}(\{IN\}, \emptyset) & &= syn_{SA}(\{OUT\}, \emptyset) \\ &= \lambda^1 & &= \lambda^2 \end{aligned}$$

for some $\lambda^1, \lambda^2 \in \Lambda_{SA}$. At the statement justification stage, $prj[id](\mathfrak{L}_{SA}, \varphi) = \{\lambda^1, \lambda^2\}$, and $L_{SSJ}(\varphi) = syn_{SSJ}(\{\lambda^1, \lambda^2\}, \emptyset)$, which may give rise to a different outcome than in case C1.

In the cases C1 and C2 the statement justification of φ must be the same by the AF approach, while the statement justification of φ may be different by the SF approach. Hence, we conclude that the AF approach is unable to capture some distinctions that can be captured by the SF approach.

5.2 A Distinction Expressible only by the AF Approach

C3. Consider a case C3, where, similarly to case C1, there are two arguments A and B such that, for some statement φ , $sup(\{\varphi\}) = \{A, B\}$ and $sup(cnt(\varphi)) = \emptyset$. Suppose also that the outcome

of the argument acceptance stage consists of two labellings, i.e. $\mathfrak{L}_{AA}(\mathcal{A}) = \{L_{AA}^1, L_{AA}^2\}$ such that $L_{AA}^1(A) = \text{IN}$ and $L_{AA}^1(B) = \text{IN}$, while $L_{AA}^2(A) = \text{IN}$ and $L_{AA}^2(B) = \text{OUT}$.

- In the AF approach, at the argument justification stage, we get

$$\begin{aligned} L_{AJ}(A) &= \text{syn}_{AJ}(\text{prj}[\text{id}](\mathfrak{L}_{AA}, A)) & L_{AJ}(B) &= \text{syn}_{AJ}(\text{prj}[\text{id}](\mathfrak{L}_{AA}, B)) \\ &= \text{syn}_{AJ}(\{\text{IN}\}) & &= \text{syn}_{AJ}(\{\text{IN}, \text{OUT}\}) \\ &= \lambda & &= \lambda' \end{aligned}$$

It follows that $\text{prj}[\text{supp}](\{L_{AJ}\}, \varphi) = \{\lambda, \lambda'\}$ while $\text{prj}[\text{cntr}](\{L_{AJ}\}, \varphi) = \emptyset$, from which $L_{ASJ}(\varphi) = \text{syn}_{ASJ}(\{\lambda, \lambda'\}, \emptyset)$.

- In the SF approach, at the statement acceptance stage, we get

$$\begin{aligned} L_{SA}^1(\varphi) &= \text{syn}_{SA}(\text{prj}[\text{supp}](\{L_{AA}^1\}, \varphi), \emptyset) & L_{SA}^2(\varphi) &= \text{syn}_{SA}(\text{prj}[\text{supp}](\{L_{AA}^2\}, \varphi), \emptyset) \\ &= \text{syn}_{SA}(\{\text{IN}\}, \emptyset) & &= \text{syn}_{SA}(\{\text{IN}, \text{OUT}\}, \emptyset) \\ &= \lambda^1 & &= \lambda^2 \end{aligned}$$

for some $\lambda^1, \lambda^2 \in \Lambda_{SA}$. Then $\text{prj}[\text{id}](\mathfrak{L}_{SA}, \varphi) = \{\lambda^1, \lambda^2\}$ on which $L_{SSJ}(\varphi)$ functionally depends.

C4. Consider now a case C4 which differs from case C3 because there is an additional argument labelling L_{AA}^3 , and thus $\mathfrak{L}_{AA}(\mathcal{A}) = \{L_{AA}^1, L_{AA}^2, L_{AA}^3\}$ where L_{AA}^1 and L_{AA}^2 are as in case C3, while $L_{AA}^3(A) = \text{OUT}$ and $L_{AA}^3(B) = \text{IN}$.

- In the AF approach, at the argument justification stage, we get

$$\begin{aligned} L_{AJ}(A) &= \text{syn}_{AJ}(\text{prj}[\text{id}](\mathfrak{L}_{AA}, A)) & L_{AJ}(B) &= \text{syn}_{AJ}(\text{prj}[\text{id}](\mathfrak{L}_{AA}, B)) \\ &= \text{syn}_{AJ}(\{\text{IN}, \text{OUT}\}) & &= \text{syn}_{AJ}(\{\text{IN}, \text{OUT}\}) \\ &= \lambda' & &= \lambda' \end{aligned}$$

It follows that $\text{prj}[\text{supp}](\{L_{AJ}\}, \varphi) = \{\lambda'\}$ and $L_{ASJ}(\varphi) = \text{syn}_{ASJ}(\{\lambda'\}, \emptyset)$, which may give rise to a different outcome than in case C3.

- In the SF approach, at the statement acceptance stage, we get $L_{SA}^1(\varphi)$ and $L_{SA}^2(\varphi)$ as above and

$$\begin{aligned} L_{SA}^3(\varphi) &= \text{syn}_{SA}(\text{prj}[\text{supp}](\{L_{AA}^3\}, \varphi), \emptyset) \\ &= \text{syn}_{SA}(\{\text{IN}, \text{OUT}\}, \emptyset) \\ &= \lambda^2 \end{aligned}$$

hence $L_{SA}^3(\varphi) = L_{SA}^2(\varphi) = \lambda^2$. Then also in the case C4 we get $\text{prj}[\text{id}](\mathfrak{L}_{SA}, \varphi) = \{\lambda^1, \lambda^2\}$ and $L_{SSJ}(\varphi)$ must be the same as in case C3.

In the cases C3 and C4 the statement justification of φ may be different by the AF approach, while the statement justification of φ must be the same by the SF approach. Hence, we conclude that the SF approach is unable to capture some distinctions that can be captured by the AF approach.

While, as shown above, in general the two approaches are not comparable, if one restricts to the case where \mathfrak{L}_{AA} is a singleton⁵ it turns out that the SF approach is strictly more expressive than the AF approach.

Proposition 5.1. *Let \mathcal{AC} be an ACS. For any AF MLS $\text{mgen}_{ASJ} \circ \text{mgen}_{AJ}$, there exists a SF MLS $\text{mgen}_{SSJ} \circ \text{pgen}_{SA}$ such that for every $\mathfrak{L}_{AA} \in \mathfrak{L}^*(\mathcal{A}, \Lambda_{AA})$ with $|\mathfrak{L}_{AA}| = 1$, it holds that $\text{mgen}_{ASJ} \circ \text{mgen}_{AJ}(\mathfrak{L}_{AA}) = \text{mgen}_{SSJ} \circ \text{pgen}_{SA}(\mathfrak{L}_{AA})$.*

Proof. The basic idea is that, in any case where $|\mathfrak{L}_{AA}| = 1$, for any AF MLS $\text{mgen}_{ASJ} \circ \text{mgen}_{AJ}$ one can build a SF MLS $\text{mgen}_{SSJ} \circ \text{pgen}_{SA}$ such that $\mathfrak{L}_{SA} = \{\text{L}_{ASJ}\}$ and then build the trivial SF statement justification labelling such that $\text{L}_{SSJ} = \text{L}_{ASJ}$. Details of the proof follow.

- For any AF MLS $\text{mgen}_{ASJ} \circ \text{mgen}_{AJ}$ assume $|\mathfrak{L}_{AA}| = 1$ and let in particular $\mathfrak{L}_{AA} = \{\text{L}_{AA}^1\}$. For every $A \in \mathcal{A}$ we get $\text{prj}[\text{id}](\mathfrak{L}_{AA}, A) = \{\text{L}_{AA}^1(A)\}$. Then $\text{L}_{AJ}(A) = \text{syn}_{AJ}(\{\text{L}_{AA}^1(A)\})$ and for every statement $\varphi \in \mathcal{L}$, $\text{L}_{ASJ}(\varphi) = \text{syn}_{ASJ}(\text{prj}[\text{supp}](\{\text{L}_{AJ}\}, \varphi), \text{prj}[\text{cntr}](\{\text{L}_{AJ}\}, \varphi))$.
- The same labelling can be obtained for every statement $\varphi \in \mathcal{L}$ in the SF approach as follows. We have $\text{pgen}_{SA}(\{\text{L}_{AA}^1\}) = \{\text{L}_{SA}^1\}$ where for each statement $\varphi \in \mathcal{L}$ we have:

$$\text{L}_{SA}^1(\varphi) = \text{syn}_{SA}(\text{prj}[\text{supp}](\{\text{L}_{AA}^1\}, \varphi), \text{prj}[\text{cntr}](\{\text{L}_{AA}^1\}, \varphi)).$$

Let us assume that $\Lambda_{SA} = \Lambda_{ASJ}$ and define the function $F_{AJ} : \text{pow}(\Lambda_{AA}) \rightarrow \text{pow}(\Lambda_{AJ})$ such that $F_{AJ}(\Lambda) = \{\text{syn}_{AJ}(\{\lambda\}) \mid \lambda \in \Lambda\}$ for every $\Lambda \subseteq \Lambda_{AA}$. For $\Lambda_1, \Lambda_2 \subseteq \Lambda_{AA}$ one can then define $\text{syn}_{SA}(\Lambda_1, \Lambda_2) = \text{syn}_{ASJ}(F_{AJ}(\Lambda_1), F_{AJ}(\Lambda_2))$. Hence, for each statement $\varphi \in \mathcal{L}$ we have:

$$\begin{aligned} \text{L}_{SA}^1(\varphi) &= \text{syn}_{ASJ}(F_{AJ}(\text{prj}[\text{supp}](\{\text{L}_{AA}^1\}, \varphi)), F_{AJ}(\text{prj}[\text{cntr}](\{\text{L}_{AA}^1\}, \varphi))) \\ &= \text{syn}_{ASJ}(\{\text{syn}_{AJ}(\{\lambda\}) \mid \lambda \in \text{prj}[\text{supp}](\{\text{L}_{AA}^1\}, \varphi)\}, \\ &\quad \{\text{syn}_{AJ}(\{\lambda\}) \mid \lambda \in \text{prj}[\text{cntr}](\{\text{L}_{AA}^1\}, \varphi)\}) \\ &= \text{syn}_{ASJ}(\text{prj}[\text{supp}](\{\text{L}_{AJ}\}, \varphi), \text{prj}[\text{cntr}](\{\text{L}_{AJ}\}, \varphi)). \end{aligned}$$

That is, one can define syn_{SA} such that, given any acceptance labelling L_{AA}^1 , for every statement $\varphi \in \mathcal{L}$, $\text{L}_{SA}^1(\varphi) = \text{L}_{ASJ}(\varphi)$. Letting then, for any $\Lambda_1, \Lambda_2 \subseteq \Lambda_{SA} (= \Lambda_{ASJ})$, $\text{syn}_{SSJ}(\Lambda_1, \Lambda_2) = \lambda$ whenever $\Lambda_1 = \{\lambda\}$ (the actual definition of syn_{SSJ} in the cases where Λ_1 is not a singleton is irrelevant since Λ_1 is guaranteed to be a singleton under the hypothesis $|\mathfrak{L}_{AA}| = 1$), we finally get that for every statement $\varphi \in \mathcal{L}$, $\text{L}_{SSJ}(\varphi) = \text{L}_{ASJ}(\varphi)$. □

The converse does not hold: indeed even with a single-status semantics there are distinctions which can be expressed by the SF approach while they cannot be captured by the AF approach, as shown by the following example.

C5. Consider three arguments A, B and C such that $\text{con}(A) = \varphi$, $\text{con}(B) = \psi$, $\text{con}(C) = \chi$ and suppose $\text{cnt}(\varphi) = \{\psi, \chi\}$, $\text{cnt}(\psi) = \{\varphi\}$, $\text{cnt}(\chi) = \{\varphi\}$. Suppose now that $\mathfrak{L}_{AA} = \{\text{L}_{AA}\}$ with $\text{L}_{AA}(A) = \lambda_1$, $\text{L}_{AA}(B) = \lambda_2$, $\text{L}_{AA}(C) = \lambda_3$.

- In the SF approach we first observe that:

$$\begin{aligned} \text{L}_{SA}(\varphi) &= \text{syn}_{SA}(\{\lambda_1\}, \{\lambda_2, \lambda_3\}), \\ \text{L}_{SA}(\psi) &= \text{syn}_{SA}(\{\lambda_2\}, \{\lambda_1\}), \\ \text{L}_{SA}(\chi) &= \text{syn}_{SA}(\{\lambda_3\}, \{\lambda_1\}), \end{aligned}$$

5. This case is significant since, in abstract argumentation, it corresponds to the adoption of a single-status semantics (see Appendix A).

and it is possible that the three labels $L_{SA}(\varphi)$, $L_{SA}(\psi)$ and $L_{SA}(\chi)$ are pairwise different. Then, for the statement φ , we get $L_{SSJ}(\varphi) = \text{syn}_{SSJ}(\{L_{SA}(\varphi)\}, \{L_{SA}(\psi), L_{SA}(\chi)\})$.

- In the AF approach we get

$$\begin{aligned} L_{AJ}(A) &= \text{syn}_{AJ}(\{\lambda_1\}), \\ L_{AJ}(B) &= \text{syn}_{AJ}(\{\lambda_2\}), \\ L_{AJ}(C) &= \text{syn}_{AJ}(\{\lambda_3\}). \end{aligned}$$

For the statement φ , $L_{ASJ}(\varphi) = \text{syn}_{ASJ}(\{\text{syn}_{AJ}(\{\lambda_1\})\}, \{\text{syn}_{AJ}(\{\lambda_2\}), \text{syn}_{AJ}(\{\lambda_3\})\})$.

C6. Consider now a variant of case C5 where the only difference is that $\text{cnt}(\chi) = \{\varphi, \psi\}$. We suppose that the acceptance evaluation is the same, i.e. $\mathfrak{L}_{AA} = \{L_{AA}\}$ with $L_{AA}(A) = \lambda_1$, $L_{AA}(B) = \lambda_2$, $L_{AA}(C) = \lambda_3$.

- In the SF approach, letting L'_{SA} be the statement acceptance labelling produced, we get:

$$\begin{aligned} L'_{SA}(\varphi) &= \text{syn}_{SA}(\{\lambda_1\}, \{\lambda_2, \lambda_3\}), \\ L'_{SA}(\psi) &= \text{syn}_{SA}(\{\lambda_2\}, \{\lambda_1\}), \\ L'_{SA}(\chi) &= \text{syn}_{SA}(\{\lambda_3\}, \{\lambda_1, \lambda_2\}), \end{aligned}$$

thus it may be the case that $L'_{SA}(\chi) \neq L_{SA}(\chi)$, while $L'_{SA}(\varphi) = L_{SA}(\varphi)$ and $L'_{SA}(\psi) = L_{SA}(\psi)$. Letting L'_{SSJ} be the statement acceptance labelling produced in this case we get $L'_{SSJ}(\varphi) = \text{syn}_{SSJ}(\{L'_{SA}(\varphi)\}, \{L'_{SA}(\psi), L'_{SA}(\chi)\})$ which may differ from $L_{SSJ}(\varphi) = \text{syn}_{SSJ}(\{L_{SA}(\varphi)\}, \{L_{SA}(\psi), L_{SA}(\chi)\})$.

- In the AF approach it can be easily seen that the difference introduced in the set $\text{cnt}(\chi)$ does not affect the argument justification labelling, i.e. letting L'_{AJ} be the argument justification labelling produced in this case we get

$$\begin{aligned} L'_{AJ}(A) &= \text{syn}_{AJ}(\{\lambda_1\}), \\ L'_{AJ}(B) &= \text{syn}_{AJ}(\{\lambda_2\}), \\ L'_{AJ}(C) &= \text{syn}_{AJ}(\{\lambda_3\}), \end{aligned}$$

hence $L'_{AJ} = L_{AJ}$. Letting L'_{ASJ} be the statement justification labelling produced in this case we get $L'_{ASJ}(\varphi) = \text{syn}_{ASJ}(\{\text{syn}_{AJ}(\{\lambda_1\})\}, \{\text{syn}_{AJ}(\{\lambda_2\}), \text{syn}_{AJ}(\{\lambda_3\})\})$. Therefore $L'_{ASJ}(\varphi) = L_{ASJ}(\varphi)$, i.e. the evaluation of statement φ must be the same as the evaluation in case C5.

Summing up, in the cases C5 and C6, where $|\mathfrak{L}_{AA}| = 1$, the statement justification of φ may be different by the SF approach, while the statement justification of φ must be the same by the AF approach. Hence, we conclude that the AF approach is unable to capture some distinctions which can be captured by the SF approach. In combination with Proposition 5.1 this shows that, in the context of our formalization, under the hypothesis that $|\mathfrak{L}_{AA}| = 1$ the AF approach is strictly less expressive than the SF approach.

It can be observed⁶ that while the incomparability in terms of expressiveness of the AF approach and the SF approach at a general level does not depend on the use of contraries in the evaluation (in fact in sections 5.1 and 5.2 we could assume for simplicity $\text{sup}(\text{cnt}(\varphi)) = \emptyset$), the fact that the AF approach is strictly less expressive than the SF approach under the hypothesis that $|\mathfrak{L}_{AA}| = 1$ strictly depends on the role of contraries in evaluation. In particular, if one additionally assumes that $\text{cnt}(\varphi) = \emptyset$ for every statement φ , the AF and the SF approach have the same expressiveness. This ensues from the following proposition (in combination with the result of Proposition 5.1).

6. The authors are grateful to one of the anonymous reviewers for pointing out this fact.

Proposition 5.2. *Let \mathcal{AC} be an ACS with \mathcal{L} such that for every $\varphi \in \mathcal{L}$ $\text{cnt}(\varphi) = \emptyset$. For any SF MLS $\text{mgen}_{\text{SSJ}} \circ \text{pgen}_{\text{SA}}$, there exists a AF MLS $\text{mgen}_{\text{ASJ}} \circ \text{mgen}_{\text{AJ}}$ such that for every $\mathfrak{L}_{\text{AA}} \in \mathfrak{L}^*(\mathcal{A}, \Lambda_{\text{AA}})$ with $|\mathfrak{L}_{\text{AA}}| = 1$, it holds that $\text{mgen}_{\text{ASJ}} \circ \text{mgen}_{\text{AJ}}(\mathfrak{L}_{\text{AA}}) = \text{mgen}_{\text{SSJ}} \circ \text{pgen}_{\text{SA}}(\mathfrak{L}_{\text{AA}})$.*

Proof. For any SF MLS $\text{mgen}_{\text{ASJ}} \circ \text{mgen}_{\text{AJ}}$ assume $|\mathfrak{L}_{\text{AA}}| = 1$ and let in particular $\mathfrak{L}_{\text{AA}} = \{\text{L}_{\text{AA}}^1\}$. The labelling generator pgen_{SA} then yields a set $\{\text{L}_{\text{SA}}^1\}$ consisting of a single labelling L_{SA}^1 such that for every $\varphi \in \mathcal{L}$

$$\text{L}_{\text{SA}}^1(\varphi) = \text{syn}_{\text{SA}}(\text{prj}[\text{supp}]({\text{L}_{\text{AA}}^1}, \varphi), \emptyset).$$

Letting $\{\text{L}_{\text{SSJ}}\} = \text{mgen}_{\text{SSJ}}(\{\text{L}_{\text{SA}}^1\})$ for each statement φ we get:

$$\text{L}_{\text{SSJ}}(\varphi) = \text{syn}_{\text{SSJ}}(\{\text{L}_{\text{SA}}^1(\varphi)\}, \emptyset).$$

The same labelling can be obtained for every statement $\varphi \in \mathcal{L}$ in the AF approach as follows. Assume a trivial argument justification stage where syn_{AJ} is the identity function, i.e. such that for each argument A $\text{L}_{\text{AJ}}(A) = \text{syn}_{\text{AJ}}(\text{prj}[\text{id}](\mathfrak{L}_{\text{AA}}, A)) = \text{syn}_{\text{AJ}}(\text{L}_{\text{AA}}^1(A)) = \text{L}_{\text{AA}}^1(A)$. Then letting $\{\text{L}_{\text{ASJ}}\} = \text{mgen}_{\text{ASJ}}(\{\text{L}_{\text{AJ}}\})$, for each statement φ we get:

$$\text{L}_{\text{ASJ}}(\varphi) = \text{syn}_{\text{ASJ}}(\text{prj}[\text{supp}]({\text{L}_{\text{AA}}^1}, \varphi), \emptyset)$$

and it is possible to obtain $\text{L}_{\text{ASJ}}(\varphi) = \text{L}_{\text{SSJ}}(\varphi)$ by defining for any $\Lambda_1, \Lambda_2 \subseteq \Lambda_{\text{ASJ}}$, $\text{syn}_{\text{ASJ}}(\Lambda_1, \Lambda_2) = \text{syn}_{\text{SSJ}}(\{\text{syn}_{\text{SA}}(\Lambda_1, \Lambda_2)\}, \emptyset)$. □

5.3 The AF and SF Approaches in Perspective

The AF and SF approaches are two prominent alternative ways to conceive the derivation of statement justification labellings from argument acceptance labellings. Indeed, both approaches can be put in correspondence with the design choices of several formalisms in the literature, as it will be shown in Section 7. In that regard, we will see that some literature formalisms can be reconstructed both in the AF and in the SF approach, while others belong to just one of the two camps. Whether other sensible ways of deriving statement justification can be devised appears to be an interesting question for future research.

The incomparability results between the AF and SF approaches suggest that there would be no point in debating about which of the two approaches is better in general. However, as shown in Proposition 5.1, the SF approach turns out to be more expressive in the specific case where the starting point is represented by a unique argument acceptance labelling and the contrariness relation is taken into account. While our general analysis establishes some sort of upper limit to the expressiveness of each approach, an actual argumentation formalism may share the limits of both.

For this reason, also a question like ‘For which kind of application an approach is better than the other?’ would not be appropriate, since application-oriented evaluations should be carried out at the level of actual formalisms and with reference to their expressiveness (crucially depending on the synthesizers adopted) rather than at the level of the approach they belong to (possibly both).

That being stated, membership in one approach clearly implies inheriting its limitations, and can be used to draw some high level considerations. In particular, at an intuitive level, it can be noted that the AF approach ‘forgets’ earlier the information about the multiplicity of argument acceptance labellings (synthesised by a unique justification labelling), while the SF approach ‘forgets’ earlier

the relation between arguments and conclusions, by moving from argument acceptance labellings to statement acceptance labellings.

As it will be discussed extensively in Section 11, such limitations have some consequences on the treatment of the so-called floating conclusions. In particular, the SF approach is able to capture a specific notion of skeptical acceptance of these conclusions which is not encompassed by the AF approach. On the other hand, however, examples in Section 5.2 show that the AF approach can ‘react’ to an addition to the set of argument acceptance labellings (which might correspond to an additional viewpoint in a judgement aggregation context (Caminada & Pigozzi, 2011)), while (at least in some cases) such an addition may have no effect in the SF approach. Floating conclusions and the treatment of multiple viewpoints are largely debated issues in the literature, with a variety of proposals available, some inherently subjective facets involved, and many research questions still open.

While not aiming at providing definite answers to inherently open questions, we believe that further studies of the AF and SF approaches may be useful to identify, at a general level, theoretical and practical limits of argumentation formalisms in various contexts, like those mentioned above. Moreover, these studies can stimulate, in a formalism agnostic manner, the investigation of alternative proposals aimed at overcoming, if possible, the limits of both approaches.

6. MLS Properties in the AF and SF Approaches

In this section we provide some useful terminology and notation for the instantiation of MLS properties in the AF and SF approaches and discuss the relationships of our approach with consistency and rationality postulates.

6.1 Instantiating Properties

To support the analysis of particular argumentation formalisms as AF or SF MLSs in Section 7, it is useful to introduce a suitable terminology concerning the instances of the properties introduced in Section 3.3 in the context of the AF and SF approaches.

Before doing this, we recall that a generic AF MLS consists of the composition $\text{mgen}_{\text{ASJ}} \circ \text{mgen}_{\text{AJ}}$ where $\text{mgen}_{\text{AJ}} = \text{mgen}[\text{ID}, \text{syn}_{\text{AJ}}]$ and $\text{mgen}_{\text{ASJ}} = \text{mgen}[\text{SC}, \text{syn}_{\text{ASJ}}]$. Since the n -influences ID and SC are fixed, the properties of a specific AF MLS depend on the adopted synthesizers syn_{AJ} and syn_{ASJ} .

Similarly, a generic SF MLS consists of the composition $\text{mgen}_{\text{SSJ}} \circ \text{pgen}_{\text{SA}}$ where $\text{pgen}_{\text{SA}} = \text{pgen}[\text{SC}, \text{syn}_{\text{SA}}]$ and $\text{mgen}_{\text{SSJ}} = \text{mgen}[\text{IC}, \text{syn}_{\text{SSJ}}]$. Also in this case, SC and IC being fixed, the properties of a specific SF MLS depend on the adopted synthesizers syn_{SA} and syn_{SSJ} .

First, we introduce a terminology for coverage at the different stages of AF and SF MLSs.

Definition 6.1 (AF Coverage). *An AF MLS $\text{mgen}_{\text{ASJ}} \circ \text{mgen}_{\text{AJ}}$ is said to satisfy:*

- coverage of argument justification *iff* mgen_{AJ} *satisfies coverage, and*
- coverage of statement justification *iff* mgen_{ASJ} *satisfies coverage.*

Definition 6.2 (SF Coverage). *A SF MLS $\text{mgen}_{\text{SSJ}} \circ \text{pgen}_{\text{SA}}$ is said to satisfy:*

- coverage of statement acceptance *iff* pgen_{SA} *satisfies coverage, and*
- coverage of statement justification *iff* mgen_{SSJ} *satisfies coverage.*

Clearly, in actual instances, the four coverage properties orderly introduced in the definitions 6.1 and 6.2 depend essentially on the synthesizers syn_{AJ} , syn_{ASJ} , syn_{SA} , and syn_{SSJ} respectively.

Second, concerning indistinguishability, we introduce a terminology for the 2-synthesizers syn_{ASJ} , syn_{SA} and syn_{SSJ} , which share the property that their first argument is obtained through a projection based on support while their second argument is obtained through a projection based on contrariness.

Notation 6.1 (Support and contrary-based indistinguishability). *Given $syn \in \{syn_{ASJ}, syn_{SA}, syn_{SSJ}\}$ and $\Lambda^a, \Lambda^b \in \text{dom}^1(syn)$, we say that Λ^a and Λ^b are support-based strongly (weakly) syn -indistinguishable if they are strongly (weakly) syn -1-indistinguishable. Similarly, given $\Lambda^a, \Lambda^b \in \text{dom}^2(syn)$ we say that Λ^a and Λ^b are contrary-based strongly (weakly) syn -indistinguishable if they are strongly (weakly) syn -2-indistinguishable.*

We also aim at providing a direct formal counterpart to the fact that, while there is some notion of contrariness between statements in every argumentation formalism, this notion does not always play an explicit role in the definition of acceptance and justification labellings of statements. To this purpose we introduce a notion of contrary-(in)sensitivity for the relevant operators.

Notation 6.2 (Contrary-(in)sensitivity). *Let $syn \in \{syn_{ASJ}, syn_{SA}, syn_{SSJ}\}$. We say that syn is contrary-insensitive iff it is 2-insensitive, contrary-sensitive otherwise.*

Eventually, for the 1-synthesizer syn_{AJ} we will directly speak of syn_{AJ} -indistinguishability instead of syn_{AJ} -1-indistinguishability, given that specifying -1- is unnecessary for such a synthesizer.

6.2 On Consistency and Rationality Postulates

In the discussion of the various formalisms in Section 7 we will take into account a basic consistency property which is satisfied by all of them at the level of argument acceptance labellings: stated in general terms, arguments whose conclusions are contraries cannot be accepted together in a consistent assessment.

This property is related to the notion of *direct consistency*, one of the *rationality postulates* introduced by Caminada and Amgoud (2007) for the family of rule-based argumentation systems where a distinction between strict and defeasible rules is encompassed and an extension-based mechanism for argument acceptance is used. Note that while direct consistency has effects on how formalisms satisfying it are modelled as MLSs, this property is not affected (and cannot be affected) by the mechanism of derivation of statement labellings from argument labellings, since it simply lies at a different level, i.e. it holds or not at the argument acceptance level independently of what processing is done later. Other properties considered by Caminada and Amgoud (2007) are *closure*, namely the fact that if a set of statements Φ is accepted according to an extension then also the statements derivable from Φ using strict rules should be accepted according to the same extension, and *indirect consistency*, namely the fact that if a set of statements Φ is accepted according to an extension, then the statements derivable from Φ using strict rules should not include contraries.

Our approach is completely agnostic with respect to the argument production mechanism (in particular we do not assume the existence of rules nor the distinction between strict and defeasible rules) and the argument acceptance mechanism (in particular we do not assume the existence of an underlying set of extensions). Hence, strictly speaking, rationality postulates as introduced by Caminada and Amgoud (2007) are outside the scope of the MLS formalism and no direct relationships

can be drawn with them. Designing novel generalised notions of rationality postulates, suitable for the wider context of MLSs and possibly tailored for the AF and SF approaches is beyond the scope of the present paper and appears to be an interesting direction of future work.

7. Argumentation Formalisms as Multi-labelling Systems

In this section, several argument-based formalisms are interpreted and analysed in the model of multi-labelling systems. For each formalism we first recall its essential definitions and provide a corresponding formulation as a multi-labelling system, discussing its membership to the AF or the SF approach. Then we analyse and comment the status of the formalism with respect to the properties introduced in Sections 3.3 and 6. Finally, we provide some instantiated elements of comparison by describing the application to our leading example.

Notation 7.1. *To ease readability in the rest of the paper, we use short names for synthesizers in the context of specific formalisms. For instance, the synthesizer concerning argument justification in the context of $ASPIC^+$ will be denoted as AJ^{A^+} rather than $syn_{AJ}^{A^+}$ and similarly for other synthesizers.*

7.1 $ASPIC^+$

$ASPIC^+$ (denoted as A^+ for short) is a rich rule-based argumentation formalism (Prakken, 2010; Modgil & Prakken, 2013, 2014). Recalling in detail the full formal apparatus of $ASPIC^+$ is beyond the scope of the present paper and we limit ourselves to quickly review the basic elements which are relevant to interpret $ASPIC^+$ as a MLS. The reader is referred to the cited references for a complete view of the formalism.

First of all, $ASPIC^+$ assumes the existence of a generic language \mathcal{L} equipped with a generic contrariness relation (see Definition 2 of Modgil & Prakken, 2013) which directly corresponds to our Definition 4.2. $ASPIC^+$ arguments are built by using (strict and defeasible) rules and each argument has a conclusion, which is an element of \mathcal{L} . Abstracting away the underlying construction mechanism, the set of arguments produced by an $ASPIC^+$ argumentation system (see Definition 5 of Modgil & Prakken, 2013) can thus be regarded as an argument-conclusion structure according to our Definition 4.3, and for each set of statements Φ it is possible to identify the set of supporting arguments according to our Definition 4.4.

$ASPIC^+$ arguments may attack and defeat each other (see Definitions 8 and 9 of Modgil & Prakken, 2013), and on the basis of the attack/defeat relation an abstract argumentation framework (Dung, 1995) can be built (see Definition 11 of Modgil & Prakken, 2013).

Argument acceptance is then based on the use of a semantics for abstract argumentation frameworks (Dung, 1995; Baroni, Caminada, & Giacomin, 2011). Accordingly, it is possible to refer to the labelling-based version of abstract argumentation semantics (Caminada & Gabbay, 2009; Baroni et al., 2011), where a set of three argument acceptance labels is adopted, namely $\Lambda_{AA}^{IOU} = \{IN, OUT, UN\}$ (see Appendix A). We can note that the Λ_{AA}^{IOU} -based argument acceptance labellings based on the various abstract argumentation semantics proposed in the literature are all total. It has to be remarked however that *stable* semantics does not admit the UN label and that it fails to produce any labelling in some cases.

Accordingly, in the MLS model, we can identify the first stage as the pair $\Xi_0 = \langle \mathcal{A}, \Lambda_{AA}^{IOU} \rangle$, where \mathcal{A} is the set of arguments produced by the considered $ASPIC^+$ argumentation system. On this basis, we proceed to discuss the interpretation of $ASPIC^+$ as an AF or SF MLS in the next subsections.

7.1.1 ASPIC⁺ AS AN AF MLS

On the basis of argument acceptance labellings, ASPIC⁺ provides a definition of argument justification, hence it can be naturally regarded as belonging to the AF approach. In particular it adopts the traditional notion of skeptical and credulous justification (see Definition 1 of Modgil & Prakken, 2013) which states that an argument is skeptically justified (denoted SKJ) if it is labelled IN in all labellings prescribed by the adopted semantics, while it is credulously justified (denoted CRJ) if it is labelled IN in some labellings. We slightly adapt the original definition to keep the two notions disjoint.

Definition 7.1. *Given a set of arguments \mathcal{A} and a set \mathfrak{L}_{AA} of $\Lambda_{AA}^{\text{IOU}}$ -labellings of \mathcal{A} , for every argument $A \in \mathcal{A}$,*

- *A is skeptically justified iff $\forall L_{AA} \in \mathfrak{L}_{AA} L_{AA}(A) = \text{IN}$;*
- *A is credulously justified iff $\exists L_{AA} \in \mathfrak{L}_{AA} : L_{AA}(A) = \text{IN}$ and $\exists L_{AA} \in \mathfrak{L}_{AA} : L_{AA}(A) \neq \text{IN}$.*

It is easy to see that, letting $\Lambda_{AJ}^{\text{A}^+} = \{\text{SKJ}, \text{CRJ}\}$, Definition 7.1 corresponds to a stage $\Xi_1^{\text{AF}} = \langle \mathcal{A}, \Lambda_{AJ}^{\text{A}^+}, \text{mgen}_{AJ} \rangle$ where $\text{mgen}_{AJ} = \text{mgen}[\text{ID}, \text{AJ}^{\text{A}^+}]$ and the synthesizer $\text{AJ}^{\text{A}^+} : \text{pow}(\Lambda_{AA}^{\text{IOU}}) \rightarrow \Lambda_{AJ}^{\text{A}^+}$ is defined, for any $S \subseteq \Lambda_{AA}^{\text{IOU}}$, as follows

- $\text{AJ}^{\text{A}^+}(S) = \text{SKJ}$ iff $S = \{\text{IN}\}$;
- $\text{AJ}^{\text{A}^+}(S) = \text{CRJ}$ iff $S \supsetneq \{\text{IN}\}$.

The correspondence can be formally stated with the following proposition.

Proposition 7.1. *Given a set of arguments \mathcal{A} and a set \mathfrak{L}_{AA} of $\Lambda_{AA}^{\text{IOU}}$ -labellings of \mathcal{A} , the argument justification labelling $L_{AJ}^{\text{A}^+}$ prescribed by ASPIC⁺ according to Definition 7.1 is such that $\{L_{AJ}^{\text{A}^+}\} = \text{mgen}_{AJ}(\mathfrak{L}_{AA})$.*

Proof. For the sake of conciseness let $\text{mgen}_{AJ}(\mathfrak{L}_{AA}) = \{L'_{AJ}\}$, we need to show that for every argument A , $L_{AJ}^{\text{A}^+}(A) = L'_{AJ}(A)$. According to Definition 7.1, given an argument A , $L_{AJ}^{\text{A}^+}(A) = \text{SKJ}$ iff it is labelled IN in all the labellings in \mathfrak{L}_{AA} (i.e. those prescribed by the chosen argumentation semantics), which occurs iff $\text{prj}[\text{id}](\mathfrak{L}_{AA}, A) = \{\text{IN}\}$ leading to $\text{AJ}^{\text{A}^+}(\text{prj}[\text{id}](\mathfrak{L}_{AA}, A)) = \text{SKJ}$ and then $L'_{AJ}(A) = \text{SKJ}$. Similarly, $L_{AJ}^{\text{A}^+}(A) = \text{CRJ}$ iff it is labelled IN in some but not all the labellings in \mathfrak{L}_{AA} , which occurs iff $\text{prj}[\text{id}](\mathfrak{L}_{AA}, A) \supsetneq \{\text{IN}\}$ leading to $\text{AJ}^{\text{A}^+}(\text{prj}[\text{id}](\mathfrak{L}_{AA}, A)) = \text{CRJ}$ and then $L'_{AJ}(A) = \text{CRJ}$. \square

In ASPIC⁺, statements inherit directly the justification status of the ‘best justified’ argument supporting them (see Definition 15 of Modgil & Prakken, 2013): a statement is skeptically justified if and only if it is the conclusion of a skeptically justified argument, while it is credulously justified if and only if it is the conclusion of a credulously justified argument. Also in this case we introduce a little modification to keep the two notions disjoint.

Definition 7.2. *Given a language \mathcal{L} , a set of arguments \mathcal{A} with conclusions in \mathcal{L} , and a $\Lambda_{AJ}^{\text{A}^+}$ -justification labelling $L_{AJ}^{\text{A}^+}$ of \mathcal{A} , for every statement $\varphi \in \mathcal{L}$,*

- *φ is skeptically justified iff $\exists A \in \text{sup}(\{\varphi\})$ such that A is skeptically justified (i.e. $L_{AJ}^{\text{A}^+}(A) = \text{SKJ}$);*
- *φ is credulously justified iff $\exists A \in \text{sup}(\{\varphi\})$ such that A is credulously justified (i.e. $L_{AJ}^{\text{A}^+}(A) = \text{CRJ}$) and $\nexists A \in \text{sup}(\{\varphi\})$ such that A is skeptically justified;*

Letting $\Lambda_{ASJ}^{A^+} = \{\text{skj}, \text{crj}\}$, Definition 7.2 corresponds to a stage $\Xi_2^{AF} = \langle \mathcal{L}, \Lambda_{ASJ}^{A^+}, \text{mgen}_{ASJ} \rangle$ where $\text{mgen}_{ASJ} = \text{mgen}[\text{SC}, \text{ASJ}^{A^+}]$ and $\text{ASJ}^{A^+} : \text{pow}(\Lambda_{AJ}^{A^+}) \times \text{pow}(\Lambda_{AJ}^{A^+}) \rightarrow \Lambda_{ASJ}^{A^+}$ is defined, for any $S, U \in \text{pow}(\Lambda_{AJ}^{A^+})$, as follows

- $\text{ASJ}^{A^+}(S, U) = \text{skj}$ iff $\text{SKJ} \in S$;
- $\text{ASJ}^{A^+}(S, U) = \text{crj}$ iff $\text{CRJ} \in S$ and $\text{SKJ} \notin S$.

The correspondence is formally stated by the following proposition.

Proposition 7.2. *Given a language \mathcal{L} , a set of arguments \mathcal{A} with conclusions in \mathcal{L} , and a $\Lambda_{AJ}^{A^+}$ -justification labelling $L_{AJ}^{A^+}$ of \mathcal{A} , the statement justification labelling $L_{ASJ}^{A^+}$ prescribed by ASPIC^+ according to Definition 7.2 is such that $\{L_{ASJ}^{A^+}\} = \text{mgen}_{ASJ}(\{L_{AJ}^{A^+}\})$.*

Proof. For the sake of conciseness let $\text{mgen}_{ASJ}(\{L_{AJ}^{A^+}\}) = \{L'_{ASJ}\}$, we need to show that for every statement φ $L_{ASJ}^{A^+}(\varphi) = L'_{ASJ}(\varphi)$. According to Definition 7.2, $L_{ASJ}^{A^+}(\varphi) = \text{skj}$ iff $\exists A \in \text{sup}(\{\varphi\})$ such that $L_{AJ}^{A^+}(A) = \text{SKJ}$ iff $\text{SKJ} \in \text{prj}[\text{supp}](\{L_{AJ}^{A^+}\}, \varphi)$ leading to $\text{ASJ}^{A^+}(\text{prj}[\text{supp}](\{L_{AJ}^{A^+}\}, \varphi), \text{prj}[\text{ctr}](\{L_{AJ}^{A^+}\}, \varphi)) = \text{skj}$ and then $L'_{ASJ}(\varphi) = \text{skj}$. Similarly, $L_{ASJ}^{A^+}(\varphi) = \text{crj}$ iff $\exists A \in \text{sup}(\{\varphi\})$ such that $L_{AJ}^{A^+}(A) = \text{CRJ}$ and $\nexists A \in \text{sup}(\{\varphi\})$ such that $L_{AJ}^{A^+}(A) = \text{SKJ}$ iff $\text{CRJ} \in \text{prj}[\text{supp}](\{L_{AJ}^{A^+}\}, \varphi)$ and $\text{SKJ} \notin \text{prj}[\text{supp}](\{L_{AJ}^{A^+}\}, \varphi)$ leading to $\text{ASJ}^{A^+}(\text{prj}[\text{supp}](\{L_{AJ}^{A^+}\}, \varphi), \text{prj}[\text{ctr}](\{L_{AJ}^{A^+}\}, \varphi)) = \text{crj}$ and then $L'_{ASJ}(\varphi) = \text{crj}$. \square

Having reconstructed ASPIC^+ as an AF MLS, we can proceed to analyse its properties.

Coverage. It can be first observed that both AJ^{A^+} and ASJ^{A^+} do not satisfy coverage. AJ^{A^+} does not cover the cases where $\text{IN} \notin \text{prj}[\text{id}](\mathcal{L}_{AA}, A)$, i.e. the justification status of arguments which do not appear in any extension is left undefined, while ASJ^{A^+} does not cover the cases where $\text{prj}[\text{supp}](\{L_{AJ}^{A^+}\}, \varphi) \cap \{\text{SKJ}, \text{CRJ}\} = \emptyset$, i.e. the justification status is left undefined in all the various cases where a statement is not supported by any justified argument. These observations lead directly to the following proposition

Proposition 7.3. *The AF MLS $\text{mgen}[\text{SC}, \text{ASJ}^{A^+}] \circ \text{mgen}[\text{ID}, \text{AJ}^{A^+}]$ corresponding to the ASPIC^+ formalism does not satisfy coverage of argument justification nor of statement justification.*

The lack of the coverage property can be explained by the emphasis on acceptance in the definitions provided for ASPIC^+ . It can be observed however that this limitation is, in a sense, more formal than substantial, since, both for arguments and statements, it is easy to recover a full coverage by introducing a third status corresponding to the absence of justification, while preserving the same meaning. This is achieved by the following definitions.

Definition 7.3. *Letting $\Lambda_{AJ}^{A^+ \cdot \text{fc}} = \{\text{SKJ}, \text{CRJ}, \text{NOJ}\}$, the 1-synthesizer $\text{AJ}^{A^+ \cdot \text{fc}}$ from $\Lambda_{AA}^{\text{IOU}}$ to $\Lambda_{AJ}^{A^+ \cdot \text{fc}}$ is defined for $S \in \text{pow}(\Lambda_{AA}^{\text{IOU}})$ as*

- $\text{AJ}^{A^+ \cdot \text{fc}}(S) = \text{SKJ}$ iff $S = \{\text{IN}\}$;
- $\text{AJ}^{A^+ \cdot \text{fc}}(S) = \text{CRJ}$ iff $S \supseteq \{\text{IN}\}$;
- $\text{AJ}^{A^+ \cdot \text{fc}}(S) = \text{NOJ}$ otherwise.

Definition 7.4. *Letting $\Lambda_{ASJ}^{A^+ \cdot \text{fc}} = \{\text{skj}, \text{crj}, \text{noj}\}$, the 2-synthesizer $\text{ASJ}^{A^+ \cdot \text{fc}}$ from $\Lambda_{AJ}^{A^+ \cdot \text{fc}}$ to $\Lambda_{ASJ}^{A^+ \cdot \text{fc}}$ is defined for $S, U \in \text{pow}(\Lambda_{AJ}^{A^+ \cdot \text{fc}})$ as*

- $\text{ASJ}^{A^+ \cdot \text{fc}}(S, U) = \text{skj}$ iff $\text{SKJ} \in S$;

- $ASJ^{A^+.fc}(S, U) = crj$ iff $CRJ \in S$ and $SKJ \notin S$;
- $ASJ^{A^+.fc}(S, U) = noj$ otherwise.

Table 1 illustrates the statement justification labelling prescribed by Definition 7.4 depending on projections $prj[supp](\{L_{AJ}^{A^+}, \varphi\})$ and $prj[ctr](\{L_{AJ}^{A^+}, \varphi\})$. In this table and similar tables in other sections, grey cells with n.a. indicate impossible cases (i.e. pairs not belonging to the domain $\text{dom}(ASJ^{A^+.fc})$), to be commented in more detail later. For the sake of conciseness the row and column indicated with ‘ $\supseteq \{SKJ\}$ ’ synthesise all the equal rows and columns corresponding to the sets $\{SKJ\}$, $\{SKJ, NOJ\}$, $\{SKJ, CRJ\}$ and $\{SKJ, CRJ, NOJ\}$.

Table 1: Justification status of a statement for $ASPIC^+$.

$prj[ctr] \setminus prj[supp]$	\emptyset	$\{NOJ\}$	$\{CRJ\}$	$\{NOJ, CRJ\}$	$\supseteq \{SKJ\}$
\emptyset	noj	noj	crj	crj	skj
$\{NOJ\}$	noj	noj	crj	crj	skj
$\{CRJ\}$	noj	noj	crj	crj	n.a.
$\{NOJ, CRJ\}$	noj	noj	crj	crj	n.a.
$\supseteq \{SKJ\}$	noj	noj	n.a.	n.a.	n.a.

It is then immediate to see that the AF MLS $\text{mgen}[SC, ASJ^{A^+.fc}] \circ \text{mgen}[ID, AJ^{A^+.fc}]$ satisfies both coverage of argument justification and of statement justification and still satisfies the correspondences with $ASPIC^+$ stated in Proposition 7.1 for arguments labelled SKJ or CRJ and in Proposition 7.2 for statements labelled skj or crj.

Distinguishability. To analyse the properties of $ASPIC^+$ with respect to the (in)distinguishability of sets of labels, we treat the versions of the operators with and without full coverage (this difference has no actual effects for the purposes of the present subsection).

Let us start the analysis with the argument justification stage and the relevant 1-synthesizers AJ^{A^+} and $AJ^{A^+.fc}$. First, it can be noted that for both operators the actual domain of definition depends on the adopted semantics and that AJ^{A^+} is defined only for sets of labels including IN. This gives rise to the following cases.

1. In the case of a single-status semantics (like grounded or ideal semantics), each argument gets one and only one label, thus
 - $\text{dom}(AJ^{A^+}) = \{\{IN\}\}$ and
 - $\text{dom}(AJ^{A^+.fc}) = \{\{IN\}, \{OUT\}, \{UN\}\}$.
2. In the case of a multiple-status semantics which always prescribes at least a labelling (i.e. all the multiple-status semantics considered in this paper, except stable semantics), it has been proved (Baroni, Giacomin, & Guida, 2004) that all combinations of labels (but the empty one) are possible, hence
 - $\text{dom}(AJ^{A^+}) = \{\Lambda \subseteq \Lambda_{AA}^{IOU} \mid IN \in \Lambda\}$ and
 - $\text{dom}(AJ^{A^+.fc}) = \text{pow}(\Lambda_{AA}^{IOU}) \setminus \emptyset$.
3. In the case of stable semantics, the label UN is not possible but it may happen that no labelling is prescribed, hence
 - $\text{dom}(AJ^{A^+}) = \{\{IN\}, \{IN, OUT\}\}$ and
 - $\text{dom}(AJ^{A^+.fc}) = \text{pow}(\{IN, OUT\})$.

In all cases, it can be noted that the set of labels $\{\text{OUT}\}$ and $\{\text{UN}\}$ are strongly indistinguishable with respect to both AJ^{A^+} and $\text{AJ}^{\text{A}^+.\text{fc}}$.

Proposition 7.4. *It holds that $\{\text{OUT}\} \equiv [1, \text{AJ}^{\text{A}^+}]\{\text{UN}\}$ and $\{\text{OUT}\} \equiv [1, \text{AJ}^{\text{A}^+.\text{fc}}]\{\text{UN}\}$*

Proof. Consider first AJ^{A^+} . With reference to Definition 3.12, let $\Lambda_1 = \{\text{IN}, \text{OUT}, \text{UN}\}$: we have to prove that $\forall \Lambda' \subseteq \Lambda_1$ such that $\Lambda' \cup \{\text{OUT}\}, \Lambda' \cup \{\text{UN}\} \in \text{dom}(\text{AJ}^{\text{A}^+})$, it holds that $\text{AJ}^{\text{A}^+}(\Lambda' \cup \{\text{OUT}\}) = \text{AJ}^{\text{A}^+}(\Lambda' \cup \{\text{UN}\})$. The following cases hold:

- $\text{IN} \in \Lambda'$: then, if $(\Lambda' \cup \{\text{OUT}\}), (\Lambda' \cup \{\text{UN}\}) \in \text{dom}(\text{AJ}^{\text{A}^+})$, we get $\text{AJ}^{\text{A}^+}(\Lambda' \cup \{\text{OUT}\}) = \text{AJ}^{\text{A}^+}(\Lambda' \cup \{\text{UN}\}) = \text{CRJ}$;
- $\text{IN} \notin \Lambda'$ then $(\Lambda' \cup \{\text{OUT}\}), (\Lambda' \cup \{\text{UN}\}) \notin \text{dom}(\text{AJ}^{\text{A}^+})$.

Therefore, it holds that $\{\text{OUT}\} \equiv [1, \text{AJ}^{\text{A}^+}]\{\text{UN}\}$.

Consider now $\text{AJ}^{\text{A}^+.\text{fc}}$. Referring again to Definition 3.12 and letting $\Lambda_1 = \{\text{IN}, \text{OUT}, \text{UN}\}$, we have to prove similarly that $\forall \Lambda' \subseteq \Lambda_1$ such that $\Lambda' \cup \{\text{OUT}\}, \Lambda' \cup \{\text{UN}\} \in \text{dom}(\text{AJ}^{\text{A}^+.\text{fc}})$, it holds that $\text{AJ}^{\text{A}^+.\text{fc}}(\Lambda' \cup \{\text{OUT}\}) = \text{AJ}^{\text{A}^+.\text{fc}}(\Lambda' \cup \{\text{UN}\})$. The following cases hold:

- $\text{IN} \in \Lambda'$ then if $(\Lambda' \cup \{\text{OUT}\}), (\Lambda' \cup \{\text{UN}\}) \in \text{dom}(\text{AJ}^{\text{A}^+.\text{fc}})$ then $\text{AJ}^{\text{A}^+.\text{fc}}(\Lambda' \cup \{\text{OUT}\}) = \text{AJ}^{\text{A}^+.\text{fc}}(\Lambda' \cup \{\text{UN}\}) = \text{CRJ}$;
- $\text{IN} \notin \Lambda'$ then if $(\Lambda' \cup \{\text{OUT}\}), (\Lambda' \cup \{\text{UN}\}) \in \text{dom}(\text{AJ}^{\text{A}^+.\text{fc}})$ then $\text{AJ}^{\text{A}^+.\text{fc}}(\{\text{OUT}\}) = \text{AJ}^{\text{A}^+.\text{fc}}(\{\text{UN}\}) = \text{NOJ}$.

Therefore, it holds that $\{\text{OUT}\} \equiv [1, \text{AJ}^{\text{A}^+.\text{fc}}]\{\text{UN}\}$. \square

Corollary 7.1. *The sets of labels $\{\text{OUT}\}$, $\{\text{UN}\}$ and $\{\text{OUT}, \text{UN}\}$ are pairwise strongly AJ^{A^+} -indistinguishable and strongly $\text{AJ}^{\text{A}^+.\text{fc}}$ -indistinguishable.*

Proof. Follows directly from Proposition 7.19 and Proposition 3.2. \square

Turning to ASJ^{A^+} and $\text{ASJ}^{\text{A}^+.\text{fc}}$, it has to be observed that ASPIC^+ satisfies some basic rationality constraints. In particular, two arguments with contrary conclusions cannot belong to the same extension. It follows that for every statement φ if $\text{SKJ} \in \text{prj}[\text{supp}](\{\text{L}_{\text{AJ}}^{\text{A}^+}\}, \varphi)$ then $\{\text{SKJ}, \text{CRJ}\} \cap \text{prj}[\text{cntr}](\{\text{L}_{\text{AJ}}^{\text{A}^+}\}, \varphi) = \emptyset$ and vice versa if $\text{SKJ} \in \text{prj}[\text{cntr}](\{\text{L}_{\text{AJ}}^{\text{A}^+}\}, \varphi)$ then $\{\text{SKJ}, \text{CRJ}\} \cap \text{prj}[\text{supp}](\{\text{L}_{\text{AJ}}^{\text{A}^+}\}, \varphi) = \emptyset$. This leads to identify the following domains of definition:

- $\text{dom}(\text{ASJ}^{\text{A}^+}) = \text{pow}(\{\text{SKJ}, \text{CRJ}\}) \times \{\emptyset\} \cup \{\emptyset\} \times \text{pow}(\{\text{SKJ}, \text{CRJ}\}) \cup \{(\{\text{CRJ}\}, \{\text{CRJ}\})\}$;
- $\text{dom}(\text{ASJ}^{\text{A}^+.\text{fc}}) = \text{pow}(\{\text{SKJ}, \text{CRJ}, \text{NOJ}\}) \times \text{pow}(\{\text{NOJ}\}) \cup \text{pow}(\{\text{NOJ}\}) \times \text{pow}(\{\text{SKJ}, \text{CRJ}, \text{NOJ}\}) \cup \text{pow}(\{\text{CRJ}, \text{NOJ}\}) \times \text{pow}(\{\text{CRJ}, \text{NOJ}\})$.

In all cases, the empty set and $\{\text{NOJ}\}$ are support-based strongly $\text{ASJ}^{\text{A}^+.\text{fc}}$ -indistinguishable.

Proposition 7.5. *The sets of labels \emptyset and $\{\text{NOJ}\}$ are support-based strongly $\text{ASJ}^{\text{A}^+.\text{fc}}$ -indistinguishable.*

Proof. With reference to Notation 6.1 and Definition 3.12, let $\Lambda_1 = \{\text{SKJ}, \text{CRJ}, \text{NOJ}\}$: we have to prove that $\forall \Lambda', \Lambda'' \subseteq \Lambda_1$ such that $(\Lambda', \Lambda''), (\Lambda' \cup \{\text{NOJ}\}, \Lambda'') \in \text{dom}(\text{ASJ}^{\text{A}^+.\text{fc}})$ it holds that $\text{ASJ}^{\text{A}^+.\text{fc}}(\Lambda', \Lambda'') = \text{ASJ}^{\text{A}^+.\text{fc}}(\Lambda' \cup \{\text{NOJ}\}, \Lambda'')$. The following cases hold:

- $\text{SKJ} \in \Lambda'$ then for every $\Lambda'' \in \text{pow}(\{\text{NOJ}\})$ it holds that $\text{ASJ}^{\text{A}^+.\text{fc}}(\Lambda', \Lambda'') = \text{ASJ}^{\text{A}^+.\text{fc}}(\Lambda' \cup \{\text{NOJ}\}, \Lambda'') = \text{skj}$;
- $\text{SKJ} \notin \Lambda'$ and $\text{CRJ} \in \Lambda'$, then for every $\Lambda'' \in \text{pow}(\{\text{CRJ}, \text{NOJ}\})$ it holds that $\text{ASJ}^{\text{A}^+.\text{fc}}(\Lambda', \Lambda'') = \text{ASJ}^{\text{A}^+.\text{fc}}(\Lambda' \cup \{\text{NOJ}\}, \Lambda'') = \text{crj}$;

- $\Lambda' \subseteq \{\text{NOJ}\}$ then for every $\Lambda'' \in \text{pow}(\{\text{SKJ}, \text{CRJ}, \text{NOJ}\})$ it holds that $\text{ASJ}^{\text{A}^+ \cdot \text{fc}}(\Lambda', \Lambda'') = \text{ASJ}^{\text{A}^+ \cdot \text{fc}}(\Lambda' \cup \{\text{NOJ}\}, \Lambda'') = \text{noj}$.

Therefore, the sets of labels \emptyset and $\{\text{NOJ}\}$ are support-based strongly $\text{ASJ}^{\text{A}^+ \cdot \text{fc}}$ -indistinguishable. \square

Moreover, both ASJ^{A^+} and $\text{ASJ}^{\text{A}^+ \cdot \text{fc}}$ are contrary-insensitive, i.e. all sets of labels are strongly contrary-based indistinguishable with respect to these operators. This follows from the fact that ASJ^{A^+} and $\text{ASJ}^{\text{A}^+ \cdot \text{fc}}$ do not actually use their second parameter, i.e. $\text{prj}[\text{cntr}](\{\text{L}_{\text{AJ}}^{\text{A}^+}\}, \varphi)$ has no effects in the definition of $\text{L}_{\text{SJ}}^{\text{A}^+}(\varphi)$ and $\text{L}_{\text{SJ}}^{\text{A}^+ \cdot \text{fc}}(\varphi)$.

Corollary 7.2. *ASJ^{A^+} and $\text{ASJ}^{\text{A}^+ \cdot \text{fc}}$ are contrary-insensitive.*

Nevertheless, the fact that ASJ^{A^+} and $\text{ASJ}^{\text{A}^+ \cdot \text{fc}}$ are contrary-insensitive, which may appear as a technical limitation, can be partially compensated by other considerations. First of all, the constraints on the domain reveal a sort of (partial and implicit) contrary-sensitiveness, at least in the case $\text{SKJ} \in \text{prj}[\text{cntr}](\{\text{L}_{\text{AJ}}^{\text{A}^+}\}, \varphi)$. Moreover, the set of rules used in a specific instance of AS-PIC^+ might be defined in such a way as to induce further constraints on the actual possible values of the pair $(\text{prj}[\text{supp}](\{\text{L}_{\text{AJ}}^{\text{A}^+}\}, \varphi), \text{prj}[\text{cntr}](\{\text{L}_{\text{AJ}}^{\text{A}^+}\}, \varphi))$, thus somehow capturing some form of contrary-sensitiveness (again partial and implicit).

Example 7.1 (continues Example 1.1). *Referring to the acceptance evaluation given in Example 4.3, which corresponds to the set of acceptance labellings prescribed by Dung's complete semantics, arguments are labelled according to the labelling $\text{L}_{\text{AJ}}^{\text{A}^+ \cdot \text{fc}}$ as follows:*

	A1	A2	A3	A4
$\text{L}_{\text{AJ}}^{\text{A}^+ \cdot \text{fc}}(\cdot)$	CRJ	CRJ	NOJ	SKJ

On this basis, statements obtain the following justification statuses:

	s1	\neg s1	s2	\neg s2	s3
$\text{L}_{\text{ASJ}}^{\text{A}^+ \cdot \text{fc}}(\cdot)$	crj	crj	noj	skj	noj

As a first comment we can remark that the inability to distinguish the status of s2 and s3 is related to the fact that $\{\text{NOJ}\}$ and \emptyset are support-based strongly $\text{ASJ}^{\text{A}^+ \cdot \text{fc}}$ -indistinguishable, thus a statement (s3) which is the conclusion of no argument is equated to one (s2) which is the conclusion of a rejected argument. It can be also be observed that the same outcome would be obtained using preferred, stable and semi-stable semantics, which would prescribe $\mathfrak{L}_{\text{AA}} = \{\text{L}_{\text{AA}}^1, \text{L}_{\text{AA}}^2\}$. The fact that the presence or absence of L_{AA}^3 in \mathfrak{L}_{AA} does not make any difference is related to the fact that $\{\text{OUT}\}$ and $\{\text{UN}\}$ are strongly indistinguishable with respect to $\text{AJ}^{\text{A}^+ \cdot \text{fc}}$. Hence the addition of L_{AA}^3 has no effect on the justification labels of arguments A1 and A2 (labelled UN by L_{AA}^3), given that they are labelled OUT respectively by L_{AA}^2 and L_{AA}^1 .

The outcome would instead be different with the adoption of grounded, ideal, or eager semantics which would prescribe $\mathfrak{L}_{\text{AA}} = \{\text{L}_{\text{AA}}^3\}$. In this case we would get labelling $\text{L}_{\text{AJ}}^{\text{A}^+ \cdot \text{fc}}$ as follows:

	A1	A2	A3	A4
$\text{L}_{\text{AJ}}^{\text{A}^+ \cdot \text{fc}}(\cdot)$	NOJ	NOJ	NOJ	SKJ

and

	s1	\neg s1	s2	\neg s2	s3
$\text{L}_{\text{ASJ}}^{\text{A}^+ \cdot \text{fc}}(\cdot)$	noj	noj	noj	skj	noj

In this case the inability to distinguish also the status of $s1$ and $\neg s1$ with respect to the one of $s2$ is again rooted in the fact that $\{\text{OUT}\}$ and $\{\text{UN}\}$ are strongly (hence weakly) indistinguishable with respect to $\text{AJ}^{\text{A}^+ \cdot \text{fc}}$ leading to the same justification status for $A1$, $A2$, and $A3$. \square

7.1.2 ASPIC^+ AS A SF MLS: A NEGATIVE RESULT

With the above analysis and in particular with propositions 7.1 and 7.2, we have built an AF MLS corresponding to ASPIC^+ . It can be shown (using the same line of reasoning presented in the second part of Section 5) that it is impossible to build a SF MLS which, starting from the evaluation $\mathcal{L}_{\text{AA}}(\mathcal{AC})$, produces the same statement justification labelling as ASPIC^+ in all cases. This can be seen by instantiating cases C3 and C4 presented in Section 5 as follows.

C3. The situation of arguments A and B in the case C3 can be obtained, for instance⁷, with a Dung’s argumentation framework consisting of three arguments, A , B , C , where A is unattacked while B and C mutually attack each other. Applying stable semantics to this framework gives rise to two labellings L_{AA}^1 and L_{AA}^2 whose restriction on A and B is as described in Section 5.

C4. The situation of arguments A and B in the case C4 can be obtained with a Dung’s argumentation framework consisting of four arguments, A , B , C , D where A and D mutually attack each other, B and C mutually attack each other, and in addition D attacks C . Again, applying stable semantics to this framework gives rise to three labellings L_{AA}^1 , L_{AA}^2 , L_{AA}^3 whose restriction on A and B is as described in Section 5.

Under the assumption that $\text{sup}(\{\varphi\}) = \{A, B\}$, we get that in case C3, $L_{\text{AJ}}^{\text{A}^+}(A) = \text{SKJ}$, $L_{\text{AJ}}^{\text{A}^+}(B) = \text{CRJ}$, from which $L_{\text{SJ}}^{\text{A}^+}(\varphi) = \text{skj}$. In case C4 we get $L_{\text{AJ}}^{\text{A}^+}(A) = L_{\text{AJ}}^{\text{A}^+}(B) = \text{CRJ}$ from which $L_{\text{ASJ}}^{\text{A}^+}(\varphi) = \text{crj}$. So the justification of the statement φ is different in the two cases, while as shown in Section 5 this difference cannot be obtained in the statement-focused model. This shows that the argument and statement justification mechanisms adopted in ASPIC^+ , as defined by Modgil and Prakken (2014), belong exclusively to the AF camp. Since ASPIC^+ is a generic formalism admitting many instances, note also that this does not show that it is in general impossible to reconstruct actual instances of ASPIC^+ in the SF approach: there can be some instance-specific constraints preventing cases like the ones illustrated above to actually occur.

7.2 Assumption-Based Argumentation

Assumption-based argumentation (denoted as ABA for short) is a rule-based argumentation formalism whose essential features are recalled below, the reader may refer to the tutorial by Toni (2014) for a more extensive treatment. ABA assumes the existence of a generic language \mathcal{L} including a non-empty set of assumptions $\text{As} \subseteq \mathcal{L}$. It is assumed that a contrariness relation $\bar{\cdot} : \text{As} \rightarrow \mathcal{L}$ is given identifying for each assumption α exactly one contrary in \mathcal{L} (denoted as $\bar{\alpha}$), while contraries are not defined for non assumption elements of the language, i.e. for the members of $\mathcal{L} \setminus \text{As}$. Clearly this can be reconstructed as a special case of our contrariness relation in Definition 4.2 by putting $\text{cnt}(\alpha) = \{\bar{\alpha}\}$ for every $\alpha \in \text{As}$ and $\text{cnt}(\varphi) = \emptyset$ for every $\varphi \in \mathcal{L} \setminus \text{As}$.

ABA assumes then the existence of a set of rules which are used to build arguments supporting a conclusion (or claim) in \mathcal{L} . In words, an argument for a claim φ is a deduction for φ , built

7. We omit the underlying rule based-reasoning and admit that these small ad-hoc examples can be felt as somehow unrealistic: the same situation for A and B could be obtained in a realistic rule-based reasoning scenario with a larger number of arguments.

using some rules and a set of assumptions. Thus also for *ABA*, abstracting away the underlying construction mechanism, the set of arguments can be regarded as an argument-conclusion structure according to our Definition 4.3, and for each set of statements Φ it is possible to identify the set of supporting arguments according to Definition 4.4.

In *ABA* rules are assumed to be strict and assumptions are the only weak, hence attackable, parts of an argument. Accordingly a relation of attack between arguments⁸ is defined. In particular, an argument A attacks another argument B if the claim of A is the contrary of one of the assumptions of B . On the basis of the attack relation, an abstract argumentation framework can be built.

Then, as in the case of *ASPIC*⁺, acceptable sets of arguments are identified by applying a semantics for abstract argumentation frameworks. Accordingly, in the MLS model, we can identify the first stage as the pair $\Xi_0 = \langle \mathcal{A}, \Lambda_{AA}^{IOU} \rangle$, where \mathcal{A} is the set of arguments produced by the considered *ABA* system. On this common basis, an articulated situation concerning the subsequent stages must be set down, since both a credulous and a skeptical⁹ stance (Dimopoulos et al., 2002, Section 3) concerning statement justification have been considered in the literature. We discuss next the interpretation of these stances within the MLS model.

7.2.1 *ABA* (CREDULOUS STANCE) AS AN AF MLS

In the credulous stance, see a detailed description in the tutorial by Toni (2014), each acceptance labelling identifies a so-called *winning* set of arguments and an argument is in turn winning if it belongs to a winning set. This can be directly reconstructed in the AF approach: a winning argument corresponds to an argument labelled IN in at least one argument acceptance labelling.

Definition 7.5. *Given a set of arguments \mathcal{A} and a set \mathcal{L}_{AA} of Λ_{AA}^{IOU} -labellings of \mathcal{A} , for every argument $A \in \mathcal{A}$, A is winning iff $\exists L_{AA} \in \mathcal{L}_{AA} : L_{AA}(A) = \text{IN}$.*

Given the set of argument justification labels $\Lambda_{AJ}^{\text{AB-cr}} = \{\text{WIN}\}$, Definition 7.5 corresponds to a stage $\Xi_1^{\text{AF}} = \langle \mathcal{A}, \Lambda_{AJ}^{\text{AB-cr}}, \text{mgen}_{AJ} \rangle$ where $\text{mgen}_{AJ} = \text{mgen}[\text{ID}, \text{AJ}^{\text{AB-cr}}]$ and $\text{AJ}^{\text{AB-cr}} : \text{pow}(\Lambda_{AA}^{IOU}) \rightarrow \Lambda_{AJ}^{\text{AB-cr}}$ is defined, for any $S \subseteq \Lambda_{AA}^{IOU}$, as $\text{AJ}^{\text{AB-cr}}(S) = \text{WIN}$ iff $S \supseteq \{\text{IN}\}$. The correspondence can be formally stated by the following proposition whose easy proof is omitted.

Proposition 7.6. *Given a set of arguments \mathcal{A} and a set \mathcal{L}_{AA} of Λ_{AA}^{IOU} -labellings of \mathcal{A} , the argument justification labelling $L_{AJ}^{\text{AB-cr}}$ prescribed by *ABA* according to Definition 7.5 is such that $\{L_{AJ}^{\text{AB-cr}}\} = \text{mgen}_{AJ}(\mathcal{L}_{AA})$.*

In the credulous stance of *ABA*, statements inherit directly the justification status from arguments: a statement is winning if it is the conclusion of a winning argument (quoting Toni, 2014, Section 5: ‘Given an acceptable/winning set of arguments, a sentence can be deemed acceptable/winning if it is the claim of an argument in the set.’).

Definition 7.6. *Given a language \mathcal{L} , a set of arguments \mathcal{A} with conclusions in \mathcal{L} , and a $\Lambda_{AJ}^{\text{AB-cr}}$ -justification labelling $L_{AJ}^{\text{AB-cr}}$ of \mathcal{A} , for every statement $\varphi \in \mathcal{L}$, φ is winning iff $\exists A \in \text{sup}(\{\varphi\})$ such that A is winning (i.e. $L_{AJ}^{\text{AB-cr}}(A) = \text{WIN}$).*

8. The notion of attack can be formulated in two equivalent ways in *ABA*, involving arguments or sets of assumptions. We recall the form concerning arguments, which more directly corresponds to our formalisation.

9. Unfortunately these terms are overloaded in the literature. In particular the skeptical stance adopted by Dimopoulos, Nebel, and Toni (2002) is significantly different e.g. from the notion of skeptical justification in *ASPIC*⁺. To avoid to introduce a new terminology we are forced to the use of the same term with different meanings in different formalisms.

Letting $\Lambda_{ASJ}^{AB\text{-}cr} = \{\text{win}\}$, Definition 7.6 corresponds to a stage $\Xi_2^{AF} = \langle \mathcal{L}, \Lambda_{ASJ}^{AB\text{-}cr}, \text{mgen}_{ASJ} \rangle$ where $\text{mgen}_{ASJ} = \text{mgen}[\text{SC}, \text{ASJ}^{AB\text{-}cr}]$ and $\text{ASJ}^{AB\text{-}cr} : \text{pow}(\Lambda_{AJ}^{AB\text{-}cr}) \times \text{pow}(\Lambda_{ASJ}^{AB\text{-}cr}) \rightarrow \Lambda_{ASJ}^{AB\text{-}cr}$ is defined, for any $S, U \in \text{pow}(\Lambda_{AJ}^{AB\text{-}cr})$, as $\text{ASJ}^{AB\text{-}cr}(S, U) = \text{win}$ iff $\text{WIN} \in S$. The correspondence is formally stated by the following proposition, whose proof is also obvious.

Proposition 7.7. *Given a language \mathcal{L} , a set of arguments \mathcal{A} with conclusions in \mathcal{L} , and a $\Lambda_{AJ}^{AB\text{-}cr}$ -justification labelling $L_{AJ}^{AB\text{-}cr}$ of \mathcal{A} , the statement justification labelling $L_{ASJ}^{AB\text{-}cr}$ prescribed by ABA according to Definition 7.6 is such that $\{L_{ASJ}^{AB\text{-}cr}\} = \text{mgen}_{ASJ}(\{L_{AJ}^{AB\text{-}cr}\})$.*

Having reconstructed ABA as a (simple) AF MLS, we can proceed to analyse its properties.

Coverage. Both $\text{AJ}^{AB\text{-}cr}$ and $\text{ASJ}^{AB\text{-}cr}$ do not satisfy coverage. Indeed $\text{AJ}^{AB\text{-}cr}$ does not cover the cases where $\text{WIN} \notin \text{prj}[\text{id}](\mathcal{L}_{AA}, A)$, while $\text{ASJ}^{AB\text{-}cr}$ does not cover the cases where $\text{WIN} \notin \text{prj}[\text{supp}](\{L_{AJ}^{AB\text{-}cr}\}, \varphi)$. These observations lead directly to the following proposition.

Proposition 7.8. *The AF MLS $\text{mgen}[\text{SC}, \text{ASJ}^{AB\text{-}cr}] \circ \text{mgen}[\text{ID}, \text{AJ}^{AB\text{-}cr}]$ corresponding to the ABA formalism does not satisfy coverage of argument justification nor of statement justification.*

Nevertheless, full coverage can be straightforwardly provided by introducing a complementary ‘not winning’ labels NOWIN for arguments and nowin for statements (one could argue that this was left implicit in the definition of the formalism).

Definition 7.7. *Letting $\Lambda_{AJ}^{AB\text{-}cr\text{-}fc} = \{\text{WIN}, \text{NOWIN}\}$ the 1-synthesizer $\text{AJ}^{AB\text{-}cr\text{-}fc}$ from $\Lambda_{AA}^{\text{IOU}}$ to $\Lambda_{AJ}^{AB\text{-}cr\text{-}fc}$ is defined for $S \in \text{pow}(\Lambda_{AA}^{\text{IOU}})$ as*

- $\text{AJ}^{AB\text{-}cr\text{-}fc}(S) = \text{WIN}$ iff $S \supseteq \{\text{IN}\}$;
- $\text{AJ}^{AB\text{-}cr\text{-}fc}(S) = \text{NOWIN}$ otherwise.

Definition 7.8. *Letting $\Lambda_{ASJ}^{AB\text{-}cr\text{-}fc} = \{\text{win}, \text{nowin}\}$ the 1-synthesizer $\text{ASJ}^{AB\text{-}cr\text{-}fc}$ from $\Lambda_{AJ}^{AB\text{-}cr\text{-}fc}$ to $\Lambda_{ASJ}^{AB\text{-}cr\text{-}fc}$ is defined for $S, U \in \text{pow}(\Lambda_{AJ}^{AB\text{-}cr\text{-}fc})$ as*

- $\text{ASJ}^{AB\text{-}cr\text{-}fc}(S, U) = \text{win}$ iff $\text{WIN} \in S$;
- $\text{ASJ}^{AB\text{-}cr\text{-}fc}(S, U) = \text{nowin}$ otherwise.

Table 2 illustrates the statement justification labelling prescribed by Definition 7.8 depending on $\text{prj}[\text{supp}](\{L_{AJ}^{AB\text{-}cr\text{-}fc}\}, \varphi)$ and $\text{prj}[\text{cntr}](\{L_{AJ}^{AB\text{-}cr\text{-}fc}\}, \varphi)$.

Table 2: Justification status of a statement for ABA (credulous stance).

$\text{prj}[\text{cntr}] \setminus \text{prj}[\text{supp}]$	\emptyset	$\{\text{WIN}\}$	$\{\text{NOWIN}\}$	$\{\text{WIN}, \text{NOWIN}\}$
\emptyset	nowin	win	nowin	win
$\{\text{WIN}\}$	nowin	win	nowin	win
$\{\text{NOWIN}\}$	nowin	win	nowin	win
$\{\text{WIN}, \text{NOWIN}\}$	nowin	win	nowin	win

An AF MLS $\text{mgen}[\text{SC}, \text{ASJ}^{AB\text{-}cr\text{-}fc}] \circ \text{mgen}[\text{ID}, \text{AJ}^{AB\text{-}cr\text{-}fc}]$ satisfies both coverage of argument justification and of statement justification and still satisfies the correspondences with ABA stated in Proposition 7.6 for arguments labelled WIN and in Proposition 7.12 for statements labelled win .

Distinguishability. The (in)distinguishability of sets of labels in ABA can be examined for the versions of the operators with and without full coverage.

Regarding the argument justification labellings $L_{AJ}^{AB\cdot cr}$ and $L_{AJ}^{AB\cdot cr\cdot fc}$ and the relevant 1-synthesizers $AJ^{AB\cdot cr}$ and $AJ^{AB\cdot cr\cdot fc}$, for both operators the actual domain of definition depends on the adopted semantics and on the fact that $AJ^{AB\cdot cr}$ is defined only for sets of labels including IN . Accordingly, we develop the following considerations, analogous to those drawn in Section 7.1.1. In the case of a single-status semantics, each argument gets one and only one label, thus

- $\text{dom}(AJ^{AB\cdot cr}) = \{\{IN\}\}$ and
- $\text{dom}(AJ^{AB\cdot cr\cdot fc}) = \{\{IN\}, \{OUT\}, \{UN\}\}$.

In the case of a multiple-status semantics except the stable semantics, each argument gets at least one label, hence

- $\text{dom}(AJ^{AB\cdot cr}) = \{\Lambda \subseteq \Lambda_{AA}^{IOU} \mid IN \in \Lambda\}$ and
- $\text{dom}(AJ^{AB\cdot cr\cdot fc}) = \text{pow}(\Lambda_{AA}^{IOU}) \setminus \{\emptyset\}$.

In the case of stable semantics, an argument may have no labels, hence

- $\text{dom}(AJ^{AB\cdot cr}) = \{\{IN\}, \{IN, OUT\}\}$ and
- $\text{dom}(AJ^{AB\cdot cr\cdot fc}) = \text{pow}(\{IN, OUT\})$.

In all cases, for the argument justification labellings $L_{AJ}^{AB\cdot cr}$ and $L_{AJ}^{AB\cdot cr\cdot fc}$, the set of labels $\{OUT\}$ and $\{UN\}$ are indistinguishable.

Proposition 7.9. *It holds that $\{OUT\} \equiv [1, AJ^{AB\cdot cr}]\{UN\}$ and $\{OUT\} \equiv [1, AJ^{AB\cdot cr\cdot fc}]\{UN\}$.*

Proof. The statement follows from the fact that in every case, for every $\Lambda \in \text{dom}(AJ^{AB\cdot cr})$ the value of $AJ^{AB\cdot cr}(\Lambda)$ depends only on whether $IN \in \Lambda$. The same holds for $AJ^{AB\cdot cr\cdot fc}$. \square

Corollary 7.3. *The sets of labels $\{OUT\}$, $\{UN\}$ and $\{OUT, UN\}$ are pairwise strongly $AJ^{AB\cdot cr}$ -indistinguishable and strongly $AJ^{AB\cdot cr\cdot fc}$ -indistinguishable.*

As to $ASJ^{AB\cdot cr}$ and $ASJ^{AB\cdot cr\cdot fc}$, we have the following domains of definition:

- $\text{dom}(ASJ^{AB\cdot cr}) = \text{pow}(\{WIN\}) \times \text{pow}(\{WIN\})$;
- $\text{dom}(ASJ^{AB\cdot cr\cdot fc}) = \text{pow}(\{WIN, NOWIN\}) \times \text{pow}(\{WIN, NOWIN\})$.

In all cases, the empty set and $\{NOWIN\}$ are support-based strongly indistinguishable.

Proposition 7.10. *The sets of labels \emptyset and $\{NOWIN\}$ are support-based strongly $ASJ^{AB\cdot cr}$ -indistinguishable and support-based strongly $ASJ^{AB\cdot cr\cdot fc}$ -indistinguishable.*

Proof. The statement follows from the fact that in every case, for every $(\Lambda, \Lambda') \in \text{dom}(ASJ^{AB\cdot cr})$ the value of $ASJ^{AB\cdot cr}(\Lambda, \Lambda')$ depends only on whether $WIN \in \Lambda$. The same holds for $ASJ^{AB\cdot cr\cdot fc}$. \square

The simple consideration in the proof above (see also Table 2) shows as well that $ASJ^{AB\cdot cr}$ and $ASJ^{AB\cdot cr\cdot fc}$ are contrary-insensitive, i.e. all sets of labels are contrary-based indistinguishable.

Corollary 7.4. *$ASJ^{AB\cdot cr}$ and $ASJ^{AB\cdot cr\cdot fc}$ are contrary-insensitive.*

Example 7.2 (continues Example 1.1). *Referring to the acceptance evaluation prescribed by complete semantics given in Example 4.3, arguments are labelled according to the labelling $L_{AJ}^{AB\cdot cr\cdot fc}$ as follows:*

	A1	A2	A3	A4
$L_{AJ}^{AB\text{-}cr\text{-}fc}(\cdot)$	WIN	WIN	NOWIN	WIN

Accordingly, statements obtain the following justification statuses:

	s1	$\neg s1$	s2	$\neg s2$	s3
$L_{ASJ}^{AB\text{-}cr\text{-}fc}(\cdot)$	win	win	nowin	win	nowin

Similarly to the case of ASPIC⁺, the inability to distinguish the status of s2 and s3 is related to the fact that \emptyset and {NOWIN} are support-based strongly ASJ^{AB-cr.fc}-indistinguishable and the same outcome would be obtained with preferred, stable, and semi-stable semantics given that {OUT} and {UN} are pairwise strongly AJ^{AB-cr.fc}-indistinguishable.

The outcome would instead be different with the adoption of grounded, ideal, or eager semantics which would prescribe $\mathfrak{L}_{AA} = \{L_{AA}^3\}$. In this case we would get:

	A1	A2	A3	A4
$L_{AJ}^{AB\text{-}cr\text{-}fc}(\cdot)$	NOWIN	NOWIN	NOWIN	WIN

and

	s1	$\neg s1$	s2	$\neg s2$	s3
$L_{ASJ}^{AB\text{-}cr\text{-}fc}(\cdot)$	nowin	nowin	nowin	win	nowin

The inability to distinguish also the status of s1 and $\neg s1$ with respect to the one of s2 is again rooted in the fact that {OUT} and {UN} are strongly AJ^{AB-cr.fc}-indistinguishable leading to the same justification status for A1, A2, and A3. \square

We conclude this section by remarking that ABA in the credulous stance can be easily reconstructed also as a SF MLS. We omit the details of this alternative reconstruction for the sake of conciseness.

7.2.2 ABA (SKEPTICAL STANCE) AS A SF MLS

In addition to the credulous statement justification presented by Toni (2014) and reviewed above, Dimopoulos et al. (2002) considered a skeptical notion of justification: essentially a statement φ is skeptically justified if all acceptance labellings support φ , i.e. in every acceptance labelling there is one argument labelled IN with conclusion φ . On this basis, a more articulated classification of statement justification, distinguishing credulously, skeptically and not justified statements can be introduced. This stance, denoted as AB-sk, can be reconstructed in the SF approach as illustrated below. First, we formalise the notion of a labelling supporting a statement in Definition 7.9.

Definition 7.9. Given a language \mathcal{L} , a set of arguments \mathcal{A} and a labelling L_{AA} in a set \mathfrak{L}_{AA} of Λ_{AA}^{IOU} -labellings of \mathcal{A} , for every statement $\varphi \in \mathcal{L}$:

- φ is supported iff $\exists A \in \text{sup}(\{\varphi\}) : L_{AA}(A) = \text{IN}$;
- φ is not supported otherwise.

Letting $\Lambda_{SA}^{AB\text{-}sk} = \{\text{in}, \text{nin}\}$, Definition 7.9 corresponds to a stage $\Xi_1^{\text{SF}} = \langle \mathcal{L}, \Lambda_{SA}^{AB\text{-}sk}, \text{pgen}_{SA} \rangle$ where $\text{pgen}_{SA} = \text{pgen}[\text{SC}, \text{SA}^{AB\text{-}sk}]$ and $\text{SA}^{AB\text{-}sk} : \text{pow}(\Lambda_{AA}^{IOU}) \times \text{pow}(\Lambda_{AA}^{IOU}) \rightarrow \Lambda_{SA}^{AB\text{-}sk}$ is defined, for any $S, U \in \text{pow}(\Lambda_{AA}^{IOU})$, as

- $\text{SA}^{AB\text{-}sk}(S, U) = \text{in}$ iff $\text{IN} \in S$;
- $\text{SA}^{AB\text{-}sk}(S, U) = \text{nin}$ otherwise.

The correspondence is formally stated by the following proposition whose trivial proof is omitted.

Proposition 7.11. *Given a set of arguments \mathcal{A} and a set \mathfrak{L}_{AA} of $\Lambda_{AA}^{\text{IOU}}$ -labellings of \mathcal{A} , the set of statement acceptance labellings $\mathfrak{L}_{SA}^{\text{AB-sk}}$ prescribed by ABA according to Definition 7.9 is such that $\mathfrak{L}_{SA}^{\text{AB-sk}} = \text{pge}_{\Omega_{SA}}(\mathfrak{L}_{AA})$.*

The three possible statement justification statuses are then formalised in Definition 7.10

Definition 7.10. *Given a language \mathcal{L} , a set of arguments \mathcal{A} with conclusions in \mathcal{L} , and a set of statement acceptance $\Lambda_{SA}^{\text{AB-sk}}$ -labellings $\mathfrak{L}_{SA}^{\text{AB-sk}}$, for every statement $\varphi \in \mathcal{L}$,*

- φ is skeptically justified iff $\forall L \in \mathfrak{L}_{SA}^{\text{AB-sk}} : L(\varphi) = \text{in}$.
- φ is credulously justified iff $\exists L \in \mathfrak{L}_{SA}^{\text{AB-sk}} : L(\varphi) = \text{in}$ and $\exists L \in \mathfrak{L}_{SA}^{\text{AB-sk}} : L(\varphi) \neq \text{in}$.
- φ is not justified otherwise.

Letting $\Lambda_{SSJ}^{\text{AB-sk}} = \{\text{skj}, \text{crj}, \text{noj}\}$, Definition 7.10 corresponds to a stage $\Xi_2^{\text{SF}} = \langle \mathcal{L}, \Lambda_{SSJ}^{\text{AB-sk}}, \text{mgen}_{SSJ} \rangle$ where $\text{mgen}_{SSJ} = \text{mgen}[\text{SC}, \text{SSJ}^{\text{AB-sk}}]$ and $\text{SSJ}^{\text{AB-sk}} : \text{pow}(\Lambda_{SA}^{\text{AB-sk}}) \times \text{pow}(\Lambda_{SA}^{\text{AB-sk}}) \rightarrow \Lambda_{SSJ}^{\text{AB-sk}}$ is defined, for any $S, U \in \text{pow}(\Lambda_{SA}^{\text{AB-sk}})$,

- $\text{SSJ}^{\text{AB-sk}}(S, U) = \text{skj}$ iff $S = \{\text{in}\}$;
- $\text{SSJ}^{\text{AB-sk}}(S, U) = \text{crj}$ iff $S \supseteq \{\text{in}\}$;
- $\text{SSJ}^{\text{AB-sk}}(S, U) = \text{noj}$ otherwise.

The correspondence is formally stated by the following proposition, whose proof is again straightforward.

Proposition 7.12. *Given a language \mathcal{L} , a set of arguments \mathcal{A} with conclusions in \mathcal{L} , and a set of statement acceptance $\Lambda_{SA}^{\text{AB-sk}}$ -labelling $\mathfrak{L}_{SA}^{\text{AB-sk}}$, the statement justification labelling $\mathbb{L}_{SSJ}^{\text{AB-sk}}$ prescribed by $\text{AB} \cdot \text{sk}$ ABA according to Definition 7.10 is such that $\{\mathbb{L}_{SSJ}^{\text{AB-sk}}\} = \text{mgen}_{SSJ}(\mathfrak{L}_{SA}^{\text{AB-sk}})$.*

Table 3 illustrates the statement justification labelling prescribed by Definition 7.10 depending on $\text{prj}[\text{supp}](\mathfrak{L}_{SA}^{\text{AB-sk}}, \varphi)$ and $\text{prj}[\text{cntr}](\mathfrak{L}_{SA}^{\text{AB-sk}}, \varphi)$.

Table 3: Justification status of a statement for ABA (skeptical stance).

$\text{prj}[\text{cntr}] \setminus \text{prj}[\text{supp}]$	$\{\text{nin}\}$	$\{\text{in}\}$	$\{\text{nin}, \text{in}\}$
$\{\text{nin}\}$	noj	skj	crj
$\{\text{in}\}$	noj	n.a.	n.a.
$\{\text{nin}, \text{in}\}$	noj	n.a.	crj

Let us now turn to the properties of $\text{AB} \cdot \text{sk}$. First of all, coverage is obviously satisfied.

Coverage. The 2-synthesizers $\text{SA}^{\text{AB-sk}}$ and $\text{SSJ}^{\text{AB-sk}}$ provide full coverage of statement acceptance and justification respectively, and thus the SF MLS $\text{mgen}_{SSJ} \circ \text{pge}_{\Omega_{SA}}$ provides coverage for the two stages.

Proposition 7.13. *The SF MLS $\text{mgen}_{SSJ} \circ \text{pge}_{\Omega_{SA}}$ does satisfy coverage of statement acceptance and of statement justification.*

Distinguishability. As to $\text{dom}(\text{SA}^{\text{AB-sk}})$, it can be observed that in general a statement can be the conclusion of many arguments (with different acceptance labels) and possibly also of no argument at all. Thus in any case $\text{dom}(\text{AJ}^{\text{AB-cr.fc}}) = \text{pow}(\Lambda_{AA}^{\text{IOU}}) \times \text{pow}(\Lambda_{AA}^{\text{IOU}})$.

In all cases, the sets of labels \emptyset , $\{\text{OUT}\}$, $\{\text{UN}\}$ are support-based strongly $\text{SA}^{\text{AB-sk}}$ -indistinguishable.

Proposition 7.14. *It holds that $\emptyset \equiv [1, \text{SA}^{\text{AB-sk}}]\{\text{UN}\}$, $\emptyset \equiv [1, \text{SA}^{\text{AB-sk}}]\{\text{OUT}\}$, and $\{\text{OUT}\} \equiv [1, \text{SA}^{\text{AB-sk}}]\{\text{UN}\}$.*

Proof. The statement follows from the fact that for every $(\Lambda, \Lambda') \in \text{dom}(\text{SA}^{\text{AB-sk}})$ the value of $\text{SA}^{\text{AB-sk}}(\Lambda, \Lambda')$ depends only on whether $\text{IN} \in \Lambda$. \square

Corollary 7.5. *The sets of labels \emptyset , $\{\text{OUT}\}$, $\{\text{UN}\}$ and $\{\text{OUT}, \text{UN}\}$ are pairwise support-based strongly $\text{SA}^{\text{AB-sk}}$ -indistinguishable.*

From the simple consideration in the proof of Proposition 7.14 the following proposition also follows.

Proposition 7.15. *$\text{SA}^{\text{AB-sk}}$ is contrary-insensitive.*

Turning to $\text{SSJ}^{\text{AB-sk}}$, we have that $\text{dom}(\text{SSJ}^{\text{AB-sk}}) = \{(\{\text{in}\}, \{\text{nin}\}), (\{\text{in}, \text{nin}\}, \{\text{nin}\}), (\{\text{in}, \text{nin}\}, \{\text{in}, \text{nin}\}), (\{\text{nin}\}, \{\text{nin}\}), (\{\text{nin}\}, \{\text{in}\}), (\{\text{nin}\}, \{\text{in}, \text{nin}\})\}$ (see also Table 3). This is due to the fact that if a statement φ is labelled in in a statement acceptance labelling, none of its contraries can be labelled in in the same labelling. It follows that if φ is labelled in in all statement acceptance labellings, i.e. $\text{prj}[\text{supp}](\{\text{L}_{\text{AJ}}^{\text{AB-sk}}\}, \varphi) = \{\text{in}\}$, it cannot be the case that $\text{in} \in \text{prj}[\text{cntr}](\{\text{L}_{\text{AJ}}^{\text{AB-sk}}\}, \varphi)$. A dual observation applies to the case $\text{prj}[\text{cntr}](\{\text{L}_{\text{AJ}}^{\text{AB-sk}}\}, \varphi) = \{\text{in}\}$.

It is easy to see that the sets $\{\text{in}\}$, $\{\text{nin}\}$, $\{\text{in}, \text{nin}\}$ are support-based distinguishable $\text{SSJ}^{\text{AB-sk}}$, while they are all strongly contrary-based indistinguishable since $\text{SSJ}^{\text{AB-sk}}$ is contrary-insensitive, as evident from its definition.

Proposition 7.16. *$\text{SSJ}^{\text{AB-sk}}$ is contrary-insensitive.*

Example 7.3 (continues Example 1.1). *Referring to the acceptance evaluation given in Example 4.3, with $\mathfrak{L}_{\text{AA}} = \{\text{L}_{\text{AA}}^1, \text{L}_{\text{AA}}^2, \text{L}_{\text{AA}}^3\}$, we get $\text{pgen}_{\text{SA}}(\mathfrak{L}_{\text{AA}}) = \{\text{L}_{\text{SA}}^{\text{AB-sk}\cdot 1}, \text{L}_{\text{SA}}^{\text{AB-sk}\cdot 2}, \text{L}_{\text{SA}}^{\text{AB-sk}\cdot 3}\}$ as follows:*

	s1	\neg s1	s2	\neg s2	s3
$\text{L}_{\text{SA}}^{\text{AB-sk}\cdot 1}(\cdot)$	in	nin	nin	in	nin
$\text{L}_{\text{SA}}^{\text{AB-sk}\cdot 2}(\cdot)$	nin	in	nin	in	nin
$\text{L}_{\text{SA}}^{\text{AB-sk}\cdot 3}(\cdot)$	nin	nin	nin	in	nin

Applying then mgen_{SSJ} , we get the following statement justification labelling.

	s1	\neg s1	s2	\neg s2	s3
$\text{L}_{\text{SSJ}}^{\text{AB-sk}}(\cdot)$	crj	crj	noj	skj	noj

Statements s2 and s3 get the same justification status noj, due to the fact that \emptyset , $\{\text{OUT}\}$, $\{\text{UN}\}$ are support-based strongly $\text{SA}^{\text{AB-sk}}$ -indistinguishable, and, as in the previous examples, the outcome would be the same applying preferred, stable or semi-stable semantics. With grounded, ideal, or eager semantics, which would prescribe $\mathfrak{L}_{\text{AA}} = \{\text{L}_{\text{AA}}^3\}$, we would get $\text{pgen}_{\text{SA}}(\mathfrak{L}_{\text{AA}}) = \{\text{L}_{\text{SA}}^{\text{AB-sk}\cdot 3}\}$ and then all statements but \neg s2 would get the justification label noj. The inability to distinguish the status of s1 and \neg s1 with respect to s2 and s3 in this case is again due to the fact that \emptyset , $\{\text{OUT}\}$, $\{\text{UN}\}$ are support-based strongly $\text{SA}^{\text{AB-sk}}$ -indistinguishable. \square

To conclude this section we observe that a reconstruction of $_{AB} \cdot sk$ is not possible in the AF approach. This can be proved with the same line of reasoning used in the first part of Section 5 and in particular by instantiating cases C1 and C2 as follows.

C1. The situation of arguments A and B in the case C1 can be obtained, for instance with a Dung’s argumentation framework consisting of four arguments, A, B, C, D , where C and D mutually attack each other and C attacks A , while D attacks B . Applying stable semantics to this framework gives rise to two labellings L_{AA}^1 and L_{AA}^2 whose restriction on A and B is as described in Section 5.

C2. The situation of argument A in the case C2 can be obtained with a Dung’s argumentation framework consisting of three arguments, A, B, C , where B and C mutually attack each other, and B attacks A . Again, applying stable semantics to this framework gives rise to two labellings L_{AA}^1 , L_{AA}^2 , whose restriction on A is as described in Section 5.

Under the assumption that $\text{sup}(\{\varphi\}) = \{A, B\}$, we get that in case C1, two statement acceptance labellings in both of which φ is labelled in, from which $L_{SSJ}^{AB \cdot sk}(\varphi) = skj$. In case C4 we get two statement acceptance labellings: one where φ is labelled in and the other where φ is labelled nin, from which $L_{SSJ}^{AB \cdot sk}(\varphi) = crj$. So the justification of statement φ is different in the two cases, while as shown in Section 5 this difference cannot be obtained in the argument-focused model.

7.3 Defeasible Logic Programming

Defeasible Logic Programming (denoted as *DeLP*) ‘provides a computational reasoning system that uses an argumentation engine to obtain answers from a knowledge base represented in a logic programming language extended with defeasible rules’ (García & Simari, 2014).

DeLP assumes a language based on a set of atoms *Atoms* and equipped with two forms of negation, namely strong negation (denoted as \sim) and default negation (denoted as *not*). A literal of the language is either an atom or an atom preceded by strong negation, while atoms preceded by default negation are called extended literals. As it will be explained later, *DeLP* does not encompass the evaluation of extended literals, whose existence can therefore be abstracted away in our formalisation. Accordingly, for our purposes we can assume a language $\mathcal{L} = Atoms \cup \{\sim\alpha \mid \alpha \in Atoms\}$ and the notion of strong negation can be captured by our contrariness relation in Definition 4.2 by letting $\text{cnt}(\alpha) = \{\sim\alpha\}$, $\text{cnt}(\sim\alpha) = \{\alpha\}$ for every $\alpha \in Atoms$. Using the terminology of García and Simari (2014), α and $\sim\alpha$ are *complement* of each other.

A *DeLP* program consists of *facts*, i.e. information holding with certainty in the application domain, *strict rules* and *defeasible rules*. An argument for a conclusion φ is a minimal set S of defeasible rules such that φ can be derived from S together with facts and strict rules and some internal consistency requirements are satisfied. Abstracting away the construction mechanism and the consistency requirements, the set of arguments built on the basis of a *DeLP* program can therefore be regarded as an argument-conclusion structure according to Definition 4.3, and for each set of statements $\Phi \subseteq \mathcal{L}$ it is possible to identify the set of supporting arguments.

Differently from the approaches surveyed in the previous sections, *DeLP* does not use Dung’s framework for argument acceptance evaluation, rather it adopts a dialectical procedure corresponding to a single-status approach where each argument is marked as D(efeated) or U(ndefeated). Letting $\Lambda_{AA}^{De} = \{D, U\}$, in the MLS model we can then identify the first stage as the pair $\Xi_0 = \langle \mathcal{A}, \Lambda_{AA}^{De} \rangle$, where \mathcal{A} is the set of arguments produced by the considered *DeLP* program and the set \mathfrak{L}_{AA}^{De} of Λ_{AA}^{De} -labellings of \mathcal{A} is a singleton by construction.

7.3.1 *DeLP* AS AN AF MLS

In order to reconstruct *DeLP* as an AF MLS, we first observe that when the set \mathcal{L}_{AA} of argument acceptance labellings is always a singleton, the distinction between argument acceptance and argument justification becomes substantially void and the two notions coincide. Accordingly it is possible to define a simple stage $\Xi_1^{AF} = \langle \mathcal{A}, \Lambda_{AA}^{De}, \text{mgen}_{AJ} \rangle$ where $\text{mgen}_{AJ} = \text{mgen}[\text{ID}, \text{AJ}^{De}]$ and $\text{AJ}^{De} : \{\{D\}, \{U\}\} \rightarrow \{D, U\}$ is defined, as $\text{AJ}^{De}(\{D\}) = D$; $\text{AJ}^{De}(\{U\}) = U$.

A statement is said ‘warranted’ if it is the conclusion of an argument whose justification label is U . On this simple basis, an articulated notion of justification status for a literal φ based on four labels (corresponding to the possible answers to a *DeLP* query) is introduced (see Section 4 of García & Simari, 2014 and Definition 5.3 of García & Simari, 2004).

Definition 7.11. *Given a literal φ , there are four possible answers for a *DeLP* query about φ :*

- yes iff φ is warranted;
- no iff the complement of φ is warranted;¹⁰
- und(ecided) iff neither φ nor its complement are warranted;
- unk(nown) iff φ is not in the signature of the program¹¹.

Letting $\Lambda_{ASJ}^{De} = \{\text{yes}, \text{no}, \text{und}, \text{unk}\}$, Definition 7.11 can be formulated as a stage $\Xi_2^{AF} = \langle \mathcal{L}, \Lambda_{ASJ}^{De}, \text{mgen}_{ASJ} \rangle$ where $\text{mgen}_{ASJ} = \text{mgen}[\text{SC}, \text{ASJ}^{De}]$ and $\text{ASJ}^{De} : \text{pow}(\Lambda_{AA}^{De}) \times \text{pow}(\Lambda_{AA}^{De}) \rightarrow \Lambda_{ASJ}^{De}$ is defined, for any $S, U \in \text{pow}(\text{ASJ}^{De})$, as

- $\text{ASJ}^{De}(S, U) = \text{yes}$ iff $U \in S$;
- $\text{ASJ}^{De}(S, U) = \text{no}$ iff $U \in U$;
- $\text{ASJ}^{De}(S, U) = \text{und}$ iff $S \cup U = \{D\}$;
- $\text{ASJ}^{De}(S, U) = \text{unk}$ iff $S \cup U = \emptyset$.

To prove the correspondence of the above defined stage Ξ_2^{AF} with Definition 7.11, a suitable relevance hypothesis is required for literals included in the signature of a program.

Definition 7.12. *A literal φ is relevant for a *DeLP* program Pr if there is at least an argument whose conclusion is φ or its complement.*

Proposition 7.17. *Given a language \mathcal{L} , let Pr be a *DeLP* program based on \mathcal{L} such that every literal in the signature of Pr is relevant for Pr , \mathcal{A} be the set of arguments built from Pr , and L_{AJ}^{De} be a Λ_{AA}^{De} -justification labelling of \mathcal{A} . The statement justification labelling L_{ASJ}^{De} prescribed by *DeLP* according to Definition 7.11 is such that $\{L_{ASJ}^{De}\} = \text{mgen}_{ASJ}(\{L_{AJ}^{De}\})$.*

Proof. For the sake of conciseness let $\text{mgen}_{ASJ}(\{L_{AJ}^{De}\}) = \{L'_{ASJ}\}$, we need to show that for every literal φ $L_{ASJ}^{De}(\varphi) = L'_{ASJ}(\varphi)$.

- According to Definition 7.11, $L_{ASJ}^{De}(\varphi) = \text{yes}$ iff φ is warranted, iff $\exists A \in \text{sup}(\{\varphi\})$ such that $L_{AJ}^{De}(A) = U$, iff $U \in \text{prj}[\text{supp}](\{L_{AJ}^{De}\}, \varphi)$ leading to $\text{ASJ}^{De}(\text{prj}[\text{supp}](\{L_{AJ}^{De}\}, \varphi), \text{prj}[\text{cntr}](\{L_{AJ}^{De}\}, \varphi)) = \text{yes}$ and then $L'_{ASJ}(\varphi) = \text{yes}$.
- Similarly, $L_{ASJ}^{De}(\varphi) = \text{no}$ iff the complement of φ is warranted iff $\exists A \in \text{sup}(\{\text{cnt}(\varphi)\})$ such that $L_{AJ}^{De}(A) = U$ iff $U \in \text{prj}[\text{cntr}](\{L_{AJ}^{De}\}, \varphi)$ leading to $\text{ASJ}^{De}(\text{prj}[\text{supp}](\{L_{AJ}^{De}\}, \varphi), \text{prj}[\text{cntr}](\{L_{AJ}^{De}\}, \varphi)) = \text{no}$ and then $L'_{ASJ}(\varphi) = \text{no}$.

10. Note that the formalism guarantees that a statement and its complement cannot be warranted at the same time.

11. The signature of the program is the set of all literals mentioned in the facts and rules of the program. We do not recall the relevant detailed formalisation for the sake of conciseness.

- As to $L_{ASJ}^{De}(\varphi) = \text{unk}$, the condition holds iff φ is not in the signature of the program which implies that $\text{sup}(\{\varphi\}) = \text{sup}(\{\text{cnt}(\varphi)\}) = \emptyset$ leading to $L'_{ASJ}(\varphi) = \text{ASJ}^{De}(\emptyset, \emptyset) = \text{unk}$.
- Finally, as to $L_{ASJ}^{De}(\varphi) = \text{und}$ the condition holds iff φ is in the signature of the program, which, by the relevance hypothesis, implies that $\text{sup}(\{\varphi\}) \neq \emptyset$ or $\text{sup}(\{\text{cnt}(\varphi)\}) \neq \emptyset$. We have also that neither φ nor $\sim\varphi$ are warranted, which implies that $U \notin \text{prj}[\text{supp}](\{L_{AJ}^{De}\}, \varphi)$ and $U \notin \text{prj}[\text{cntr}](\{L_{AJ}^{De}\}, \varphi)$ which, together with the previous non emptiness conditions, implies $\text{prj}[\text{supp}](\{L_{AJ}^{De}\}, \varphi) \cup \text{prj}[\text{cntr}](\{L_{AJ}^{De}\}, \varphi) = \{D\}$ and then $L'_{ASJ}(\varphi) = \text{ASJ}^{De}(\text{prj}[\text{supp}](\{L_{AJ}^{De}\}, \varphi), \text{prj}[\text{cntr}](\{L_{AJ}^{De}\}, \varphi)) = \text{und}$.

□

Table 4 illustrates the statement justification labelling prescribed by Definition 7.11 depending on $\text{prj}[\text{supp}](\{L_{AJ}^{De}\}, \varphi)$ and $\text{prj}[\text{cntr}](\{L_{AJ}^{De}\}, \varphi)$.

Table 4: Justification status of a statement for *DeLP*.

$\text{prj}[\text{cntr}] \setminus \text{prj}[\text{supp}]$	\emptyset	$\{U\}$	$\{D\}$	$\{U, D\}$
\emptyset	unk	yes	und	yes
$\{U\}$	no	n.a.	no	n.a.
$\{D\}$	und	yes	und	yes
$\{U, D\}$	no	n.a.	no	n.a.

As a remark, we may note that if one drops the hypothesis of relevance, the coincidence stated in Proposition 7.17 might fail for those literals which are in the signature of the program but are not relevant: they would be labelled *und* according to Definition 7.11, while would be labelled *unk* by $\text{mgen}[\text{SC}, \text{ASJ}^{De}]$. We suggest that this remark shows a little conceptual incongruence in the original definition of *DeLP*: a literal φ such that $\text{sup}(\{\varphi\}) = \text{sup}(\{\text{cnt}(\varphi)\}) = \emptyset$ is labelled *unk* if it is not included in the signature of the program, and is labelled *und* otherwise. However, under the condition $\text{sup}(\{\varphi\}) = \text{sup}(\{\text{cnt}(\varphi)\}) = \emptyset$, the fact that φ is included in the signature of the program means that φ is mentioned in a rule included in the program but which actually is never applied, and it seems reasonable that rules which are never applied should not affect the justification status of a statement. To put it in other words, if the hypothesis of relevance is removed, the labelling produced by $\text{mgen}[\text{SC}, \text{ASJ}^{De}]$ can be regarded as an improved version of the one given by Definition 7.11 as far as the statements where they disagree are concerned.

Having reconstructed *DeLP* as an AF MLS, we can proceed to analyse its properties.

Coverage. It is immediate to see that both AJ^{De} and ASJ^{De} satisfy coverage, leading to the following proposition.

Proposition 7.18. *The AF MLS $\text{mgen}[\text{SC}, \text{ASJ}^{De}] \circ \text{mgen}[\text{ID}, \text{AJ}^{De}]$ corresponding to the *DeLP* formalism satisfies coverage of argument justification and of statement justification.*

Distinguishability.

As to argument justification, $\text{dom}(\text{AJ}^{De}) = \{\{D\}, \{U\}\}$ and clearly $\{D\}$ and $\{U\}$ are distinguishable.

As to statement justification, we have that $\text{dom}(\text{ASJ}^{De}) = \text{pow}(\{D\}) \times \text{pow}(\{D, U\}) \cup \{\{U\}, \{D, U\}\} \times \text{pow}(\{D\})$ and it can be noted that the set of labels $\{U\}$ and $\{D, U\}$ are strongly indistinguishable with respect to ASJ^{De} .

Proposition 7.19. *It holds that $\{U\} \equiv [1, \text{ASJ}^{\text{De}}]\{D, U\}$ and $\{U\} \equiv [2, \text{ASJ}^{\text{De}}]\{D, U\}$.*

Proof. With reference Definition 3.12, let $\Lambda_1 = \{D, U\}$. For the first part of the statement we have to prove that $\forall \Lambda', \Lambda'' \subseteq \Lambda_1$ such that $(\Lambda' \cup \{U\}, \Lambda''), (\Lambda' \cup \{D, U\}, \Lambda'') \in \text{dom}(\text{ASJ}^{\text{De}})$ it holds that $\text{ASJ}^{\text{De}}(\Lambda' \cup \{U\}, \Lambda'') = \text{ASJ}^{\text{De}}(\Lambda' \cup \{D, U\}, \Lambda'')$. By inspection of the definition of ASJ^{De} it is evident that the statement holds with $\text{ASJ}^{\text{De}}(\Lambda' \cup \{U\}, \Lambda'') = \text{ASJ}^{\text{De}}(\Lambda' \cup \{D, U\}, \Lambda'') = \text{yes}$. The proof of the second part of the statement is analogous: $\forall \Lambda', \Lambda'' \subseteq \Lambda_1$ such that $(\Lambda', \Lambda'' \cup \{U\}), (\Lambda', \Lambda'' \cup \{D, U\}) \in \text{dom}(\text{ASJ}^{\text{De}})$ it is easy to see that $\text{ASJ}^{\text{De}}(\Lambda', \Lambda'' \cup \{U\}) = \text{ASJ}^{\text{De}}(\Lambda', \Lambda'' \cup \{D, U\}) = \text{no}$. \square

As a direct consequence, when an argument for a literal (or its complement) is warranted, the presence of other arguments (warranted or not) for the same literal makes no difference, i.e. the sets $\{U\}$ and $\{D, U\}$ are both support-based and contrary-based strongly ASJ^{De} indistinguishable.

Corollary 7.6. *The sets of labels $\{U\}$ and $\{D, U\}$ are support-based and contrary-based strongly ASJ^{De} -indistinguishable.*

It can also be noted that statement justification labelling distinguishes three cases of non acceptance (while non acceptance was somehow overlooked in the previously surveyed formalisms): this is also related to the fact that this approach is contrary-sensitive, as it is clear from the definition of ASJ^{De} .

Proposition 7.20. *ASJ^{De} is contrary-sensitive.*

Example 7.4 (continues Example 1.1). *According to the dialectical semantics of DeLP, only the uncontroversial argument A4 is undefeated, giving rise to the acceptance labelling $\mathcal{L}_{\text{AA}}^{\text{De}} = \{L_{\text{AA}}^{\text{De}}\}$ where $L_{\text{AA}}^{\text{De}}$ is as follows:*

	A1	A2	A3	A4
$L_{\text{AA}}^{\text{De}}(\cdot)$	D	D	D	U

and then to an identical justification labelling $L_{\text{AJ}}^{\text{De}} = L_{\text{AA}}^{\text{De}}$ formally obtained by applying mgen_{AJ} to \mathcal{L}_{AA} . Accordingly, theddd statements obtain the following justification statuses:

	s1	\neg s1	s2	\neg s2	s3
$L_{\text{ASJ}}^{\text{De}}(\cdot)$	und	und	no	yes	unk

Thus DeLP is able to fully distinguish the status of the statements involved in our simple example, regarding as undecided the statements s1 and \neg s1 as the arguments supporting them are involved in a mutual conflict, recognising that s2 is strongly rejected since \neg s2 is warranted, and labelling s3 as unknown. \square

To conclude, we observe that, in virtue of Proposition 5.1, the DeLP statement justification labelling can be reconstructed in the SF approach too.

7.4 Towards Tunable Justification Notions

It emerges that different argumentation formalisms adopt quite different notions of justification, both at the level of arguments and of statements, featuring different properties and sometimes failing to satisfy some intuitive requirements like full coverage and contrary-sensitivity. However, these differences do not seem to be caused by technical motivations, but rather depend on specific

choices based on the intended use of the notion of justification in the presentation of the formalisms themselves.

These observations back up our claim that the notion of justification (and in particular of statement justification) has received comparatively less attention in the development of argumentation formalisms, often more focused on the notion of argument acceptance. Moreover they suggest that justification notions, instead of being ‘hardwired’ in the definitions could be better conceived as tunable components of any argumentation formalism, with a role similar to those played by argumentation semantics in *ASPIC*⁺ or *ABA*. These formalisms do not stick to a single argumentation semantics, rather they assume that one is chosen among the various available ones (including possibly those to be developed in the future).

Further, it is interesting to appreciate the different advantages of the reviewed approaches. For instance, *DeLP* has shown the best ability to distinguish statement justification statuses in our simple example. However, since *DeLP* is single-status at the stage of argument acceptance, it encompasses a single notion of positive justification for arguments, while a multiple-status approach could induce a finer distinction. This suggests that combining the most expressive aspects of different approaches may give rise to a more general treatment of the notion of argument and statement justification.

On the basis of these considerations, we show in the next sections that, as in the case of argumentation semantics, a portfolio of alternatives can be conceived for statement justification notions. To this purpose we will provide several examples of generic approaches to statement justification and illustrate how they can be integrated within the formalisms we have reviewed.

8. Bivalent Labellings

Bivalent labellings assume a simple binary justification for statements (say, ‘yes’ or ‘no’) without further sophistication. Assuming accordingly that $\Lambda_{SJ}^{yn} = \{\text{yes}, \text{no}\}$ and regarding full coverage as a basic requirement, the adoption of a bivalent labelling in a formalism essentially amounts to specify a criterion to determine whether a statement φ is justified, and to regard φ as not justified if the criterion is not satisfied. In the context of the AF approach, this can be formalised by a stage $\Xi_2^{AF} = \langle \mathcal{L}, \Lambda_{SJ}^{yn}, \text{mgen}_{ASJ} \rangle$ where $\text{mgen}_{ASJ} = \text{mgen}[\text{SC}, \text{syn}_{ASJ}]$ and syn_{ASJ} is a 2-synthesizer from Λ_{AJ} to Λ_{SJ}^{yn} (with Λ_{AJ} the set of argument justification labels used in the stage Ξ_1^{AF}). Similarly, in the context of the SF approach this can be formalised by a stage $\Xi_2^{SF} = \langle \mathcal{L}, \Lambda_{SJ}^{yn}, \text{mgen}_{SSJ} \rangle$ where $\text{mgen}_{SSJ} = \text{mgen}[\text{IC}, \text{syn}_{SSJ}]$ and syn_{SSJ} is a 2-synthesizer from Λ_{SA} to Λ_{SJ}^{yn} (with Λ_{SA} the set of statement acceptance labels used in the stage Ξ_1^{SF}). Let us now see how this idea can be plugged into the formalisms we considered.

8.1 *ASPIC*⁺

Let us first look at *ASPIC*⁺ as an AF MLS (Section 7.1.1) and assume $\Lambda_{AJ}^{A+fc} = \{\text{SKJ}, \text{CRJ}, \text{NOJ}\}$ (Definition 7.3). Two basic versions of bivalent statement labelling can be set up. A first option, which is skeptically oriented, corresponds to the idea that a statement is labelled yes if it is supported by a skeptically justified argument, no otherwise. A second option, which is credulously oriented, labels a statement yes if it is supported by a skeptically or credulously justified argument, no otherwise. They can be captured in our scheme by ‘plugging’ in the AF MLS scheme the following 2-synthesizers respectively.

Definition 8.1. *The skeptical bivalent 2-synthesizer for ASPIC⁺, denoted as $ASJ^{A^+ \cdot yn-sk}$, is defined as follows*

- $ASJ^{A^+ \cdot yn-sk}(S, U) = \text{yes}$ iff $SKJ \in S$;
- $ASJ^{A^+ \cdot yn-sk}(S, U) = \text{no}$ otherwise.

Definition 8.2. *The credulous bivalent 2-synthesizer for ASPIC⁺, denoted as $ASJ^{A^+ \cdot yn-cr}$, is defined as follows*

- $ASJ^{A^+ \cdot yn-cr}(S, U) = \text{yes}$ iff $\{SKJ, CRJ\} \cap S \neq \emptyset$;
- $ASJ^{A^+ \cdot yn-cr}(S, U) = \text{no}$ otherwise.

Tables 5 and 6 illustrate the statement justification labellings prescribed by Definitions 8.1 and 8.2 respectively.

Table 5: Bivalent labelling for ASPIC⁺ (skeptical version)

prj[ctr] \ prj[supp]	\emptyset	{NOJ}	{CRJ}	{NOJ, CRJ}	$\supseteq \{SKJ\}$
\emptyset	no	no	no	no	yes
{NOJ}	no	no	no	no	yes
{CRJ}	no	no	no	no	n.a.
{NOJ, CRJ}	no	no	no	no	n.a.
$\supseteq \{SKJ\}$	no	no	n.a.	n.a.	n.a.

Table 6: Bivalent labelling for ASPIC⁺ (credulous version)

prj[ctr] \ prj[supp]	\emptyset	{NOJ}	{CRJ}	{NOJ, CRJ}	$\supseteq \{SKJ\}$
\emptyset	no	no	yes	yes	yes
{NOJ}	no	no	yes	yes	yes
{CRJ}	no	no	yes	yes	n.a.
{NOJ, CRJ}	no	no	yes	yes	n.a.
$\supseteq \{SKJ\}$	no	no	n.a.	n.a.	n.a.

Clearly both $ASJ^{A^+ \cdot yn-sk}$ and $ASJ^{A^+ \cdot yn-cr}$ ensure full coverage and are contrary-insensitive. We can also note that they represent two alternatives which are less refined versions than $ASJ^{A^+ \cdot fc}$ (Definition 7.4). In particular (compare Table 1 with Tables 5 and 6 respectively):

- $ASJ^{A^+ \cdot fc}$ refines $ASJ^{A^+ \cdot yn-sk}$ with $\text{ref}(\text{yes}) = \{skj\}$ and $\text{ref}(\text{no}) = \{crj, noj\}$;
- $ASJ^{A^+ \cdot fc}$ refines $ASJ^{A^+ \cdot yn-cr}$ with $\text{ref}(\text{yes}) = \{skj, crj\}$ and $\text{ref}(\text{no}) = \{noj\}$.

Example 8.1. *Applying $ASJ^{A^+ \cdot yn-sk}$ in the context of Example 7.1 in the case of complete semantics we get the following statement justification labelling:*

	s1	$\neg s1$	s2	$\neg s2$	s3
$L_{ASJ}^{A^+ \cdot yn-sk}(\cdot)$	no	no	no	yes	no

while applying $ASJ^{A^+ \cdot yn-cr}$ we get:

	s1	$\neg s1$	s2	$\neg s2$	s3
$L_{ASJ}^{A^+ \cdot yn-cr}(\cdot)$	yes	yes	no	yes	no

□

It is worth noting that the two proposed bivalent labellings for $ASPIC^+$ are probably the simplest possible ones (in particular they are contrary-insensitive). It can be observed, however, that the simplistic choice of bivalent labelling does not prevent that a contrary-sensitive synthesizer can be devised. For instance a ‘semi-credulous’ approach might be defined as follows.

Definition 8.3. *The semi-credulous bivalent 2-synthesizer for $ASPIC^+$, denoted as $ASJ^{A^+ \cdot yn \cdot sc}$, is defined as follows*

- $ASJ^{A^+ \cdot yn \cdot sc}(S, U) = \text{yes}$ iff $(SKJ \in S) \vee (CRJ \in S \wedge CRJ \notin U)$;
- $ASJ^{A^+ \cdot yn \cdot sc}(S, U) = \text{no}$ otherwise.

The idea underlying the semi-credulous can be viewed as a novel intermediate attitude between the traditional skeptical and credulous views, inspired by the property of contrary-sensitivity. We leave the investigation of this kind of developments to future work, and remark that they can be regarded as a further confirmation of the expressiveness of our approach.

8.2 ABA

We discuss bivalent labellings in distinct subsections for the two stances of ABA .

8.2.1 ABA (CREDULOUS STANCE)

As to the credulous stance of ABA (Section 7.2.1), it can be noted that it already implicitly adopts a bivalent credulous labelling if one replaces win with yes and nowin with no (we will refer to this replacement as $SSJ^{AB \cdot yn \cdot cr}$). Hence the case of the credulous stance of ABA does not require further discussion.

8.2.2 ABA (SKEPTICAL STANCE)

As to integrating bivalent labelling in the skeptical stance of ABA , it amounts to replacing the 2-synthesizer $SSJ^{AB \cdot sk}$ introduced in Section 7.2.2 with a 2-synthesizer $SSJ^{AB \cdot yn \cdot sk}$ from $\Lambda_{SA}^{AB \cdot sk}$ to Λ_{SJ}^{yn} . Among various possible ways of defining $SSJ^{AB \cdot yn \cdot sk}$ we propose the following one¹² (illustrated in Table 7).

Definition 8.4. *The skeptical bivalent 2-synthesizer for ABA , denoted as $SSJ^{AB \cdot yn \cdot sk}$, is defined as follows*

- $ASJ^{AB \cdot yn \cdot sk}(S, U) = \text{yes}$ iff $S = \{\text{in}\}$;
- $ASJ^{AB \cdot yn \cdot sk}(S, U) = \text{no}$ otherwise.

Table 7: Bivalent labelling for ABA (skeptical stance).

prj[ctr] \ prj[supp]	{nin}	{in}	{nin, in}
{nin}	no	yes	no
{in}	no	n.a.	n.a.
{nin, in}	no	n.a.	no

12. Though belonging to a different context, $SSJ^{AB \cdot yn \cdot sk}$ has clearly a similarity with $ASJ^{A^+ \cdot yn \cdot sk}$ defined in Section 8.1. Other variants similar to $ASJ^{A^+ \cdot yn \cdot cr}$ and $ASJ^{A^+ \cdot yn \cdot sc}$ could also be conceived. This is left to future work.

Clearly $SSJ^{AB\cdot yn\cdot sk}$ ensures full coverage, is contrary-insensitive and represents a less refined version of $SSJ^{AB\cdot sk}$ where $ref(\text{yes}) = \{skj\}$ and $ref(\text{no}) = \{crj, noj\}$.

Example 8.2. Applying $SSJ^{AB\cdot yn\cdot sk}$ in the context of Example 7.3 in the case of complete semantics we get the following statement justification labelling:

	s1	\neg s1	s2	\neg s2	s3
$L_{SSJ}^{AB\cdot yn\cdot sk}(\cdot)$	no	no	no	yes	no

□

8.3 DeLP

As to *DeLP*, in order to define a bivalent statement labelling, it is rather natural to assign the label yes to the statements already labelled yes, and the label no to all the other cases. This leads to introduce the bivalent 2-synthesizer $ASJ^{De\cdot yn}$ as follows (see Table 8).

Definition 8.5. The bivalent 2-synthesizer for *DeLP*, denoted as $ASJ^{De\cdot yn}$, is defined as follows:

- $ASJ^{De\cdot yn}(S, U) = \text{yes}$ iff $U \in S$;
- $ASJ^{De\cdot yn}(S, U) = \text{no}$ otherwise.

Table 8: Bivalent labelling for *DeLP*.

prj[ctr] \ prj[supp]	\emptyset	{U}	{D}	{U, D}
\emptyset	no	yes	no	yes
{U}	no	n.a.	no	n.a.
{D}	no	yes	no	yes
{U, D}	no	n.a.	no	n.a.

Clearly $ASJ^{De\cdot yn}$ ensures full coverage, is contrary-insensitive and represents a less refined versions of ASJ^{De} , where $ref(\text{yes}) = \{\text{yes}\}$ and $ref(\text{no}) = \{\text{no}, \text{und}, \text{unk}\}$.

Example 8.3. Applying $ASJ^{De\cdot yn}$ in the context of Example 7.4 we get the following statement justification labelling:

	s1	\neg s1	s2	\neg s2	s3
$L_{ASJ}^{De\cdot yn}(\cdot)$	no	no	no	yes	no

□

8.4 Considerations on Bivalent Labellings

In summary, whilst bivalent labellings can be regarded as the simplest common conceptual basis for statement justification, they are generally less refined than various proposals for statement justification. Indeed, existing approaches differ in the way they specialise and extend the notions of raw justification (yes) and its complement (no).

Simple as they are, bivalent labellings make easy the task of ensuring full coverage, while they tend to be (but not necessarily are) contrary-insensitive. Moreover, in the case of multiple status

argument acceptance evaluation there is a design dilemma, since one has to decide whether the label yes coincides with skeptical acceptance or covers both skeptical and credulous acceptance. We provided a definition for either alternative in $ASPIC^+$ and showed that further intermediate options are possible. In the considered example, an agreement on statement labelling is reached among the different formalisms in the context of the skeptical and credulous perspective since for each statement $S \in \{s1, \neg s1, s2, \neg s2, s3\}$ we get $L_{ASJ}^{A^+.yn-sk}(S) = L_{ASJ}^{A^+.yn-sc}(S) = L_{SSJ}^{AB.yn-sk}(S) = L_{ASJ}^{De.yn}(S)$ and $L_{ASJ}^{A^+.yn-cr}(S) = L_{ASJ}^{AB.yn-cr}(S)$ respectively.

9. Doubt-Tolerant Labellings

Bivalent labellings do not give any particular reason why a statement is labelled no. It may be because it is falsified in some way or just because it lacks sufficient support. As a first step to support this distinction one may assume the set of labels $\Lambda_{S_j}^{dt} = \{\text{yes}, \text{fal}, \text{ni}\}$ where fal indicates that the statement is falsified, while ni captures a less clearcut situation, intermediate between yes and fal. Labellings based on $\Lambda_{S_j}^{dt}$ are called doubt-tolerant because they encompass an explicit distinction between explicit rejection and unresolved doubt, not captured by bivalent labellings.

In this section, we examine how this idea can be integrated in the formalisms reviewed in this paper by defining suitable doubt-tolerant 2-synthesizers. Note that all the doubt-tolerant 2-synthesizers are defined so as to be a refinement of their bivalent counterpart and to be contrary-sensitive.

9.1 $ASPIC^+$

$ASPIC^+$ can be equipped with a skeptically or a credulously oriented option for doubt-tolerant statement labellings. In the former, a statement is labelled yes if it is supported by a skeptically justified argument, fal if one of its contraries is supported by a skeptically justified argument, and ni otherwise. In the latter option, a statement is labelled yes if it is supported by a skeptically or credulously justified argument, fal if it is not supported by a skeptically or credulously justified argument and one of its contraries is supported by a skeptically or credulously justified argument, and ni otherwise. The corresponding 2-synthesizers are introduced in the following definitions and illustrated in Tables 9 and 10.

Definition 9.1. *The skeptical doubt-tolerant 2-synthesizer for $ASPIC^+$, denoted as $ASJ^{A^+.dt-sk}$, is defined as follows*

- $ASJ^{A^+.dt-sk}(S, U) = \text{yes}$ iff $SKJ \in S$;
- $ASJ^{A^+.dt-sk}(S, U) = \text{fal}$ iff $SKJ \in U$;
- $ASJ^{A^+.dt-sk}(S, U) = \text{ni}$ otherwise.

Definition 9.2. *The credulous doubt-tolerant 2-synthesizer for $ASPIC^+$, denoted as $ASJ^{A^+.dt-cr}$, is defined as follows*

- $ASJ^{A^+.dt-cr}(S, U) = \text{yes}$ iff $\{SKJ, CRJ\} \cap S \neq \emptyset$;
- $ASJ^{A^+.dt-cr}(S, U) = \text{fal}$ iff $\{SKJ, CRJ\} \cap S = \emptyset$ and $\{SKJ, CRJ\} \cap U \neq \emptyset$;
- $ASJ^{A^+.dt-cr}(S, U) = \text{ni}$ otherwise.

Clearly both $ASJ^{A^+.dt-sk}$ and $ASJ^{A^+.dt-cr}$ ensure full coverage, are contrary-sensitive, and are a refinement of the corresponding bivalent 2-synthesizers. In particular:

Table 9: Skeptical doubt-tolerant labelling for $ASPIC^+$.

prj[ctr] \ prj[supp]	\emptyset	{NOJ}	{CRJ}	{NOJ, CRJ}	\supseteq {SKJ}
\emptyset	ni	ni	ni	ni	yes
{NOJ}	ni	ni	ni	ni	yes
{CRJ}	ni	ni	ni	ni	n.a.
{NOJ, CRJ}	ni	ni	ni	ni	n.a.
\supseteq {SKJ}	fal	fal	n.a.	n.a.	n.a.

 Table 10: Credulous doubt-tolerant labelling for $ASPIC^+$.

prj[ctr] \ prj[supp]	\emptyset	{NOJ}	{CRJ}	{NOJ, CRJ}	\supseteq {SKJ}
\emptyset	ni	ni	yes	yes	yes
{NOJ}	ni	ni	yes	yes	yes
{CRJ}	fal	fal	yes	yes	n.a.
{NOJ, CRJ}	fal	fal	yes	yes	n.a.
\supseteq {SKJ}	fal	fal	n.a.	n.a.	n.a.

- $ASJ^{A^+ \cdot dt \cdot sk}$ refines $ASJ^{A^+ \cdot yn \cdot sk}$ with $ref(yes) = \{yes\}$ and $ref(no) = \{fal, ni\}$;
- $ASJ^{A^+ \cdot dt \cdot cr}$ refines $ASJ^{A^+ \cdot yn \cdot cr}$ with $ref(yes) = \{yes\}$ and $ref(no) = \{fal, ni\}$.

It can also be noted that neither $ASJ^{A^+ \cdot dt \cdot sk}$ nor $ASJ^{A^+ \cdot dt \cdot cr}$ is a refinement of $ASJ^{A^+ \cdot fc}$ nor vice-versa. This can be regarded as further indication of the variety of possible design options whose investigation appears to be a fertile research area.

Example 9.1. Applying $ASJ^{A^+ \cdot dt \cdot sk}$ in the context of Example 7.1 in the case of complete semantics we get the following statement justification labelling:

$L_{ASJ}^{A^+ \cdot dt \cdot sk}(\cdot)$	s1	$\neg s1$	s2	$\neg s2$	s3
	ni	ni	fal	yes	ni

while applying $ASJ^{A^+ \cdot dt \cdot cr}$ we get

$L_{ASJ}^{A^+ \cdot dt \cdot cr}(\cdot)$	s1	$\neg s1$	s2	$\neg s2$	s3
	yes	yes	fal	yes	ni

It can be observed that the statement justification prescribed by $ASPIC^+$ for complete semantics ($L_{ASJ}^{A^+ \cdot fc}$ in Example 7.1) has some similarity with $L_{ASJ}^{A^+ \cdot dt \cdot sk}$, since both labellings distinguish the statuses of s1 and $\neg s1$ (crj and ni, respectively), s2 (noj and fal), and $\neg s2$ (skj and yes). A difference can be noted in the treatment of s3 which has the same status as s2 (noj) in $L_{ASJ}^{A^+ \cdot fc}$, while it has the same status as s1 and $\neg s1$ (ni) according to $L_{ASJ}^{A^+ \cdot dt \cdot sk}$. While one may remark that equating ignorance to either a status of rejection or of controversy is anyway debatable, this shows that our proposal provides alternative choices not encompassed by the current version of existing formalisms.

Another significant difference which is worth pointing out with respect to Example 7.1 concerns the case of a single-status semantics (like grounded or ideal semantics) where we get:

$L_{ASJ}^{A^+ \cdot dt \cdot sk}(\cdot)$	s1	$\neg s1$	s2	$\neg s2$	s3
	ni	ni	fal	yes	ni

and applying $ASJ^{A^+ \cdot dt \cdot cr}$ yields the same outcome

$L_{ASJ}^{A^+,dt,cr}(\cdot)$	s1	\neg s1	s2	\neg s2	s3
	ni	ni	fal	yes	ni

We remark that, with a single-status semantics, while the statements s1 (or \neg s1) and s2 have the same status (namely noj) in Example 7.1 under the contrary-insensitive statement justification labelling $L_{ASJ}^{A^+,fc}(\cdot)$, with the same semantics the contrary-sensitive statement justification labellings $L_{ASJ}^{A^+,dt,sk}(\cdot)$ and $L_{ASJ}^{A^+,dt,cr}(\cdot)$ allow us to distinguish the status of statements s1 (or \neg s1) and s2. This illustrates the ability of the doubt-tolerant 2-synthesizers to allow finer distinctions about statements when ‘plugged’ into ASPIC⁺ in the case of a single-status semantics while keeping any other aspect of the formalism unchanged. In fact such a finer distinction is achieved in the original formulation of ASPIC⁺ only by resorting to multiple-status semantics, while our proposal shows that single-status semantics are not inherently limited in this respect and can yield analogous distinctions, provided that a suitable statement labelling is adopted. \square

9.2 ABA

We discuss doubt-tolerant labellings in distinct subsections for the two stances of ABA.

9.2.1 ABA (CREDULOUS STANCE)

In the credulous stance, ABA has been reconstructed as an AF MLS using the set of labels $\Lambda_{AJ}^{AB,cr,fc} = \{\text{WIN}, \text{NOWIN}\}$ for argument justification. Taking into account that there can be winning arguments both for a statement and its contraries, also in this case two alternatives (one skeptically oriented and one credulously oriented) for defining a doubt-tolerant statement labelling can be considered (see Table 11 and 12). In a skeptical orientation, a statement is labelled yes if it supported by at least a winning argument and there are no winning arguments for any of its contraries, fal if it is not supported by a winning argument and there are winning arguments for any of its contraries, ni in the other cases. In a credulous orientation, a statement is labelled yes if it supported by at least a winning argument, fal if it is not supported by a winning argument and there are winning arguments for any of its contraries, ni in the other cases. The corresponding 2-synthesizers are introduced in the following definitions.

Definition 9.3. *The skeptical doubt-tolerant 2-synthesizer for ABA in the credulous stance, denoted as $ASJ^{AB,dt,sk}$, is defined as follows*

- $ASJ^{AB,dt,sk}(S, U) = \text{yes}$ iff $\text{WIN} \in S$ and $\text{WIN} \notin U$;
- $ASJ^{AB,dt,sk}(S, U) = \text{fal}$ iff $\text{WIN} \notin S$ and $\text{WIN} \in U$;
- $ASJ^{AB,dt,sk}(S, U) = \text{ni}$ otherwise.

Definition 9.4. *The credulous doubt-tolerant 2-synthesizer for ABA in the credulous stance, denoted as $ASJ^{AB,dt,cr}$, is defined as follows*

- $ASJ^{AB,dt,cr}(S, U) = \text{yes}$ iff $\text{WIN} \in S$;
- $ASJ^{AB,dt,cr}(S, U) = \text{fal}$ iff $\text{WIN} \notin S$ and $\text{WIN} \in U$;
- $ASJ^{AB,dt,cr}(S, U) = \text{ni}$ otherwise.

Tables 11 and 12 illustrate, respectively, the skeptical and credulous doubt-tolerant labelling for ABA depending on $\text{prj}[\text{supp}](\{L_{AJ}^{AB,cr,fc}\}, \varphi)$ and $\text{prj}[\text{cntr}](\{L_{AJ}^{AB,cr,fc}\}, \varphi)$.

Table 11: Skeptical doubt-tolerant labelling for ABA (credulous stance).

$\text{prj}[\text{cntr}] \setminus \text{prj}[\text{supp}]$	\emptyset	{WIN}	{NOWIN}	{WIN, NOWIN}
\emptyset	ni	yes	ni	yes
{WIN}	fal	ni	fal	ni
{NOWIN}	ni	yes	ni	yes
{WIN, NOWIN}	fal	ni	fal	ni

 Table 12: Credulous doubt-tolerant labelling for ABA (credulous stance).

$\text{prj}[\text{cntr}] \setminus \text{prj}[\text{supp}]$	\emptyset	{WIN}	{NOWIN}	{WIN, NOWIN}
\emptyset	ni	yes	ni	yes
{WIN}	fal	yes	fal	yes
{NOWIN}	ni	yes	ni	yes
{WIN, NOWIN}	fal	yes	fal	yes

Clearly both $\text{ASJ}^{\text{AB}\cdot\text{dt}\cdot\text{sk}}$ and $\text{ASJ}^{\text{AB}\cdot\text{dt}\cdot\text{cr}}$ ensure full coverage and are contrary-sensitive. Moreover it is easy to see that $\text{ASJ}^{\text{AB}\cdot\text{dt}\cdot\text{cr}}$ is a refinement of $\text{ASJ}^{\text{AB}\cdot\text{cr}\cdot\text{fc}}$ with $\text{ref}(\text{win}) = \{\text{yes}\}$ and $\text{ref}(\text{nowin}) = \{\text{fal}, \text{ni}\}$.

Example 9.2. Applying $\text{ASJ}^{\text{AB}\cdot\text{dt}\cdot\text{sk}}$ in the context of Example 7.1 in the case of complete semantics we get the following statement justification labelling:

	s1	\neg s1	s2	\neg s2	s3
$L_{\text{ASJ}}^{\text{AB}\cdot\text{dt}\cdot\text{sk}}(\cdot)$	ni	ni	fal	yes	ni

while applying $\text{ASJ}^{\text{AB}\cdot\text{dt}\cdot\text{cr}}$ we get

	s1	\neg s1	s2	\neg s2	s3
$L_{\text{ASJ}}^{\text{AB}\cdot\text{dt}\cdot\text{cr}}(\cdot)$	yes	yes	fal	yes	ni

□

9.2.2 ABA (SKEPTICAL STANCE)

Analogously to what we have remarked in Section 8.2.2 on bivalent labellings, there are various ways to integrate a doubt-tolerant labelling in the skeptical stance of ABA . Among them we consider the 2-synthesizer $\text{SSJ}^{\text{AB}\cdot\text{dt}\cdot\text{sk}}$, which can be regarded as conceptually similar to $\text{ASJ}^{\text{A}\cdot\text{dt}\cdot\text{sk}}$ (see Table 13)

Definition 9.5. The skeptical doubt-tolerant 2-synthesizer for ABA , denoted as $\text{SSJ}^{\text{AB}\cdot\text{dt}\cdot\text{sk}}$, is defined as follows

- $\text{SSJ}^{\text{AB}\cdot\text{dt}\cdot\text{sk}}(S, U) = \text{yes}$ iff $S = \{\text{in}\}$;
- $\text{SSJ}^{\text{AB}\cdot\text{dt}\cdot\text{sk}}(S, U) = \text{fal}$ iff $U = \{\text{in}\}$;
- $\text{SSJ}^{\text{AB}\cdot\text{dt}\cdot\text{sk}}(S, U) = \text{ni}$ otherwise.

Clearly $\text{SSJ}^{\text{AB}\cdot\text{dt}\cdot\text{sk}}$ ensures full coverage, is contrary-sensitive and represents a refinement of $\text{SSJ}^{\text{AB}\cdot\text{yn}\cdot\text{sk}}$ where $\text{ref}(\text{yes}) = \{\text{skj}\}$ and $\text{ref}(\text{no}) = \{\text{fal}, \text{ni}\}$.

Table 13: Doubt-tolerant labelling for ABA (skeptical stance).

$\text{prj}[\text{cntr}] \setminus \text{prj}[\text{supp}]$	$\{\text{nin}\}$	$\{\text{in}\}$	$\{\text{nin}, \text{in}\}$
$\{\text{nin}\}$	ni	yes	ni
$\{\text{in}\}$	fal	n.a.	n.a.
$\{\text{nin}, \text{in}\}$	ni	n.a.	ni

Example 9.3. Applying $\text{SSJ}^{\text{AB*·dt·sk}}$ in the context of Example 7.3 in the case of complete semantics we get the following statement justification labelling:

$L_{\text{SSJ}}^{\text{AB*·dt·sk}}(\cdot)$	s1	\neg s1	s2	\neg s2	s3
	ni	ni	fal	yes	ni

□

9.3 DeLP

Taking into account that $DeLP$ is a single status approach (hence skeptically oriented) and that there can not be two undefeated arguments for contrary statements, a rather straightforward way of defining a doubt-tolerant statement justification labelling emerges, with an operator $\text{ASJ}^{\text{De·dt}}$, conceptually similar to $\text{ASJ}^{\text{A+·dt·sk}}$ (see Table 14).

Definition 9.6. The doubt-tolerant 2-synthesizer for $DeLP$, denoted as $\text{ASJ}^{\text{De·dt}}$, is defined as follows:

- $\text{ASJ}^{\text{De·dt}}(S, U) = \text{yes}$ iff $U \in S$;
- $\text{ASJ}^{\text{De·dt}}(S, U) = \text{fal}$ iff $U \in U$;
- $\text{ASJ}^{\text{De·dt}}(S, U) = \text{ni}$ otherwise.

 Table 14: Doubt-tolerant labelling for $DeLP$.

$\text{prj}[\text{cntr}] \setminus \text{prj}[\text{supp}]$	\emptyset	$\{U\}$	$\{D\}$	$\{U, D\}$
\emptyset	ni	yes	ni	yes
$\{U\}$	fal	n.a.	fal	n.a.
$\{D\}$	ni	yes	ni	yes
$\{U, D\}$	fal	n.a.	fal	n.a.

The synthesizer $\text{ASJ}^{\text{De·dt}}$ ensures full coverage and is contrary-sensitive. It refines $\text{ASJ}^{\text{De·yn}}$ with $\text{ref}(\text{yes}) = \{\text{yes}\}$, $\text{ref}(\text{no}) = \{\text{fal}, \text{ni}\}$. Moreover it represents a less refined versions of ASJ^{De} , where $\text{ref}(\text{yes}) = \{\text{yes}\}$, $\text{ref}(\text{fal}) = \{\text{no}\}$ and $\text{ref}(\text{ni}) = \{\text{und}, \text{unk}\}$.

Example 9.4. Applying $\text{ASJ}^{\text{De·dt}}$ in the context of Example 7.4 we get the following statement justification labelling:

$L_{\text{ASJ}}^{\text{De·dt}}(\cdot)$	s1	\neg s1	s2	\neg s2	s3
	ni	ni	fal	yes	ni

□

9.4 Considerations on Doubt-Tolerant Labellings

Doubt-tolerant labellings refine bivalent labellings, by distinguishing the case of explicit rejection from other cases where a statement is not justified. As such, doubt-tolerant labellings are intrinsically contrary-sensitive.

As to the example, for every statement S in $\{s1, \neg s1, s2, \neg s2, s3\}$ we get:

- $L_{ASJ}^{A^+ \cdot dt \cdot sk}(S) = L_{ASJ}^{AB \cdot dt \cdot sk}(S) = L_{SSJ}^{AB^* \cdot dt \cdot sk}(S) = L_{ASJ}^{De \cdot dt}(S)$, and
- $L_{ASJ}^{A^+ \cdot dt \cdot cr}(S) = L_{ASJ}^{AB \cdot dt \cdot cr}(S)$.

Thus, as it was also the case with the bivalent labelling, in our example the doubt-tolerant labelling provides (separately for the skeptical and the credulous stance) outcomes in agreement for all the considered formalisms, while the results were different with the original definitions of statement justification.

This confirms the merits of a high-level formalism-independent analysis of statement justification and suggests that further more articulated approaches in addition to the basic bivalent and doubt-tolerant labellings are worth exploring, in order to gain further discriminative capabilities. In this regard, a simple development is the (so-called *ignorance-aware*) labelling investigated next.

10. Ignorance-Aware Labellings

In doubt-tolerant labellings, a statement is labelled *ni* if it is not labelled *yes* or *fal*. This may occur due to some lack of knowledge or because the available knowledge carries some undecidedness. To support this distinction, we thus assume the set of labels $\Lambda_{SJ}^{ia} = \{\text{yes}, \text{fal}, \text{unk}, \text{ni}\}$, where the label *unk* stands for ‘unknown’. The labelling is called *ignorance-aware* because it can be obtained as a refinement of the doubt-tolerant labelling, where absence of support is distinguished from conflicting support, a distinction which is not captured by doubt-tolerant and bivalent labellings.

10.1 ASPIC⁺

We obtain a skeptically oriented and a credulously oriented *ignorance-aware* labelling for *ASPIC⁺* as refined variations of Definitions 9.1 and 9.2 respectively, by distinguishing the case of total absence of arguments, which gets the label *unk* (see also Tables 15 and 16).

Definition 10.1. *The skeptical ignorance-aware 2-synthesizer for ASPIC⁺, denoted as $ASJ^{A^+ \cdot ia \cdot sk}$, is defined as follows*

- $ASJ^{A^+ \cdot ia \cdot sk}(S, U) = \text{yes}$ iff $SKJ \in S$;
- $ASJ^{A^+ \cdot ia \cdot sk}(S, U) = \text{fal}$ iff $SKJ \in U$;
- $ASJ^{A^+ \cdot ia \cdot sk}(S, U) = \text{unk}$ iff $S \cup U = \emptyset$;
- $ASJ^{A^+ \cdot ia \cdot sk}(S, U) = \text{ni}$ otherwise.

Definition 10.2. *The credulous ignorance-aware 2-synthesizer for ASPIC⁺, denoted as $ASJ^{A^+ \cdot ia \cdot cr}$, is defined as follows*

- $ASJ^{A^+ \cdot ia \cdot cr}(S, U) = \text{yes}$ iff $\{SKJ, CRJ\} \cap S \neq \emptyset$;
- $ASJ^{A^+ \cdot ia \cdot cr}(S, U) = \text{fal}$ iff $\{SKJ, CRJ\} \cap S = \emptyset$ and $\{SKJ, CRJ\} \cap U \neq \emptyset$;
- $ASJ^{A^+ \cdot ia \cdot cr}(S, U) = \text{unk}$ iff $S \cup U = \emptyset$;
- $ASJ^{A^+ \cdot ia \cdot cr}(S, U) = \text{ni}$ otherwise.

Table 15: Skeptical ignorance-aware labelling for $ASPIC^+$.

$\text{prj}[\text{cntr}] \setminus \text{prj}[\text{supp}]$	\emptyset	$\{\text{NOJ}\}$	$\{\text{CRJ}\}$	$\{\text{NOJ}, \text{CRJ}\}$	$\supseteq \{\text{SKJ}\}$
\emptyset	unk	ni	ni	ni	yes
$\{\text{NOJ}\}$	ni	ni	ni	ni	yes
$\{\text{CRJ}\}$	ni	ni	ni	ni	n.a.
$\{\text{NOJ}, \text{CRJ}\}$	ni	ni	ni	ni	n.a.
$\supseteq \{\text{SKJ}\}$	fal	fal	n.a.	n.a.	n.a.

 Table 16: Credulous ignorance-aware labelling for $ASPIC^+$.

$\text{prj}[\text{cntr}] \setminus \text{prj}[\text{supp}]$	\emptyset	$\{\text{NOJ}\}$	$\{\text{CRJ}\}$	$\{\text{NOJ}, \text{CRJ}\}$	$\supseteq \{\text{SKJ}\}$
\emptyset	unk	ni	yes	yes	yes
$\{\text{NOJ}\}$	ni	ni	yes	yes	yes
$\{\text{CRJ}\}$	fal	fal	yes	yes	n.a.
$\{\text{NOJ}, \text{CRJ}\}$	fal	fal	yes	yes	n.a.
$\supseteq \{\text{SKJ}\}$	fal	fal	n.a.	n.a.	n.a.

Clearly both $ASJ^{A^+ \cdot ia \cdot sk}$ and $ASJ^{A^+ \cdot ia \cdot cr}$ ensure full coverage, are contrary-sensitive, and are a refinement of the corresponding doubt-tolerant 2-synthesizers.

Example 10.1. *The only difference with respect to the doubt-tolerant labellings is that s3 is labelled unk in both the skeptical and credulous version.*

$L_{ASJ}^{A^+ \cdot ia \cdot sk}(\cdot)$	s1	\neg s1	s2	\neg s2	s3
	ni	ni	fal	yes	unk
$L_{ASJ}^{A^+ \cdot ia \cdot cr}(\cdot)$	s1	\neg s1	s2	\neg s2	s3
	yes	yes	fal	yes	unk

□

10.2 ABA

We discuss ignorance-aware labellings in distinct subsections for the two stances of *ABA*.

10.2.1 ABA (CREDULOUS STANCE)

Analogously to the case of $ASPIC^+$, ignorance-aware labellings for *ABA* in the credulous stance are obtained as refined variations of Definitions 9.3 and 9.4 (see also Tables 17 and 18).

Definition 10.3. *The skeptical ignorance-aware 2-synthesizer for ABA in the credulous stance, denoted as $ASJ^{AB \cdot ia \cdot sk}$, is defined as follows*

- $ASJ^{AB \cdot ia \cdot sk}(S, U) = \text{yes}$ iff $\text{WIN} \in S$ and $\text{WIN} \notin U$;
- $ASJ^{AB \cdot ia \cdot sk}(S, U) = \text{fal}$ iff $\text{WIN} \notin S$ and $\text{WIN} \in U$;
- $ASJ^{AB \cdot ia \cdot sk}(S, U) = \text{unk}$ iff $S \cup U = \emptyset$;
- $ASJ^{AB \cdot ia \cdot sk}(S, U) = \text{ni}$ otherwise.

Definition 10.4. *The credulous ignorance-aware 2-synthesizer for ABA in the credulous stance, denoted as $ASJ^{AB\cdot ia\cdot cr}$, is defined as follows*

- $ASJ^{AB\cdot ia\cdot cr}(S, U) = \text{yes}$ iff $\text{WIN} \in S$;
- $ASJ^{AB\cdot ia\cdot cr}(S, U) = \text{fal}$ iff $\text{WIN} \notin S$ and $\text{WIN} \in U$;
- $ASJ^{AB\cdot ia\cdot cr}(S, U) = \text{unk}$ iff $S \cup U = \emptyset$;
- $ASJ^{AB\cdot ia\cdot cr}(S, U) = \text{ni}$ otherwise.

Table 17: Skeptical ignorance-aware labelling for ABA (credulous stance).

prj[ctr] \ prj[supp]	\emptyset	{WIN}	{NOWIN}	{WIN, NOWIN}
\emptyset	unk	yes	ni	yes
{WIN}	fal	ni	fal	ni
{NOWIN}	ni	yes	ni	yes
{WIN, NOWIN}	fal	ni	fal	ni

Table 18: Credulous ignorance-aware labelling for ABA (credulous stance).

prj[ctr] \ prj[supp]	\emptyset	{WIN}	{NOWIN}	{WIN, NOWIN}
\emptyset	unk	yes	ni	yes
{WIN}	fal	yes	fal	yes
{NOWIN}	ni	yes	ni	yes
{WIN, NOWIN}	fal	yes	fal	yes

Clearly both $ASJ^{AB\cdot ia\cdot sk}$ and $ASJ^{AB\cdot ia\cdot cr}$ ensure full coverage, are contrary-sensitive, and are a refinement of the corresponding doubt-tolerant 2-synthesizers.

Example 10.2. *Again, the only difference w.r.t. the doubt-tolerant labellings is that s3 is labelled unk.*

	s1	\neg s1	s2	\neg s2	s3
$L_{ASJ}^{AB\cdot ia\cdot sk}(\cdot)$	ni	ni	fal	yes	unk
	s1	\neg s1	s2	\neg s2	s3
$L_{ASJ}^{AB\cdot ia\cdot cr}(\cdot)$	yes	yes	fal	yes	unk

□

10.2.2 ABA (SKEPTICAL STANCE)

The skeptical stance of ABA, belonging to the SF approach, requires some additional consideration: the absence of any argument concerning a given statement or its contraries needs to be ‘detected’ at the level of the stage Ξ_1^{SF} , where the projection from arguments to statements occurs, and then taken into account at the level of the subsequent stage Ξ_2^{SF} . This can be achieved by revising Definition 7.9.

Definition 10.5 (revised Def. 7.9). *Given a language \mathcal{L} , a set of arguments \mathcal{A} and a labelling in a set \mathcal{L}_{AA} of Λ_{AA}^{IOU} -labellings of \mathcal{A} , for every statement $\varphi \in \mathcal{L}$:*

- φ is supported iff $\exists A \in \text{sup}(\{\varphi\}) : L_{AA}(A) = \text{IN}$;
- φ is ignored iff $\text{sup}(\{\varphi\}) = \text{sup}(\text{cnt}(\varphi)) = \emptyset$;
- φ is not supported otherwise.

Letting $\Lambda_{SA}^{\text{AB}\cdot\text{sk}'} = \{\text{in}, \text{nin}, \text{ign}\}$, Definition 10.5 corresponds to a stage $\Xi_1^{\text{SF}} = \langle \mathcal{L}, \Lambda_{SA}^{\text{AB}\cdot\text{sk}'}, \text{pgen}_{SA} \rangle$ where $\text{pgen}_{SA} = \text{pgen}[\text{SC}, \text{SA}^{\text{AB}\cdot\text{sk}}]$ and $\text{SA}^{\text{AB}\cdot\text{sk}} : \text{pow}(\Lambda_{AA}^{\text{IOU}}) \times \text{pow}(\Lambda_{AA}^{\text{IOU}}) \rightarrow \Lambda_{SA}^{\text{AB}\cdot\text{sk}}$ is defined, for any $S, U \in \text{pow}(\Lambda_{AA}^{\text{IOU}})$, as

- $\text{SA}^{\text{AB}\cdot\text{sk}}(S, U) = \text{in}$ iff $\text{IN} \in S$;
- $\text{SA}^{\text{AB}\cdot\text{sk}}(S, U) = \text{ign}$ iff $S \cup U = \emptyset$;
- $\text{SA}^{\text{AB}\cdot\text{sk}}(S, U) = \text{nin}$ otherwise.

On this basis, we introduce in Definition 10.6 an ignorance-aware 2-synthesizer for ABA in the skeptical stance denoted as $\text{SSJ}^{\text{AB}\cdot\text{ia}\cdot\text{sk}}$ (see also Table 19). Note in particular that $\text{dom}(\text{SSJ}^{\text{AB}\cdot\text{ia}\cdot\text{sk}}) = (\{\{\text{nin}\}, \{\text{in}\}, \{\text{in}, \text{nin}\}\} \times \{\{\text{ign}\}, \{\text{nin}\}, \{\text{in}\}, \{\text{in}, \text{nin}\}\} \cup \{(\{\text{ign}\}, \{\text{ign}\}), (\{\text{ign}\}, \{\text{nin}\})\}) \setminus \{(\{\text{in}\}, \{\text{in}\}), (\{\text{in}\}, \{\text{ni}, \text{in}\}), (\{\text{ni}, \text{in}\}, \{\text{in}\})\}$. We can remark here that if a statement φ is labelled ign in one acceptance labelling, it must be labelled ign in every other acceptance labelling. Hence it cannot be the case that $\text{prj}[\text{supp}](\mathcal{L}_{SA}^{\text{AB}\cdot\text{sk}'}, \varphi) \supseteq \{\text{ign}\}$, and this also holds for $\text{prj}[\text{cntr}](\mathcal{L}_{SA}^{\text{AB}\cdot\text{sk}'}, \varphi)$ given that φ has at most one contrary. Moreover, when $\text{prj}[\text{supp}](\mathcal{L}_{SA}^{\text{AB}\cdot\text{sk}'}, \varphi) = \{\text{ign}\}$, by Definition 10.5, it holds $\text{sup}(\text{cnt}(\varphi)) = \emptyset$, hence it cannot be the case that $\text{in} \in \text{prj}[\text{cntr}](\mathcal{L}_{SA}^{\text{AB}\cdot\text{sk}'}, \varphi)$.

Definition 10.6. *The skeptical ignorance-aware 2-synthesizer for ABA , denoted as $\text{SSJ}^{\text{AB}\cdot\text{ia}\cdot\text{sk}}$, is defined as follows*

- $\text{SSJ}^{\text{AB}\cdot\text{ia}\cdot\text{sk}}(S, U) = \text{yes}$ iff $S = \{\text{in}\}$;
- $\text{SSJ}^{\text{AB}\cdot\text{ia}\cdot\text{sk}}(S, U) = \text{fal}$ iff $U = \{\text{in}\}$;
- $\text{SSJ}^{\text{AB}\cdot\text{ia}\cdot\text{sk}}(S, U) = \text{unk}$ iff $S \cup U = \{\text{ign}\}$;
- $\text{SSJ}^{\text{AB}\cdot\text{ia}\cdot\text{sk}}(S, U) = \text{ni}$ otherwise.

Table 19: Ignorance-aware labelling for ABA (skeptical stance).

$\text{prj}[\text{cntr}] \setminus \text{prj}[\text{supp}]$	$\{\text{ign}\}$	$\{\text{nin}\}$	$\{\text{in}\}$	$\{\text{nin}, \text{in}\}$
$\{\text{ign}\}$	unk	ni	yes	ni
$\{\text{nin}\}$	ni	ni	yes	ni
$\{\text{in}\}$	n.a.	fal	n.a.	n.a.
$\{\text{nin}, \text{in}\}$	n.a.	ni	n.a.	ni

Clearly $\text{SSJ}^{\text{AB}\cdot\text{ia}\cdot\text{sk}}$ ensures full coverage and it is contrary-sensitive.

Example 10.3. *Referring to Example 7.3, the only difference in each of the three statement acceptance labellings is that s3 is labelled ign. As a consequence, concerning statement justification the only difference w.r.t. the doubt-tolerant labellings is that s3 is labelled unk.*

	s1	\neg s1	s2	\neg s2	s3
$L_{\text{SSJ}}^{\text{AB}\cdot\text{ia}\cdot\text{sk}}(\cdot)$	ni	ni	fal	yes	unk

□

10.3 DeLP

As to *DeLP*, its original definition of statement justification is ignorance-aware (see Definition 7.11), modulo two label names: *no* for *fa*, and *und* for *ni*. Thus no further specific discussion is needed.

Example 10.4. *Again, the only difference in the example w.r.t. the doubt-tolerant labelling is that s3 is labelled unk.*

	s1	\neg s1	s2	\neg s2	s3
$L_{ASJ}^{De-ia}(\cdot)$	ni	ni	fa	yes	unk

□

10.4 Considerations on Ignorance-Aware Labellings

In the context of the AF approach, ignorance-aware labellings provide a simple, yet useful, refinement of doubt-tolerant labellings by adding a separate treatment of the case of total lack of knowledge for a statement, while keeping the other desirable properties and the inter-formalism agreement in the considered example, as commented in previous sections. In the context of the SF approach, exemplified by the skeptical stance of *ABA*, the case of total lack of knowledge needs to be ‘captured earlier’, i.e. at the stage of argument acceptance, thus requiring a more extensive adjustment. This provides another concrete illustration of the structural difference between the two approaches.

11. Discussion of Related Works

The idea of modelling argumentation formalisms as multi-labelling systems and of extending tunability to various phases of the argumentation process has not been considered up to now in the literature, where tunability is usually encompassed only at the level of argumentation semantics. Thus, strictly speaking, there are no direct terms of comparison for the overall approach we are proposing and for the analysis of its main properties we have carried out. However, various aspects of MLSs for modelling argumentation formalisms can be discussed with respect to the literature, especially in relation to the general distinction between the AF and the SF approach and the various stages therein.

As to the AF and SF approach, it is worth recalling that two alternative approaches for assessing the justification status of conclusions in argument-based reasoning have been studied as part of a debate about defining an appropriate notion of skeptical acceptance (Horty, 2002; Prakken, 2002; Makinson & Schlechta, 1991). In the context of this debate¹³, it is assumed, as usual, that an intelligent agent produces a set of alternative reasonable positions, called extensions, each consisting of a set of arguments with relevant conclusions, and then the agent faces the problem of deciding which conclusions to regard as justified. In particular, the agent may adopt a skeptical policy according to which conclusions are justified if, roughly speaking, they get support by every alternative and belong to the *intersection* of the alternatives. As discussed by Horty (2002, page 59) (see also the similar discussion by Makinson & Schlechta, 1991 in the context of inheritance networks), there

13. Actually this debate is not limited to argument-based reasoning but spans over various forms of defeasible reasoning. We limit our discussion to the aspects which are directly relevant to the present paper while referring the reader to the cited literature sources for a more extensive view of the subject.

are actually two approaches to implement this rough idea. One approach computes first the intersection of justified arguments at the level of extensions, i.e. the set of arguments that are present in all extensions, then the conclusions of these arguments are viewed as skeptically justified. The second approach derives first the conclusions at the level of individual extensions, i.e. identifies a set of supported conclusions for each extension, and then computes the intersection of these sets of conclusions to obtain the set of conclusions which can be deemed skeptically justified. Clearly, the first notion of skeptical justification can be regarded as an instance of the AF approach, while the second notion as an instance of the SF approach. Moreover they can be put in correspondence with the notion of skeptical justification in *ASPIC*⁺ and with the skeptical stance of *ABA*, respectively.

The two competing notions of skeptical justification gave rise to a vivid discussion, in particular because they differ in the treatment of the so-called floating conclusions, namely those conclusions that are supported by at least one argument in each extension, but not by the same argument in all extensions. Floating conclusions are not justified according to the first notion of skeptical justification, while they are according to the second one. This led Makinson and Schlechta (1991) to argue that the first approach, called directly skeptical, has to be regarded as problematic as it is intrinsically unable to deal with floating conclusions (and has also difficulties with another related phenomenon called zombi paths). To address this kind of criticism Horty (2002) proposed various reasoning examples where it is intuitive to regard floating conclusions as not justified, thus in turn questioning the validity of the second notion of skeptical justification. Prakken (2002) discussed Horty's examples by suggesting that the controversy can be solved by a more appropriate modelling of the underlying reasoning activity, in particular by the explicit consideration of additional information (e.g. some missing default rules or assumptions) left implicit in Horty's discussion. This would ensure that the desired intuitive results are obtained in Horty's examples, while still regarding floating conclusions as skeptically justified in other examples.

Our distinction between the AF and SF approaches provides a well-founded and more general theoretical framework where the above debate about different notions of skeptical justification (and, in a wider perspective, the analysis of alternative notions of justification) can be placed. In particular, the expressiveness comparison developed in Section 5 generalises the specific observations about the different behaviour of the two approaches presented, for specific cases, in the papers mentioned above. The comparison also provides a definite negative answer to any claim of prevalence of either choice, by showing that neither of the two approaches subsumes the other in the general case.

We in particular agree with a remark given by Horty (2002, page 68) where it is acknowledged that part of the conundrum concerning skeptical acceptance is related to the adoption of a binary notion of acceptance (a bivalent labelling using the terminology of the present paper). Horty then admits that an alternative approach to solve the problem would consist in overcoming the binary view and 'placing statements into several categories, depending on the degree to which they are supported by a set of premises, with floating conclusions then classified, not necessarily as unsupported, but perhaps only as less firmly supported than statements that are justified by the same argument in every extension.'

Somehow similar considerations are also drawn by Prakken and Vreeswijk (2001), where an extensive discussion of the issue of floating conclusions is developed and it is observed in particular that 'some of the different consequence notions are not mutually exclusive but can be used in parallel, as capturing different senses in which belief in a proposition can be supported by a body of information'. Furthermore it is remarked that 'in general the existence of different definitions is not a problem for, but a feature of the field of defeasible argumentation'.

This is exactly the direction pursued by this paper, which provides a formal setting supporting a systematic study of alternative, possibly more or less refined, notions of justification embeddable and tunable in a variety of argumentation formalisms.

It is worth noting that some similar intuition can also be identified in a recent work (Caminada, Sá, Alcântara, & Dvorák, 2015), devoted to the study of the relationships between logic programming semantics and argumentation semantics. This work uses the set of complete labellings as a main reference for argumentation semantics and relies on the characterisation of the set of labellings prescribed by other semantics in terms of some maximisation or minimisation condition with respect to complete labellings (e.g. preferred labellings are complete labellings where the set of arguments labelled in is maximal w.r.t. inclusion, semi-stable labellings are complete labellings where the set of arguments labelled und is minimal and so on).

It is then observed that two ways to extend these labellings to statements are possible. The first one consists in first restricting the set of labellings by applying the maximisation/minimisation conditions at the level of arguments and then projecting each resulting labelling on the conclusions (selecting the ‘best’ label according to the order in $>$ und $>$ out if a statement is the conclusion of more than one argument). The second one consists in first projecting all complete labellings on the argument conclusions (again selecting the ‘best’ label when needed) and then restricting the resulting set of labellings by applying the maximisation/minimisation conditions at the level of statements. Clearly, the first way bears some similarity with the AF approach while the second one with the SF approach. It is shown by Caminada et al. (2015) that in most cases applying the same maximisation/minimisation condition either at the argument level or at the statement level gives the same result, but, as one may expect, this does not hold in general. In particular, it turns out that minimising the und label at the argument level does not give the same result as minimising it at the statement level. Due to different interests, the issue of deriving a synthetic justification assessment for statements is not considered by Caminada et al. (2015), and therefore a full comparison with our approach cannot be directly drawn. Exploring in more detail the possible connections among MLSs, logic programming and argumentation semantics represents an interesting subject for future work.

Let us now turn to a discussion of the stages we have handled in the context of the AF and SF approaches.

As to the stage of argument acceptance, common to the AF and SF approaches, Dung’s abstract argumentation semantics, expressible in terms of Λ_{AA}^{OU} -labellings is widely adopted as an essential element of other formalisms, as in the cases of *ASPIC*⁺ and *ABA*, and is among the most investigated subjects in formal argumentation literature. Notable exceptions to this largely dominating position are anyway available. As we have seen in the paper, *DeLP* provides an example of a full formalism not relying on Dung’s model. Further, it is worth mentioning that a four-valued labelling for abstract argumentation has been proposed by Jakobovits and Vermeir (1999), where a status of ‘don’t care’ is added to the three statuses of accepted, rejected, and undecided.

As to the notion of argument justification encompassed by the AF approach, the simple traditional distinction between credulous and skeptical justification (implicitly completed by the notion of no-justification) is by far the one most often considered in the literature and, in fact, the problems of checking credulous and skeptical justification of an argument are among the standard problems in the analysis of computational complexity in relation to argumentation semantics (see e.g. Dunne & Wooldridge, 2009). It has to be mentioned however that more articulated notions of argument justification have also been proposed in the literature. For instance Cayrol and Lagasquie-Schiex

(2003) examined a classification encompassing four states, namely: uni-accepted (corresponding to traditional skeptical justification), not-accepted (corresponding to traditional no-justification), cleanly-accepted and only-exi-accepted. The two latter states provide a refinement of the traditional notion of credulous justification: an argument is cleanly accepted if it is credulously accepted and is not attacked by any credulously accepted argument, while an argument is only-exi-accepted if it is credulously accepted and is attacked by a credulously accepted argument. Baroni et al. (2004), in the context of the study of skepticism relations among argumentation semantics, seven argument justification states were identified. Essentially, these states are defined in terms of the set of the labels that are assigned to an argument in the various acceptance labellings considered in the argument acceptance stage. For instance the status called **JS1** by Baroni et al. (2004) applies to arguments which get the label IN in all acceptance labellings (and so corresponds to traditional skeptical justification). The status **JS2** concerns arguments which get the label OUT in all labellings (and so is a special case of traditional no-justification), **JS3** concerns arguments which get the label UN in all labellings (another subcase of no-justification), **JS4** concerns arguments which are labelled UN and OUT, but never IN, in the acceptance labellings: thus altogether **JS2**, **JS3**, and **JS4** refine the notion of no-justification. Finally **JS5**, **JS6**, and **JS7** refine the traditional notion of credulous justification by considering the cases where an argument is labelled IN in some acceptance labellings but gets other labels in some other acceptance labellings (also UN in **JS5**, also OUT in **JS6**, both UN and OUT in **JS7**). The actual possibility of occurrence of the 7 justification statuses is proved by Baroni et al. (2004) with reference to Dung's preferred semantics. Essentially the same kind of argument justification labelling has been independently proposed by Wu and Caminada (2010) where argument justification statuses are directly defined as subsets of the set of labels $\{IN, OUT, UN\}$, which in principle gives rise to 8 justification statuses. The additional status with respect to the seven others treated by Baroni et al. (2004) corresponds to the empty set of labels, i.e. to the case where no argument acceptance labelling is produced (in Dung's abstract argumentation this can only happen when stable semantics is applied). In the work by Wu and Caminada (2010) attention is focused on the case of complete semantics, under which only 6 argument justification states are possible, the ones corresponding to the empty set and to $\{IN, OUT\}$ not being achievable. Dvorák (2011) extended the analysis showing which of the 8 justification states may actually occur under the main argumentation semantics in the literature. In particular the 7 statuses considered by Baroni et al. (2004) are shown to possibly appear under preferred, semi-stable, stage, and resolution-based grounded semantics. Moreover, a complexity analysis of the problem of deciding whether a given argument has a given justification status for a given semantics has been carried out (Dvorák, 2011).

Some correspondences for the notion of statement acceptance encompassed by the SF approach can also be found in the literature. The conversion of argument labellings to conclusion labellings considered by Wu and Caminada (2010, Section 6) and Caminada et al. (2015, Section 3.3) corresponds, in fact, to a simple statement acceptance stage where the same set of labels Λ_{AA}^{IOU} is used for arguments and statements and the max operator is used as synthesizer. It can be observed that the same conversion is used implicitly by Dimopoulos et al. (2002) as an intermediate step towards the definition of various computational problems concerning statement justification.

Let us now turn to statement justification, which, though obtained in different ways, is the common final step for the AF and the SF approach. Several considerations on this point have already been drawn in sections 7, 8, 9 and 10 and will not be repeated here. We only add that the most articulated notion of statement justification we are aware of, encompassing eight statuses, has been proposed by Wu and Caminada (2010). In short, as already mentioned above, first from each ar-

gument acceptance labelling a conclusion acceptance labelling is derived, thus obtaining a set of conclusion acceptance labelling based on the set of labels Λ_{AA}^{OU} . Then, the final justification state of each conclusion is represented by the set of labels it collectively gets from the set of conclusion acceptance labelling. For instance, if there are two conclusion acceptance labellings and a conclusion gets the label *in* in one, and the label *und* in the other one, the final justification state will be represented by the set $\{\textit{in}, \textit{und}\}$. It is shown by Wu and Caminada (2010) that, under complete semantics, this articulated notion of justification allows one to distinguish the state of floating conclusions ($\{\textit{in}, \textit{und}\}$) from the state of conclusions supported by skeptically justified arguments ($\{\textit{in}\}$) and of credulously justified but non floating conclusions ($\{\textit{in}, \textit{out}\}$ or $\{\textit{in}, \textit{out}, \textit{und}\}$). These distinctions can be useful to solve the dilemmas posed by some problematic examples raised in the above mentioned debate about floating conclusions (Horty, 2002; Prakken, 2002), by pointing out that these conclusions feature an intermediate status which can be regarded as sufficient or not to support a decision depending on the proof standard adopted in a given context.

Altogether, the works reviewed above show a significant interest in the literature towards investigating notions of argument and statement justification which are more expressive than the simpler ‘traditional’ ones and allow one to capture more articulated forms of reasoning. Our work addresses the need of a general formal framework supporting a full development of these heterogeneous but related research lines, and it provides a set of reference properties, including in particular the relation of refinement, which are essential for a systematic analysis and comparison of different approaches.

12. Conclusions

Motivated by possible losses of sensitivity or expressiveness arising in various argument-based formalisms and to facilitate the comparison of such formalisms in the evaluation of statement statuses, we have proposed a framework for multi-labelling systems which model reasoning processes as a sequence of evaluation steps.

By using the framework of multi-labelling systems, general abstract properties can be defined in order to analyse and compare on a principled basis various approaches to the evaluation of argument and statement statuses. Accordingly, it has been shown that the argument and statement evaluation encompassed by several well-known argumentation formalisms can be modelled and compared through multi-labelling systems.

The proposed framework has allowed us to identify, in the context of argument-based reasoning, two alternative approaches for the derivation of statement justification labellings from argument acceptance labellings, namely the *argument-focused* and the *statement-focused* approach. They have been formalised as classes of multi-labelling systems and some general results about their expressiveness have been obtained.

The formalisation provides the basis for a systematic analysis and comparison of these particular formalisms and, possibly more interestingly, it enables the investigation of alternative and ‘tunable’ notions of statement justification, featuring different levels of refinement, in the context of the same formalism. It has been demonstrated that all these alternative notions are able to reconcile the differences in statement justification arising among different formalisms in some basic paradigmatic patterns of reasoning and that in some cases they can overcome the limitations of some current definitions, e.g. by enabling finer statement justification distinctions when using a single-status semantics.

An immediate future work direction concerns extending the use of multi-labelling systems to gradual argumentation formalisms (Baroni, Rago, & Toni, 2019). In particular it can be observed that most studies concerning gradual argumentation have been focused up to now on gradual evaluation at the stage of argument acceptance, while not considering explicitly gradual statement evaluation: multi-labelling systems appear to be a suitable conceptual and formal tool to fill this gap, developing proposals along the same lines considered in this paper for traditional qualitative labellings. Another interesting development concerns investigating the use of multi-labelling systems to model other forms of reasoning in argumentation contexts, like judgement aggregation (Caminada & Pigozzi, 2011) and labelling approaches to probabilistic argumentation (Riveret, Baroni, Gao, Governatori, Rotolo, & Sartor, 2018). On a wider horizon, it would be also interesting to further develop the theoretical analysis of multi-labelling systems as a generic formal tool and to explore its applicability outside the area of argumentation formalisms.

Appendix A. Abstract Argumentation

This appendix provides a quick recall of the basic notions of abstract argumentation and its semantics. The reader may refer to the original Dung's (1995) paper and the survey by Baroni et al. (2011) for more details. Abstract argumentation theory is based on the simple notion of argumentation framework.

Definition A.1. *An argumentation framework is a pair $AF = \langle \mathcal{A}, \rightarrow \rangle$ where \mathcal{A} is a set of arguments and $\rightarrow \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation on it, called attack relation.*

The fact that $(A, B) \in \rightarrow$, also denoted as $A \rightarrow B$, indicates that the argument A attacks the argument B . For conciseness, given a set of arguments $S \subseteq \mathcal{A}$ we write $S \rightarrow B$ to indicate that there exists $A \in S$ such that $A \rightarrow B$.

The simple idea behind this formalism is to capture only the fact that arguments may attack each other, while abstracting away any other property or detail, as this is sufficient for a study of argument acceptability.

The notions of conflict-freeness, defense and admissibility, recalled in Definition A.2 lie at the heart of Dung's *argumentation semantics*.

Definition A.2. *Given an argumentation framework $AF = \langle \mathcal{A}, \rightarrow \rangle$:*

- *a set $S \subseteq \mathcal{A}$ is conflict-free if $\nexists A, B \in S$ s.t. $A \rightarrow B$;*
- *an argument $A \in \mathcal{A}$ is acceptable with respect to a set $S \subseteq \mathcal{A}$ (or, equivalently, is defended by S) if $\forall B \in \mathcal{A}$ s.t. $B \rightarrow A$, $S \rightarrow B$;*
- *the function $F_{AF} : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ such that $F_{AF}(S) = \{A \mid A \text{ is acceptable w.r.t. } S\}$ is called the characteristic function of AF ;*
- *a set $S \subseteq \mathcal{A}$ is admissible if S is conflict-free and every element of S is acceptable with respect to S , i.e. $S \subseteq F_{AF}(S)$.*

In Dung's original proposal, an argumentation semantics identifies for any argumentation framework a set of *extensions*, where each extension is a set of arguments which are 'collectively acceptable', or, in other words, are able to survive together the conflict represented by the attack relation. In general, a semantics may prescribe several extensions for an argumentation framework (such a semantics is said to be *multiple-status*) but some semantics are defined in a way so as to ensure that

exactly one extension is identified for each argumentation framework (such a semantics is said to be *single-status*). The following definition recalls the semantics originally introduced¹⁴ by Dung (1995), namely *complete*, *grounded*, *preferred*, and *stable* semantics. All of them are multiple-status, but grounded semantics which is single-status. Further, stable semantics is the only one which, in some cases, may fail to prescribe any extension.

Definition A.3. Given an argumentation framework $AF = \langle \mathcal{A}, \rightarrow \rangle$:

- a set $S \subseteq \mathcal{A}$ is a complete extension if S is admissible and $\forall A \in \mathcal{A}$ s.t. A is acceptable w.r.t. S , $A \in S$ (i.e. $S = F_{AF}(S)$);
- a set $S \subseteq \mathcal{A}$ is the grounded extension if S is the least (w.r.t. set inclusion) fixed point¹⁵ of the characteristic function F_{AF} ;
- a set $S \subseteq \mathcal{A}$ is a preferred extension if S is a maximal (w.r.t. set inclusion) admissible set;
- a set $S \subseteq \mathcal{A}$ is a stable extension if S is conflict-free and $\forall A \in \mathcal{A} \setminus S$, $S \rightarrow A$.

Abstract argumentation semantics can be equivalently formulated in terms of labellings (Caminada & Gabbay, 2009; Baroni et al., 2011) based on the set of labels $\Lambda_{AA}^{IOU} = \{\text{IN}, \text{OUT}, \text{UN}\}$. In short, this formalisation is based on the fact that any extension S induces a partition of the set of arguments into three subsets and hence any extension can be put in correspondence with a Λ_{AA}^{IOU} -labelling L , as follows:

- the arguments belonging to the extension are labelled IN, formally: $\forall A \in S$ $L(A) = \text{IN}$;
- the arguments attacked by the extension are labelled OUT, formally: $\forall A \in \mathcal{A}$ such that $S \rightarrow A$ $L(A) = \text{OUT}$;
- the remaining arguments (not belonging to and not attacked by the extension) are labelled UN, formally: $\forall A \in \mathcal{A}$ such that $A \notin S$ and $S \not\rightarrow A$ $L(A) = \text{UN}$.

References

- Amgoud, L., Cayrol, C., Lagasquie-Schiex, M., & Livet, P. (2008). On bipolarity in argumentation frameworks. *Int. J. of Intelligent Systems*, 23(10), 1062–1093.
- Baroni, P., Caminada, M., & Giacomin, M. (2011). An introduction to argumentation semantics. *Knowledge Eng. Review*, 26(4), 365–410.
- Baroni, P., Giacomin, M., & Guida, G. (2004). Towards a formalization of skepticism in extension-based argumentation semantics. In *Proc. of the 4th Workshop on Computational Models of Natural Argument, CMNA 2004*, pp. 47–52.
- Baroni, P., Giacomin, M., & Liao, B. (2015). Dealing with generic contrariness in structured argumentation. In *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence, IJCAI 2015*, pp. 2727–2733.
- Baroni, P., Governatori, G., Lam, H.-P., & Riveret, R. (2016a). On the justification of statements in argumentation-based reasoning. In *Proc. of the 15th Int. Conf. on Principles of Knowledge Representation and Reasoning*, pp. 521–524.

14. Other semantics introduced subsequently in the literature are surveyed by Baroni et al. (2011).

15. It is shown in Theorem 25 of (Dung, 1995) that the grounded extension can be equivalently characterized as the least complete extension.

- Baroni, P., Governatori, G., & Riveret, R. (2016b). On labelling statements in multi-labelling argumentation. In *Proc. of the 22nd European Conference on Artificial Intelligence*, Vol. 285, pp. 489–497. IOS Press.
- Baroni, P., Rago, A., & Toni, F. (2019). From fine-grained properties to broad principles for gradual argumentation: A principled spectrum. *Int. J. of Approximate Reasoning*, 105, 252–286.
- Brewka, G., & Woltran, S. (2010). Abstract dialectical frameworks. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2010)*, pp. 102–111.
- Caminada, M., & Amgoud, L. (2007). On the evaluation of argumentation formalisms. *Artif. Intell.*, 171(5-6), 286–310.
- Caminada, M., & Pigozzi, G. (2011). On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1), 64–102.
- Caminada, M., Sá, S., Alcântara, J., & Dvorák, W. (2015). On the equivalence between logic programming semantics and argumentation semantics. *Int. J. Approx. Reasoning*, 58, 87–111.
- Caminada, M. W. A., & Gabbay, D. M. (2009). A logical account of formal argumentation. *Studia Logica*, 93(2-3), 109–145.
- Cayrol, C., & Lagasque-Schiex, M.-C. (2003). Gradual acceptability in argumentation systems. In *Proc. of the 3rd Int. Workshop on Computational Models of Natural Argument (CMNA 2003)*, pp. 55–58.
- Dimopoulos, Y., Nebel, B., & Toni, F. (2002). On the computational complexity of assumption-based argumentation for default reasoning. *Artif. Intell.*, 141(1-2), 57–78.
- Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2), 321–358.
- Dunne, P. E., & Wooldridge, M. (2009). Complexity of abstract argumentation. In Simari, G. R., & Rahwan, I. (Eds.), *Argumentation in Artificial Intelligence*, pp. 85–104. Springer.
- Dvorák, W. (2011). On the complexity of computing the justification status of an argument. In *Theory and Applications of Formal Argumentation - First Int. Workshop, TAFE 2011. Revised Selected Papers*, Vol. 7132 of *Lecture Notes in Computer Science*, pp. 32–49. Springer.
- Ellerman, D. (2010). The logic of partitions: Introduction to the dual of the logic of subsets. *Review of Symbolic Logic*, 3(2), 287–350.
- García, A. J., & Simari, G. R. (2014). Defeasible logic programming: DeLP servers, contextual queries, and explanations for answers. *Argument & Computation*, 5(1), 63 – 88.
- García, A. J., & Simari, G. R. (2004). Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4(1-2), 95–138.
- Horty, J. F. (2002). Skepticism and floating conclusions. *Artif. Intell.*, 135(1-2), 55–72.
- Jakovovits, H., & Vermeir, D. (1999). Robust semantics for argumentation frameworks. *J. Log. Comput.*, 9(2), 215–261.
- Makinson, D., & Schlechta, K. (1991). Floating conclusions and zombie paths: Two deep difficulties in the "directly skeptical" approach to defeasible inheritance nets. *Artif. Intell.*, 48(2), 199–209.

- Modgil, S., & Prakken, H. (2013). A general account of argumentation with preferences. *Artif. Intell.*, 195, 361 – 397.
- Modgil, S., & Prakken, H. (2014). The ASPIC⁺ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1), 31 – 62.
- Prakken, H. (2010). An abstract framework for argumentation with structured arguments. *Argument & Computation*, 1(2), 93–124.
- Prakken, H. (2002). Intuitions and the modelling of defeasible reasoning: some case studies. In *Proc. of the 9th Int. Workshop on Non-Monotonic Reasoning (NMR 2002)*, pp. 91–102.
- Prakken, H., & Vreeswijk, G. A. W. (2001). Logics for defeasible argumentation. In Gabbay, D. M., & Guenther, F. (Eds.), *Handbook of Philosophical Logic*, vol. 4, pp. 219–318. Springer.
- Riveret, R., Baroni, P., Gao, Y., Governatori, G., Rotolo, A., & Sartor, G. (2018). A labelling framework for probabilistic argumentation. *Annals of Mathematics and Artificial Intelligence*, 83(1), 21–71.
- Toni, F. (2014). A tutorial on assumption-based argumentation. *Argument & Computation*, 5(1), 89 – 117.
- Wu, Y., & Caminada, M. (2010). A labelling-based justification status of arguments. *Studies in Logic*, 3(4), 12–29.