Constructing Local Functions to Decompose Argumentation Semantics: Preliminary Results

Pietro Baroni¹, Federico Cerutti^{1,2} and Massimiliano Giacomin¹

¹Department of Information Engineering (University of Brescia), Italy ²Cardiff University, UK

Abstract

This paper builds on a general model for the investigation on decomposability in abstract argumentation, i.e. the possibility of determining the labellings prescribed by a semantics based on evaluations of local functions in subframeworks. A constructive procedure for identifying local functions is devised, able to enforce decomposability whenever the semantics is decomposable. In particular, two kinds of local functions are identified, and some of their properties are analyzed.

Keywords

Dung framework, Argumentation Semantics, Decomposability

1. Introduction

Dung's model provides an abstract account of argumentation where arguments are simply represented as nodes of a directed graph, called *argumentation framework*, and where the graph's edges represent binary attacks between them [1]. This formalism is able to capture several approaches in nonmonotonic reasoning and structured argumentation. Its importance lies in the formal methods, called *argumentation semantics*, to determine the justification status of a set of (typically conflicting) arguments, and thus the status of the relevant conclusions in structured instances of the abstract model.

While the original definitions of argumentation semantics evaluate arguments at a global level, referring to the whole argumentation framework, in recent years attention has been devoted to semantics definition in a modular fashion, i.e. determining the semantics outcome based on local evaluations in subframeworks [2, 3, 4]. This is motivated by the possibility of saving computation time [5, 6, 7, 8], and by investigations concerning various equivalence relations [9, 10, 11], summarizing argumentation frameworks [12], and combining different argumentation semantics [13, 14, 15].

In a previous paper [16] a general model has been devised for studying the decomposability of argumentation semantics in Dung's abstract argumentation setting. This model does not assume any constraint on the way an argumentation framework is partitioned into subframeworks, and encompasses all possible kinds of local information available for the local computations,

Al³ 2021: 5th Workshop on Advances in Argumentation in Artificial Intelligence

[☆] pietro.baroni@unibs.it (P. Baroni); federico.cerutti@unibs.it (F. Cerutti); massimiliano.giacomin@unibs.it (M. Giacomin)

^{🕑 0000-0001-5439-9561 (}P. Baroni); 0000-0003-0755-0358 (F. Cerutti); 0000-0003-4771-4265 (M. Giacomin)

^{© 0 2021} Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

under some mild constraints. On this basis, the property of *decomposability* of argumentation semantics has been introduced concerning the correspondences between semantics outcome at global and local level. A semantics is decomposable if, given a partition of an argumentation framework into a set of sub-frameworks, the outcomes produced by the semantics can be obtained as a combination of the outcomes produced by a *local function* applied separately on each sub-framework, and vice versa.

A central issue is therefore how to determine a local function for a given argumentation semantics, able to guarantee decomposability if the semantics and the local information exploited make it possible. In this regard, the paper aims at providing some general results that do not rely on specific semantics definitions. To this purpose, it introduces a constructive procedure based on the selection of argumentation frameworks, where the output of the local function can be determined by applying the semantics at hand. This model is shown general enough to encompass two kinds of local functions, both of them enforcing decomposability if possible.

After some background on Dung's model provided in Section 2, Section 3 describes the general model for decomposability introduced in [16]. The constructive procedure is then introduced in Section 4. Section 5 exploits this procedure to devise the canonical local function for a semantics, which enforces decomposability whenever possible. Section 6 identifies an alternative 'light' local function, which achieves the same result under some constraints concerning in particular the local information available. Section 7 concludes the paper. The proofs of the results already published in [16] are not reported, while all the proofs in this paper concern novel results.

2. Background

We follow the traditional definition of argumentation framework [1] and define its restriction to a subset of arguments.

Definition 1. An argumentation framework is a pair $AF = (\mathcal{A}, att)$ in which \mathcal{A} is a finite¹ set of arguments and $att \subseteq \mathcal{A} \times \mathcal{A}$. Given a set $Args \subseteq \mathcal{A}$, the restriction of AF to Args, denoted as $AF\downarrow_{Args}$, is the argumentation framework (Args, $att \cap (Args \times Args)$). The (infinite) set of all possible argumentation frameworks is denoted as SAF.

We will also need two relations and two operators between argumentation frameworks.

Definition 2. Given two argumentation frameworks $AF_1 = (\mathscr{A}_1, att_1)$ and $AF_2 = (\mathscr{A}_2, att_2)$:

- $AF_1 \subseteq AF_2$ iff $\mathscr{A}_1 \subseteq \mathscr{A}_2$ and $att_1 \subseteq att_2$
- $AF_1 \sqsubseteq AF_2$ iff $\mathscr{A}_1 \subseteq \mathscr{A}_2$ and $AF_2 \downarrow_{\mathscr{A}_1} = AF_1$
- $AF_1 \ominus AF_2 \triangleq \mathscr{A}_1 \smallsetminus \mathscr{A}_2$
- $AF_1 \setminus AF_2 \triangleq AF_1 \downarrow_{AF_1 \ominus AF_2}$

The relation \subseteq extends set inclusion to argumentation frameworks, while $AF_1 \sqsubseteq AF_2$ holds if AF_1 is a subframework² of AF_2 . In this case, $AF_2 \ominus AF_1$ returns the set of arguments of AF_2 outside AF_1 , while $AF_2 \setminus AF_1$ returns the corresponding argumentation framework.

¹In the general definition, the set of arguments may be infinite.

²It is immediate to see that \sqsubseteq is stricter than \subseteq , i.e. $AF_1 \sqsubseteq AF_2$ entails $AF_1 \subseteq AF_2$.

In this paper we adopt the labelling-based approach to the definition of argumentation semantics. A labelling assigns to each argument of an argumentation framework a label belonging to the set {in, out, undec}, where the label in means that the argument is accepted, the label out means that the argument is rejected, and the label undec means that the status of the argument is undecided. For technical reasons, we define labellings both for argumentation frameworks and for arbitrary sets of arguments.

Definition 3. Given a set of arguments Args, a labelling of Args is a total function $Lab : Args \rightarrow \{in, out, undec\}$. The set of all labellings of Args is denoted as \mathfrak{A}_{Args} . Given an argumentation framework $AF = (\mathcal{A}, att)$, a labelling of AF is a labelling of \mathcal{A} . The set of all labellings of AF is denoted as $\mathfrak{A}(AF)$. For a labelling Lab of Args, the restriction of Lab to a set of arguments $Args' \subseteq Args$, denoted as $Lab\downarrow_{Args'}$, is defined as $Lab \cap (Args' \times \{in, out, undec\})$. We extend this notation to sets of labellings, i.e. given a set of a labellings $\mathfrak{A} \subseteq \mathfrak{A}_{Args}$, $\mathfrak{A}\downarrow_{Args'} \triangleq \{Lab\downarrow_{Args'} \mid Lab \in \mathfrak{A}\}$. Moreover, if $Lab \in \mathfrak{A}(AF)$ and $AF' \subseteq AF$, where $AF' = (\mathcal{A}', att')$, $Lab\downarrow_{AF'}$ will denote $Lab\downarrow_{\mathcal{A}'}$.

A labelling-based semantics prescribes a set of labellings for each argumentation framework.

Definition 4. Given an argumentation framework $AF = (\mathcal{A}, att)$, a labelling-based semantics **S** associates with AF a subset of $\mathfrak{L}(AF)$, denoted as $L_{\mathbf{S}}(AF)$.

Many semantics exist, but since we are not concerned with specific definitions we refer the reader to [1, 17].

3. A General Model for Studying Decomposability

The model proposed in [16] for the analysis of decomposability of argumentation semantics can be articulated in two layers. The first layer deals with the modelling of the information locally used for the computation of labellings in subframeworks, the second layer represents this computation through the notion of the local function.

3.1. Local Information Function and Argumentation Framework with Input

Let us consider an argumentation framework AF^* and a subframework $AF : AF \sqsubseteq AF^*$. The information needed for the local computation of the labellings in AF should include the topology of the subframework itself, but also some knowledge of the topology of the neighboring part of the graph, as well as the labelling assigned to this part by the local computations on external subframeworks. The notion of *local information function* is able to model different kinds of available topological information.

Definition 5. A local information function is a function $LI : \{(AF^*, AF) \mid AF^*, AF \in SAF \land AF \sqsubseteq AF^*\} \rightarrow SAF$ such that $\forall AF^*, AF \in SAF : AF \sqsubseteq AF^*$

- $AF \sqsubseteq LI(AF^*, AF)$ and $LI(AF^*, AF) \subseteq AF^*$
- $if AF^* \subseteq AF^{**}$ then either $LI(AF^{**}, AF) = LI(AF^*, AF)$ or it is not the case that $LI(AF^{**}, AF) \subseteq AF^*$

For ease of notation, in the following $LI(AF^*, AF)$ will be denoted as $LI_{AF^*}(AF)$.

In the first item of the above definition, $AF \sqsubseteq LI(AF^*, AF)$ signifies that the local subframework must be known, while $LI(AF^*, AF) \subseteq AF^*$ expresses that the neighboring part of AFreturned by the function is taken from AF^* . For instance, only external attackers with the relevant attacks might be available (while the reverse attacks might be unknown), or we can have also information about the attacked external arguments and the relevant bidirectional attacks. The second item is meant to avoid implicit information hidden in the way the output of the function is selected depending on AF^* . To avoid this possibility, the constraint requires that if AF^* is enlarged, then either the output of LI does not change, or the additional elements of the enlarged global framework play an explicit role, i.e. some appear in the novel output of the local information function. In [16] it is shown that Definition 5 is able to model many different kinds of local information available (see also Example 1 below).

The information available for a specific subframework of a given framework is represented by an *argumentation framework with input*.

Definition 6. An argumentation framework with input is a tuple (AF, AF', Lab) where AF, AF' \in SAF such that $AF \sqsubseteq AF'$, and $Lab \in \mathfrak{L}_{AF' \ominus AF'}$.

Intuitively, *AF* represents a subframework, *AF*' represents the portion of the global argumentation framework which is taken into account, including *AF* itself, while *Lab* is the labelling externally assigned to arguments in $AF' \ominus AF$, i.e. belonging to the neighboring part of the subframework.

An argumentation framework with input can be derived by applying a local information function LI to a subframework AF of a global argumentation framework AF^* . If there is an argumentation framework AF^* where this is possible, the argumentation framework with input is said to be derived from LI.

Definition 7. An argumentation framework with input (AF, AF', Lab) is derived from a local information function LI in AF^{*}, written (AF, AF', Lab) $\in AF_{LI,AF^*}^{inp}$, if $AF' = LI_{AF^*}(AF)$. (AF, AF', Lab) is derived from LI, written (AF, AF', Lab) $\in AF_{LI}^{inp}$, if $\exists AF^*$ such that (AF, AF', Lab) $\in AF_{LI,AF^*}^{inp}$.

While in Definition 7 the labelling component of argumentation frameworks with input is not constrained, the notion of realizability introduced in the following definition requires the labelling component to be enforced by a labelling prescribed by the semantics.

Definition 8. An argumentation framework with input (AF, AF', Lab) is realized from a local information function LI in an argumentation framework AF^* under a semantics **S**, written (AF, AF', Lab) $\in RAF_{LI,AF^*,S}^{inp}$, if (AF, AF', Lab) $\in AF_{LI,AF^*}^{inp}$ and $\exists Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*)$ such that $Lab^* \downarrow_{AF' \ominus AF} = Lab$. (AF, AF', Lab) is realized from a local information function LI under a semantics **S**, written (AF, AF', Lab) $\in RAF_{LI,S}^{inp}$, if $\exists AF^*$ such that (AF, AF', Lab) $\in RAF_{LI,AF^*,S}^{inp}$.

Example 1. Suppose that the available external information for any subframework AF includes the outside attackers and the unidirectional attacks from them to AF. The relevant local information function inpLI can be defined as follows. First, given $AF^* = (\mathcal{A}^*, att^*) \in SAF$ and $Args \subseteq \mathcal{A}^*$, $Args_{AF^*}^{inp} \equiv \{\alpha \in \mathcal{A}^* \setminus Args \mid \exists \beta \in Args, (\alpha, \beta) \in att^*\}$ and $Args_{AF^*}^{att-inp} = att^* \cap (Args_{AF^*}^{inp} \times Args_{AF^*}^{inp})$

Args). Then, for any (AF^*, AF) such that $AF^*, AF \in SAF \land AF \sqsubseteq AF^*$, with $AF = (\mathscr{A}, att)$, $inpLI_{AF^*}(AF) \equiv (\mathscr{A} \cup \mathscr{A}_{AF^*}^{inp}, att \cup \mathscr{A}_{AF^*}^{att-inp})$. Now, consider $AF^* = (\{\alpha, \beta, \gamma, \delta_1, \delta_2\}, \{(\alpha, \beta), (\beta, \gamma), (\beta, \delta_1), (\gamma, \delta_1), (\delta_1, \delta_2), (\delta_2, \delta_1)\}$ and $AF = (\{\alpha, \beta, \gamma, \delta_1, \delta_2\}, \{(\alpha, \beta), (\beta, \gamma), (\beta, \delta_1), (\gamma, \delta_1), (\delta_1, \delta_2), (\delta_2, \delta_1)\}$

Now, consider $AF^* = (\{\alpha, \beta, \gamma, \delta_1, \delta_2\}, \{(\alpha, \beta), (\beta, \gamma), (\beta, \delta_1), (\gamma, \delta_1), (\delta_1, \delta_2), (\delta_2, \delta_1)\})$ and $AF = AF^* \downarrow_{\{\delta_1, \delta_2\}}$. We have that $inpLI_{AF^*}(AF) = AF' = (\{\beta, \gamma, \delta_1, \delta_2\}, \{(\beta, \delta_1), (\gamma, \delta_1), (\delta_1, \delta_2), (\delta_2, \delta_1)\})$. The example also shows that e.g. the argumentation framework with input $(AF, AF', \{(\beta, in), (\gamma, in)\})$ is derived from inpLI. Under most semantics **S** (e.g. the grounded or preferred semantics [1]) it also holds $(AF, AF', \{(\beta, in), (\gamma, in)\}) \in RAF_{inpLI,S}^{inp}$. However, if we change the definition of inpLI so as to include also attack between external attackers, this would not hold since most semantics prohibit conflicting arguments $(\beta \text{ and } \gamma \text{ in this case})$ from being all labelled in.

3.2. Local Function and Decomposability

A local function represents a local counterpart of the notion of semantics. It takes as input an argumentation framework with input (rather than a standard argumentation framework) and produces as output a set of labellings for the inner local argumentation framework.

Definition 9. A local function F for a local information function LI assigns to any $(AF, AF', Lab) \in AF_{LI}^{inp}$ a (possibly empty) set of labellings of AF, i.e. $F(AF, AF', Lab) \in 2^{\mathfrak{L}(AF)}$.

A semantics **S** is *decomposable* (also called fully decomposable) if the labellings prescribed on an argumentation framework AF correspond to the possible combinations of *compatible* labellings obtained by applying a local function F in the subframeworks that partition the global framework.

Definition 10. A local function F for a local information function LI enforces decomposability of a semantics S under LI iff for every argumentation framework $AF = (\mathscr{A}, att)$ and for every partition $\mathscr{P} = \{P_1, ..., P_n\}$ of \mathscr{A} , the following condition holds: $L_S(AF) = \{L_{P_1} \cup ... \cup L_{P_n} \mid L_{P_i} \in F(AF \downarrow_{P_i}, LI_{AF}(AF \downarrow_{P_i}), (\bigcup_{j=1...n, j\neq i} L_{P_j}) \downarrow_{LI_{AF}}(AF \downarrow_{P_i}) \in A$ semantics S is decomposable (or equivalently fully decomposable) under LI iff there is a local function F which enforces decomposability of S under LI.

In Definition 10, $(AF \downarrow_{P_i}, LI_{AF}(AF \downarrow_{P_i}), (\bigcup_{j=1...n, j\neq i} L_{P_j}) \downarrow_{LI_{AF}(AF \downarrow_{P_i}) \ominus AF \downarrow_{P_i}})$ is an argumentation framework with input representing the subframework of AF on the partition element P_i (first component) enriched with the available external information. In particular, the second component is the available topological information on the neighboring part. The third component is the labelling assigned to the locally known arguments outside the subframework $AF \downarrow_{P_i}$, i.e. those included in the set $LI_{AF}(AF \downarrow_{P_i}) \ominus AF \downarrow_{P_i}$. Compatibility refers to the fact that any labelling of a subframework is used by F to compute other labellings in other subframeworks. More specifically, each local labelling L_{P_i} depends on the other ones since the labelling component taken as input by F is obtained from the labellings L_{P_j} (with $j \neq i$) computed in external subframeworks.

Decomposability can be split into two partial decomposability properties.

Definition 11. A local function F for a local information function LI enforces top-down decomposability of a semantics **S** under LI iff for every argumentation framework $AF = (\mathcal{A}, att)$ and for every partition $\mathcal{P} = \{P_1, ..., P_n\}$, it holds that $\mathbf{L}_{\mathbf{S}}(AF) \subseteq \{L_{P_1} \cup ... \cup L_{P_n} \mid L_{P_i} \in \mathcal{A}\}$

$$\begin{split} F(AF\downarrow_{P_i}, LI_{AF}(AF\downarrow_{P_i}), (\bigcup_{j=1...n, j\neq i} L_{P_j})\downarrow_{LI_{AF}(AF\downarrow_{P_i})\ominus AF\downarrow_{P_i}})\}. \ A \ local \ function \ F \ for \ a \ local \ information \ function \ LI \ enforces \ bottom-up \ decomposability \ of \ a \ semantics \ S \ under \ LI \ iff \ for \ every \ argumentation \ framework \ AF = (\mathscr{A}, att) \ and \ for \ every \ partition \ \mathscr{P} = \{P_1, \ldots, P_n\}, \ it \ holds \ that \ \mathbf{L}_{\mathbf{S}}(AF) \supseteq \{L_{P_1} \cup \ldots \cup L_{P_n} \mid L_{P_i} \in F(AF\downarrow_{P_i}, LI_{AF}(AF\downarrow_{P_i}), (\bigcup_{j=1...n, j\neq i} L_{P_j})\downarrow_{LI_{AF}(AF\downarrow_{P_i})\ominus AF\downarrow_{P_i}})\}. \end{split}$$

In words, F enforces top-down decomposability if the procedure to compute labellings by means of F is complete, i.e. all labellings prescribed by **S** for AF are obtained by applying F to the subframeworks corresponding to the partition and combining the relevant labellings. On the other hand, F enforces bottom-up decomposability if the procedure is sound, i.e. all combinations of local labellings obtained by F give rise to global labellings that are valid according to **S**. It is easy to see that a semantics is decomposable under LI iff there is a local function F for LI which enforces bott top-down and bottom-up decomposability of **S** under LI.

4. A Constructive Procedure for Local Functions

Once the general model has been designed, the next issue is to identify a local function for any argumentation semantics \mathbf{S} and local information function *LI*.

Given a specific argumentation semantics S, one may rely on the relevant definition to investigate its decomposability properties under LI. Then, if the semantics turns out to be decomposable, one may determine a local function which enforces full decomposability under LI, while in the other case one may identify a local function satisfying some desired properties, e.g. achieving full decomposability under LI w.r.t. specific kinds of partitions. In order to provide a sort of guidance to this activity which is valid independently of the specific semantics definitions, we aim at identifying an expression of the local function which is parametric w.r.t. the semantics, and thus does not rely on the properties of a specific semantics.

The expression of the local function is based on the following considerations. First, given an argumentation framework with input $(AF, AF', Lab) \in AF_{LI}^{inp}$, the only way to determine the set of labellings returned as output by the local function on the basis of the semantics **S** (given as a parameter) is to apply **S** to a set of argumentation frameworks. Since the set of labellings returned by the local function is contained in $\mathfrak{Q}(AF)$, each of these argumentation frameworks AF^* must have AF as a subframework, i.e. $AF \subseteq AF^*$, and the returned labellings are obtained by restricting (some of) the labellings in $\mathbf{L}_{\mathbf{S}}(AF^*)$ to AF. Moreover, taking into account the role of AF' and Lab, the argumentation with input (AF, AF', Lab) has to be realized in AF^* from LI, and only the labellings $Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*)$ compatible with Lab (i.e. such that $Lab^* \downarrow_{AF' \ominus AF} = Lab$) should be taken into account.

In order to model all possible selections of argumentation frameworks for any $(AF, AF', Lab) \in AF_{LI}^{inp}$, we introduce the notion of *standard argumentation framework function*, which associates to any argumentation framework with input derived from LI a (possibly empty) set of argumentation frameworks in which this argumentation framework with input is realized.

Definition 12. Given a local information function LI, a standard argumentation framework function f_{ST} for LI is a (possibly partial) function which associates to any pair including a semantics **S** and an argumentation framework with input (AF, AF', Lab) $\in AF_{LI}^{inp}$, a set of argumentation frameworks, denoted as $f_{ST}^{S,LI}(AF, AF', Lab)$, such that $f_{ST}^{S,LI}(AF, AF', Lab) \subseteq \{AF^* \mid AF^*\}$

 $(AF, AF', Lab) \in RAF_{LI,AF^*,S}^{inp}$. A standard argumentation framework function for LI is finite if, $\forall (AF, AF', Lab) \in AF_{LI}^{inp}, f_{ST}^{S,LI}(AF, AF', Lab)$ is finite. It is unitary if, $\forall (AF, AF', Lab) \in AF_{LI}^{inp}$, either $f_{ST}^{S,LI}(AF, AF', Lab)$ includes a single framework or it is empty.

Note that if $(AF, AF', Lab) \notin RAF_{LI,S}^{inp}$ then $f_{ST}^{S,LI}(AF, AF', Lab)$ is not defined, i.e. returns the empty set.

Intuitively, the aim of $f_{ST}^{S,LI}(AF, AF', Lab)$ is to provide a set of argumentation frameworks 'representing' all argumentation frameworks where the argumentation framework with input (AF, AF', Lab) can be realized, meaning that such a set is sufficient to construct the output of a local function *F*. In particular, given a standard argumentation framework function f_{ST} for *LI*, for any semantics **S** a corresponding local function for *LI* can be generated as in the following definition.

Definition 13. Given a standard argumentation framework function f_{ST} for a local information function LI and a semantics **S**, the local function generated by f_{ST} for **S** and LI, denoted as $F_{f_{ST},\mathbf{S},LI}$, is the local function for LI such that for any $(AF, AF', Lab) \in AF_{LI}^{inp}$

$$F_{f_{ST},\mathbf{S},LI}(AF, AF', Lab) = \bigcup_{AF^* \in f_{ST}} \{Lab^* \downarrow_{AF} \mid Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*), Lab^* \downarrow_{AF' \ominus AF} = Lab\}$$

It is easy to see that a monotonic relation between standard argumentation framework functions and generated local functions holds.

Proposition 1. Given two standard argumentation framework functions f_{ST}^1 and f_{ST}^2 for LI and given a semantics **S**, if $f_{ST}^1^{\mathbf{S},LI}(AF, AF', Lab) \subseteq f_{ST}^2^{\mathbf{S},LI}(AF, AF', Lab)$ then $F_{f_{ST}^1, \mathbf{S},LI}(AF, AF', Lab) \subseteq F_{f_{ST}^2, \mathbf{S},LI}(AF, AF', Lab)$.

Proof. The result easily follows from Definitions 12 and 13.

Let us now turn on two possible requirements for a standard argumentation framework function.

First, constructing a local function on the basis of a standard argumentation framework function is easier if the latter is finite. Luckily, since we deal with finite argumentation frameworks, for any generated local function there is always a finite standard argumentation framework function which generates it.

Proposition 2. Given a standard argumentation framework function f_{ST}^1 for a local information function LI and a semantics **S**, there exists a finite standard argumentation framework function f_{ST}^2 for LI which generates $F_{f_{ST}^1, \mathbf{S}, LI}$.

Proof. We construct f_{ST}^2 as follows. According to Definition 13, for any $(AF, AF', Lab) \in AF_{LI}^{inp}$ the output of $F_{f_{T}^1 \mathbf{S}, LI}(AF, AF', Lab)$ can be expressed as

$$\bigcup_{AF^* \in f_{ST}^{1}^{\mathbf{S},LI}(AF,AF',Lab)} \{Lab^* \downarrow_{AF} \mid Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*) \land Lab^* \downarrow_{AF' \ominus AF} = Lab\}$$

Since the number of possible labellings of *AF*, i.e. the cardinality of $\mathfrak{Q}(AF)$, is 3^n where *n* is the number of arguments in *AF*, obviously the number of distinct labellings $Lab^* \downarrow_{AF}$ in the set above is finite as well. Thus there is a finite set of argumentation frameworks, that we let as $f_{ST}^{2,S,LI}(AF, AF', Lab)$, such that

$$F_{f_{ST}^{1},\mathbf{S},LI}(AF, AF', Lab) = \bigcup_{AF^{*} \in f_{ST}^{2}} \bigcup_{(AF, AF', Lab)} \{Lab^{*} \downarrow_{AF} \mid Lab^{*} \in \mathbf{L}_{\mathbf{S}}(AF^{*}) \land Lab^{*} \downarrow_{AF' \ominus AF} = Lab\}$$

This corresponds to our desired f_{ST}^2 (see Definition 13).

Let us now turn to the second requirement. Since by definition a decomposable semantics \mathbf{S} under a local information function LI admits a (possibly singleton) set of local functions that enforce decomposability of \mathbf{S} under LI, failing to capture all of them would not be acceptable for the above construction mechanism. This is expressed by the following definition.

Definition 14. A standard argumentation framework function f_{ST} for LI is adequate if, for every decomposable semantics **S** under LI, $F_{f_{ST},S,LI}$ enforces decomposability of **S** under LI.

An adequate standard argumentation framework function f_{ST} is pivotal for investigating the decomposability property of a semantics **S**, since it allows one to select without loss of generality the local function in the condition of Definition 10. In particular, since by Definition 14 $F_{f_{ST},\mathbf{S},LI}$ enforces decomposability of **S** if the latter is fully decomposable under a local information function *LI*, the proof that **S** is fully decomposable under *LI* can focus on this condition with $F = F_{f_{ST},\mathbf{S},LI}$. Conversely, in order to show that a semantics is not decomposable it is sufficient to identify an argumentation framework and a partition where the same condition is not satisfied by $F_{f_{ST},\mathbf{S},LI}$.

A significant question is then whether Definitions 12 and 13 or, more generally, the underlying assumptions introduced above, are general enough to capture useful local functions, i.e. whether there is (at least) one adequate standard argumentation framework function. In the next sections we provide a positive answer to this question.

5. The Canonical Local Function

In this section we consider a particular choice of a standard argumentation framework function, motivated by the fact that any local function enforcing decomposability must include as output, for any AF^* such that $(AF, AF', Lab) \in RAF_{LI,AF^*,S}^{inp}$, the restriction of the labellings of AF^* to the subframework AF. This is shown in the following proposition.

Proposition 3. Let **S** be a fully decomposable semantics under LI, and let $(AF, AF', Lab) \in AF_{LI}^{inp}$ be an argumentation framework with input derived from LI. Let AF^* be an argumentation framework such that $AF' = LI_{AF^*}(AF)$, and $Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*)$ be a labelling of AF^* such that $Lab^* \downarrow_{AF' \ominus AF} = Lab$. Then, for any local function F which enforces decomposability of **S** under LI, $Lab^* \downarrow_{AF} \in F(AF, AF', Lab)$.

We should note that the reverse of the above proposition does not hold, i.e. F may require additional labellings w.r.t. those mentioned in the proposition. A labelling included in F(AF, AF', Lab) may not play a role in forming the labellings of AF^* due to the compatibility conditions, but it may be required in a different argumentation framework. This suggests adopting the following definition of the *canonical local function*, which includes all possible labellings that play a role in some argumentation framework.

Definition 15. Given a semantics **S** and a local information function LI, the canonical local function $F_{\mathbf{S}}^{LI}$ of **S** associated to LI is defined as follows. For any $(AF, AF', Lab) \in AF_{LI}^{inp}$,

$$F_{\mathbf{S}}^{LI}(AF, AF', Lab) = \bigcup_{AF^* | (AF, AF', Lab) \in RAF_{IJAF^* \mathbf{S}}^{inp}} \{Lab^* \downarrow_{AF} | Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*) \land Lab^* \downarrow_{AF' \ominus AF} = Lab\}$$

It is easy to see that the canonical local function of a semantics **S** associated to *LI* is the local function generated by the maximal standard argumentation framework function for **S** and *LI*, i.e. returning as output *all* of the argumentation frameworks AF^* such that $(AF, AF', Lab) \in RAF_{LI,AF^*,S}^{inp}$ (see Definition 12 and Definition 13).

Due to the choice of considering all possible labellings compliant with Definitions 12 and 13, the canonical local function of any semantics **S** associated to a local information function LI enforces top-down decomposability of **S** under LI, as shown in the following proposition.

Proposition 4. For any semantics **S** and local information function LI, the canonical local function $F_{\mathbf{S}}^{LI}$ enforces top-down decomposability of **S** under LI.

Proof. According to Definition 11, we have to prove that for every $AF = (\mathscr{A}, att)$, for every partition $\mathscr{P} = \{P_1, ..., P_n\}$ and for any labelling $Lab \in \mathbf{L}_{\mathbf{S}}(AF)$, it holds that $Lab = L_{P_1} \cup ... \cup L_{P_n} \mid L_{P_i} \in F_{\mathbf{S}}^{L1}(AF\downarrow_{P_i}, LI_{AF}(AF\downarrow_{P_i}), (\bigcup_{j=1...n, j\neq i} L_{P_j})\downarrow_{LI_{AF}}(AF\downarrow_{P_i}) \oplus AF\downarrow_{P_i}$. For any $i \in \{1, ..., n\}$, let $L_{P_i} = Lab\downarrow_{P_i}$. It holds that $Lab = L_{P_1} \cup ... \cup L_{P_n}$, thus, for any $i, (\bigcup_{j=1...,n, j\neq i} L_{P_j})\downarrow_{LI_{AF}}(AF\downarrow_{P_i}) \oplus AF\downarrow_{P_i} = Lab\downarrow_{LI_{AF}}(AF\downarrow_{P_i}) \oplus AF\downarrow_{P_i}$. As a consequence, we have to prove that for any $i \in \{1, ..., n\}$ $Lab\downarrow_{P_i} \in F_{\mathbf{S}}^{L1}(AF\downarrow_{P_i}), Lab\downarrow_{LI_{AF}}(AF\downarrow_{P_i}) \oplus AF\downarrow_{P_i}$. According to the definition of canonical local function (see Definition 15) this amount to prove that there is an argumentation framework AF^* and a labelling $Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*)$ such that $LI_{AF^*}(AF\downarrow_{P_i}) = LI_{AF}(AF\downarrow_{P_i}), Lab^*\downarrow_{AF\downarrow_{P_i}} = Lab\downarrow_{P_i}$ and $Lab^*\downarrow_{LI_{AF}}(AF\downarrow_{P_i}) \oplus AF\downarrow_{P_i} = Lab\downarrow_{LI_{AF}}(AF\downarrow_{P_i}) \oplus AF\downarrow_{P_i}$. It is easy to see that all these conditions are satisfied by selecting $AF^* = AF$ and $Lab^* = Lab$. In particular, $Lab \in \mathbf{L}_{\mathbf{S}}(AF)$ holds by assumption, $LI_{AF}(AF\downarrow_{P_i}) = LI_{AF^*}(AF\downarrow_{P_i})$ is trivially satisfied, the third condition holds since $Lab\downarrow_{AF\downarrow_{P_i}} = Lab\downarrow_{P_i}$.

While top-down decomposability holds for all semantics, i.e. the output of the canonical local function is sufficient to cover all global labellings, the following proposition shows that the output of the canonical local function is necessary to enforce decomposability whenever this is possible, i.e. if the semantics is fully decomposable.

Proposition 5. Let **S** be a decomposable semantics under LI and let F be a local function which enforces decomposability of **S** under LI. Then, $\forall (AF, AF', Lab) \in AF_{LI}^{inp}, F_{\mathbf{S}}^{LI}(AF, AF', Lab) \subseteq F(AF, AF', Lab).$

The reverse of this proposition does not hold, since a local function enforcing decomposability can prescribe for a subframework spurious labellings that are not compatible with those of the other subframeworks, and thus do not alter the set of labellings obtained by joining the results of local computations.

The above results are sufficient to show that the canonical local function enforces decomposability of all decomposable semantics.

Proposition 6. If a semantics **S** is fully decomposable under a local information function LI, then $F_{\mathbf{S}}^{LI}$ enforces decomposability of **S** under LI.

According to Proposition 5, the canonical local function of a decomposable semantics **S** associated to *LI* is the minimal (w.r.t. \subseteq) local function enforcing decomposability.

6. Reduced Canonical Local Functions

As mentioned in the previous section, Proposition 3 identifies an argumentation framework AF^* and a relevant set of labellings that are necessary to enforce decomposability. On the other hand, in general a single argumentation framework is not sufficient, i.e. different argumentation frameworks may have to be identified in order to determine the whole set of labellings returned as output by the canonical local function for a given argumentation framework with input.

A single argumentation framework is sufficient, however, if some conditions are verified. These conditions, expressed in the following definition, depend both on the semantics and the local information function.

Definition 16. Let **S** be a semantics, LI a local information function and $(AF, AF', Lab) \in AF_{LI}^{inp}$ an argumentation framework with input derived from LI. An argumentation framework AF^* represents $(AF, AF', Lab) \in AF_{LI}^{inp}$ under **S** and LI, written $AF^* \in REP_{\mathbf{S}}^{LI}(AF, AF', Lab)$, if $AF' = LI_{AF^*}(AF)$, $\exists Lab'_1 \in \mathbf{L}_{\mathbf{S}}(AF^* \setminus AF)$ with $Lab'_1 \downarrow_{AF' \ominus AF} = Lab$, and $LI_{AF^*}(AF^* \setminus AF) = \emptyset$.

Some comments on the conditions of Definition 16 are in order. In particular, the key condition is $LI_{AF^*}(AF^* \setminus AF) = \emptyset$, which corresponds to a kind of unidirectional local information function, i.e. while AF is influenced by (part of) $AF^* \setminus AF$ and the relevant labelling, the reverse does not hold. Thus, the role of $AF^* \setminus AF$ is to enforce the labelling Lab in $AF' \setminus AF$ independently of AF. Enforcing such labelling is then possible if a labelling compatible with Lab is prescribed by the semantics, i.e. $AF' = LI_{AF^*}(AF)$ and $\exists Lab_1' \in \mathbf{L}_{\mathbf{S}}(AF^* \setminus AF)$ with $Lab_1' \downarrow_{AF' \ominus AF} = Lab$.

The next proposition shows that the reverse of Proposition 3 holds if the conditions of Definition 16 are satisfied, i.e. a single argumentation framework is sufficient if it represents the argumentation framework with input.

Proposition 7. Let **S** be a fully decomposable semantics under LI, and let $(AF, AF', Lab) \in AF_{LI}^{inp}$ be an argumentation framework with input derived from LI. Let AF^* be an argumentation framework such that $AF^* \in REP_{\mathbf{S}}^{LI}(AF, AF', Lab)$. Then, for any local function F which enforces decomposability of **S** under LI, $F(AF, AF', Lab) = \{Lab^* \downarrow_{AF} \mid Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*) \land Lab^* \downarrow_{AF' \ominus AF} = Lab\}$. *Proof.* Let us first consider a labelling $Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*)$ such that $Lab^* \downarrow_{AF \ominus AF} = Lab$. It is easy to see that all the hypotheses of Proposition 3 are satisfied, thus $Lab^* \downarrow_{AF} \in F(AF, AF', Lab)$.

As to the reverse direction of the proof, let us first consider the partition of AF^* identified by the subframeworks $AF^* \setminus AF$ and AF. Since by the hypothesis F enforces decomposability of **S** under *LI*, by Definition 10 and taking into account that $AF^* \downarrow_{AF^* \ominus AF} = AF^* \backslash AF$, we have that $\mathbf{L}_{\mathbf{S}}(AF^{*}) = \{Lab_{1} \cup Lab_{2} \mid Lab_{1} \in F(AF^{*} \setminus AF, LI_{AF^{*}}(AF^{*} \setminus AF), Lab_{2} \downarrow_{LI_{AF^{*}}(AF^{*} \setminus AF) \ominus (AF^{*} \setminus AF)}), Lab_{2} \in \mathcal{L}_{\mathbf{S}}(AF^{*} \setminus AF) \in \mathcal{L}_{\mathbf{S}}(AF^{*} \setminus AF)$ $F(AF, LI_{AF^*}(AF), Lab_1 \downarrow_{LI_{AF^*}(AF) \ominus AF})$. Since $LI_{AF^*}(AF^* \setminus AF) = \emptyset$ and $AF' = LI_{AF^*}(AF)$, it holds that

$$\mathbf{L}_{\mathbf{S}}(AF^{*}) = \{Lab_{1} \cup Lab_{2} \mid Lab_{1} \in F(AF^{*} \setminus AF, \emptyset, \emptyset), Lab_{2} \in F(AF, AF', Lab_{1} \downarrow_{AF' \ominus AF})\}$$
(1)

Let us then consider a labelling $Lab'_2 \in F(AF, AF', Lab)$. We have to prove that $\exists Lab^{*'} \in$

 $\mathbf{L}_{\mathbf{S}}(AF^*)$ such that $Lab^{*'}\downarrow_{AF'\ominus AF} = Lab$ and $Lab'_2 = Lab^{*'}\downarrow_{AF}$. By the hypothesis that $AF^* \in REP_{\mathbf{S}}^{LI}(AF, AF', Lab), \exists Lab'_1 \in \mathbf{L}_{\mathbf{S}}(AF^* \setminus AF)$ with $Lab'_1\downarrow_{AF'\ominus AF} = Lab$. Let us now identify $Lab^{*'}$ as $Lab'_1 \cup Lab'_2$. It obviously holds that $Lab'_2 = Lab^{*'}\downarrow_{AF}$ and $Lab^{*'}\downarrow_{AF \ominus AF} = Lab$. Moreover, since F enforces decomposability of **S** under LI, considering the argumentation framework $AF^* \setminus AF$ and the partition including a single set (which includes in turn all the relevant arguments), from $Lab'_1 \in \mathbf{L}_{\mathbf{S}}(AF^* \setminus AF)$ we get $Lab'_1 \in F(AF^* \setminus AF, \emptyset, \emptyset)$. Since $Lab'_2 \in F(AF, AF', Lab) = F(AF, AF', Lab'_1 \downarrow_{AF' \ominus AF})$, by (1) we have $Lab^{*'} \in \mathbf{L}_{\mathbf{S}}(AF^*)$. \Box

In order to exploit Proposition 7 to identify a local function generated by a unitary standard argumentation framework function, we need a number of preliminary definitions.

First, for a given argumentation framework with input (AF, AF', Lab) we need to focus on the pair $(AF' \setminus AF, Lab)$, playing for AF the role of the 'input pair' affecting the local computation of labellings in AF. Accordingly, we introduce the following definition of a pair derived from a local information function LI.

Definition 17. Given a local information function LI, a pair (AF^{in}, Lab) (where $AF^{in} \in SAF$ and $Lab \in \mathfrak{L}(AF^{in})$ is derived from LI, written $(AF^{in}, Lab) \in P_{LI}$, if $\exists (AF, AF', Lab) \in AF_{LI}^{inp}$ such that $AF' \setminus AF = AF^{in}$.

A pair is *representable* if every relevant argumentation framework with input can be represented by an argumentation framework.

Definition 18. Given a semantics **S** and a local information function LI, a pair $(AF^{in}, Lab) \in P_{LI}$ is representable under **S** and LI, written $(AF^{in}, Lab) \in P^{rep}_{\mathbf{S},LI}$, if for every $(AF, AF', Lab) \in AF^{inp}_{LI}$ such that $AF' \setminus AF = AF^{in}, \exists AF^* \in REP_{\mathbf{S}}^{LI}(AF, AF', Lab).$

Similarly to the case of a realized argumentation framework with input (see Definition 8), we introduce the notion of realizability of a pair under a semantics.

Definition 19. Given a semantics **S**, a pair (AF^{in}, Lab) is realized under **S**, written $(AF^{in}, Lab) \in P_{\mathbf{S}}^{real}$, if $\exists AF^* \in SAF$ such that $AF^{in} \subseteq AF^*$ and $\exists Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*)$ such that $Lab^* \downarrow_{AF^{in}} = Lab$.

In words, there must be an argumentation framework where AF^{in} appears as a potential external information for a subframework, and the semantics enforces the labelling Lab in AF^{n} . As shown below, if a pair is representable under **S** and *LI* then it is also realized under **S**.

Proposition 8. Given a semantics **S**, a local information function LI and a pair $(AF^{in}, Lab) \in P_{LI}$, $if(AF^{in}, Lab) \in P_{\mathbf{S},LI}^{rep}$, then $(AF^{in}, Lab) \in P_{\mathbf{S}}^{real}$.

Proof. By Definition 17, $\exists (AF, AF', Lab) \in AF_{LI}^{inp}$ such that $AF' \setminus AF = AF^{in}$. Since $(AF^{in}, Lab) \in P_{\mathbf{S},LI}^{rep}$, by Definition 18 $\exists AF^{**} \in REP_{\mathbf{S}}^{LI}(AF, AF', Lab)$. Taking into account Definitions 16 and 5, we have in particular $AF \sqsubseteq AF^{**}, AF' \subseteq AF^{**}$, and $\exists Lab'_1 \in \mathbf{L}_{\mathbf{S}}(AF^{**} \setminus AF)$ with $Lab'_1 \downarrow_{AF' \ominus AF} = Lab$. Letting $AF^* = AF^{**} \setminus AF$ and $Lab^* = Lab'_1$, it must be the case that $AF' \setminus AF \subseteq AF^*$, i.e. $AF^{in} \subseteq AF^*$, and $\exists Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*)$ with $Lab^* \downarrow_{AF' \ominus AF} = Lab$. According to Definition 19, $(AF^{in}, Lab) \in P_{\mathbf{S}}^{real}$.

On the basis of Proposition 7, if all pairs are representable (and thus realized) then it is possible to construct a local function by means of a unitary standard argumentation framework function. However, this requirement may be impossible to achieve just because of pairs that are not realized under the semantics (and thus cannot be representable). Then, a weaker requirement is that realized pairs are representable. We introduce accordingly the following definition.

Definition 20. A semantics **S** is representable w.r.t. a local information function LI if for every $(AF^{in}, Lab) \in P_{LI}$, it holds that $(AF^{in}, Lab) \in P_{\mathbf{S},LI}^{rep}$, i.e. every pair is representable. A semantics **S** is weakly representable w.r.t. LI if for every $(AF^{in}, Lab) \in P_{LI}$ such that $(AF^{in}, Lab) \in P_{\mathbf{S}}^{real}$, it holds that $(AF^{in}, Lab) \in P_{\mathbf{S},LI}$, i.e. every realized pair is representable.

It is immediate to see that a representable semantics is also weakly representable.

Example 2. Consider the local information function inpLI as defined in Example 1. By definition of inpLI, any pair $(AF^{in}, Lab) \in P_{inpLI}$ involves only initial nodes, i.e. not receiving attacks. Under most semantics **S** (e.g. admissible, complete, grounded and preferred semantics [1]) $(AF^{in}, Lab) \in P_{\mathbf{S},LI}^{rep}$, thus **S** is representable. In particular, given $(AF, AF', Lab) \in AF_{LI}^{inp}$ such that $AF' \setminus AF = AF^{in}$, we can construct AF^* as required by Definition 18 by modifying AF' as follows. For any α labelled out by Lab, we add an unattacked argument α' attacking α , and for any α labelled undec we add an argument α' which attacks itself and α (and α' is not attacked by other arguments). As required by Definition 16, most semantics return a labelling of $AF^* \setminus AF$ coinciding with Lab for the relevant arguments.

For the stable semantics [1] the above construction is possible for all pairs where the labelling does not assign under to any argument, i.e. for all realizable pairs. Thus, the stable semantics is weakly representable.

We are now in a position to introduce the notion of reduced canonical local function. Basically, for any argumentation framework with input (AF, AF', Lab) with a corresponding pair ($AF' \\ AF$, Lab) which is realizable, an argumentation framework AF^* is selected that represents (AF, AF', Lab), and the output labellings are identified as in Proposition 7. If instead the pair is not realizable, the function returns an empty set of labellings.

Definition 21. Given a local information function LI and a weakly representable semantics **S** w.r.t. LI, a reduced canonical local function of **S** w.r.t. LI is a local function RF_{S}^{LI} such that for any

 $(AF, AF', Lab) \in AF_{LI}^{inp}$

$$RF_{\mathbf{S}}^{LI}(AF, AF', Lab) = \begin{cases} \{Lab^* \downarrow_{AF} \mid Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*) \land Lab^* \downarrow_{AF' \ominus AF} = Lab \} \\ if(AF' \smallsetminus AF, Lab) \in P_{\mathbf{S}}^{real} \\ \emptyset \text{ otherwise} \end{cases}$$

where AF^* is an argumentation framework such that $AF^* \in REP_{\mathbf{S}}^{LI}(AF, AF', Lab)$ selected to represent $(AF, AF', Lab) \in AF_{LI}^{inp}$.

Definition 21 is well defined, as shown in the following proposition.

Proposition 9. Let LI be a local information function and **S** a weakly representable semantics w.r.t. LI. For any $(AF, AF', Lab) \in AF_{LI}^{inp}$, if $(AF' \setminus AF, Lab) \in P_{\mathbf{S}}^{real}$ then $\exists AF^* \in REP_{\mathbf{S}}^{LI}(AF, AF', Lab)$, *i.e. the selection of an argumentation framework* AF^* *is possible.*

Proof. Since **S** is weakly representable, if $(AF' \setminus AF, Lab) \in P_{\mathbf{S}}^{real}$ then $(AF' \setminus AF, Lab) \in P_{\mathbf{S},Ll}^{rep}$, and the conclusion follows from Definition 18.

The suitability of a reduced canonical local function is confirmed by the following propositions.

Proposition 10. Let LI be a local information function and **S** a weakly representable semantics w.r.t. LI. If **S** is fully decomposable under LI, a reduced canonical local function RF_{S}^{LI} of **S** w.r.t. LI enforces decomposability of **S** under LI.

Proof. Since **S** is fully decomposable under *LI*, there is a local function *F* for *LI* such that for every argumentation framework $AF = (\mathcal{A}, att)$ and for every partition $\mathcal{P} = \{P_1, \dots, P_n\}$

$$\mathbf{L}_{\mathbf{S}}(AF) = \{L_{P_1} \cup \dots \cup L_{P_n} \mid L_{P_i} \in F(AF \downarrow_{P_i}, LI_{AF}(AF \downarrow_{P_i}), (\bigcup_{j=1\dots,n, j\neq i} L_{P_j}) \downarrow_{LI_{AF}(AF \downarrow_{P_i}) \ominus AF \downarrow_{P_i}})\}$$
(2)

and we have to prove that for every $AF = (\mathcal{A}, att)$ and for every partition $\mathcal{P} = \{P_1, \dots, P_n\}$

$$\mathbf{L}_{\mathbf{S}}(AF) = \{L_{P_1} \cup \ldots \cup L_{P_n} \mid L_{P_i} \in RF_{\mathbf{S}}^{LI}(AF \downarrow_{P_i}, LI_{AF}(AF \downarrow_{P_i}), (\bigcup_{j=1...n, j \neq i} L_{P_j}) \downarrow_{LI_{AF}}(AF \downarrow_{P_i}) \ominus AF \downarrow_{P_i})\}$$

Let us first consider $Lab \in \mathbf{L}_{\mathbf{S}}(AF)$. By condition (2), we have that $Lab = L_{P1} \cup ... \cup L_{Pn}$ with $L_{Pi} \in F(AF \downarrow_{Pi}, LI_{AF}(AF \downarrow_{Pi}), (\bigcup_{j=1...n, j\neq i} L_{Pj}) \downarrow_{LI_{AF}}(AF \downarrow_{Pi}) \ominus AF \downarrow_{Pi}$. Taking into account that $Lab \in \mathbf{L}_{\mathbf{S}}(AF)$, obviously for any *i* the pair $(LI_{AF}(AF \downarrow_{Pi}) \setminus AF \downarrow_{Pi}, (\bigcup_{j=1...n, j\neq i} L_{Pj}) \downarrow_{LI_{AF}}(AF \downarrow_{Pi}) \ominus AF \downarrow_{Pi}$ is realized under \mathbf{S} , thus by Proposition 9 there is an argumentation framework AF^* selected for $RF_{\mathbf{S}}^{LI}$ to represent $(AF \downarrow_{Pi}, LI_{AF}(AF \downarrow_{Pi}), (\bigcup_{j=1...n, j\neq i} L_{Pj}) \downarrow_{LI_{AF}}(AF \downarrow_{Pi}) \ominus AF \downarrow_{Pi}$. We can then apply Proposition 7, obtaining $F(AF \downarrow_{Pi}, LI_{AF}(AF \downarrow_{Pi}), (\bigcup_{j=1...n, j\neq i} L_{Pj}) \downarrow_{LI_{AF}}(AF \downarrow_{Pi}) \ominus AF \downarrow_{Pi}$. According to Definition 21, this is equal to $RF_{\mathbf{S}}^{LI}(AF \downarrow_{Pi}, LI_{AF}(AF \downarrow_{Pi}), (\bigcup_{j=1...n, j\neq i} L_{Pj}) \downarrow_{LI_{AF}}(AF \downarrow_{Pi}) \ominus AF \downarrow_{Pi}$. Summing up, $Lab = L_{P1} \cup ... \cup L_{Pn}$ where $L_{Pi} \in RF_{\mathbf{S}}^{LI}(AF \downarrow_{Pi}, LI_{AF}(AF \downarrow_{Pi}), (\bigcup_{j=1...n, j\neq i} L_{Pj}) \downarrow_{LI_{AF}}(AF \downarrow_{Pi}) \ominus_{AF \downarrow_{Pi}}$. Summing up, $Lab = L_{P1} \cup ... \cup L_{Pn}$ where $L_{Pi} \in RF_{\mathbf{S}}^{LI}(AF \downarrow_{Pi}, LI_{AF}(AF \downarrow_{Pi}), (\bigcup_{j=1...n, j\neq i} L_{Pj}) \downarrow_{LI_{AF}}(AF \downarrow_{Pi}) \ominus_{AF \downarrow_{Pi}}$.

Turning to the reverse direction of the proof, consider a labellings $L_{P_1} \cup ... \cup L_{P_n}$ such that, for any $i, L_{P_i} \in RF_{\mathbf{S}}^{LI}(AF \downarrow_{P_i}, LI_{AF}(AF \downarrow_{P_i}), (\bigcup_{j=1...n, j\neq i} L_{P_j}) \downarrow_{LI_{AF}(AF \downarrow_{P_i}) \ominus AF \downarrow_{P_i}})$. According to Definition 21, AF^* is selected such that $AF^* \in REP_{\mathbf{S}}^{LI}(AF \downarrow_{P_i}, LI_{AF}(AF \downarrow_{P_i}), (\bigcup_{j=1...n, j\neq i} L_{P_j}) \downarrow_{LI_{AF}(AF \downarrow_{P_i}) \ominus AF \downarrow_{P_i}})$. Taking into account that F enforces decomposability of \mathbf{S} under LI, by Proposition 7 we have that $F(AF \downarrow_{P_i}, LI_{AF}(AF \downarrow_{P_i}), (\bigcup_{j=1...n, j\neq i} L_{P_j}) \downarrow_{LI_{AF}(AF \downarrow_{P_i}) \ominus AF \downarrow_{P_i}}) = \{Lab^* \downarrow_{AF \downarrow_{P_i}} \mid Lab^* \in \mathbf{L}_{\mathbf{S}}(AF^*) \land$ $Lab^* \downarrow_{LI_{AF}(AF \downarrow_{P_i}) \ominus AF \downarrow_{P_i}} = (\bigcup_{j=1...n, j\neq i} L_{P_j}) \downarrow_{LI_{AF}(AF \downarrow_{P_i}) \ominus AF \downarrow_{P_i}}\}$, which by Definition 21 is equal to $RF_{\mathbf{S}}^{LI}(AF \downarrow_{P_i}, LI_{AF}(AF \downarrow_{P_i}), (\bigcup_{j=1...n, j\neq i} L_{P_j}) \downarrow_{LI_{AF}(AF \downarrow_{P_i}) \ominus AF \downarrow_{P_i}})$. Thus, for every i it holds that $L_{P_i} \in$ $F(AF \downarrow_{P_i}, LI_{AF}(AF \downarrow_{P_i}), (\bigcup_{j=1...n, j\neq i} L_{P_j}) \downarrow_{LI_{AF}(AF \downarrow_{P_i}) \ominus AF \downarrow_{P_i}})$, and by (2) $L_{P_1} \cup ... \cup L_{P_n} \in \mathbf{L}_{\mathbf{S}}(AF)$. \Box

Proposition 11. Let LI be a local information function and **S** a representable semantics w.r.t. LI. If **S** is fully decomposable under LI, there is only a local function which enforces decomposability of **S** under LI, coinciding with any reduced canonical local function RF_{S}^{LI} of **S** w.r.t. LI.

Proof. Consider a reduced canonical local function $RF_{\mathbf{S}}^{LI}$ of **S** w.r.t. *LI*. If **S** is representable w.r.t. *LI*, for any $(AF, AF', Lab) \in AF_{LI}^{inp}$ it holds that $(AF' \setminus AF, Lab) \in P_{\mathbf{S},LI}^{rep}$, thus, by Proposition 8, $(AF' \setminus AF, Lab) \in P_{\mathbf{S}}^{real}$. As a consequence, $RF_{\mathbf{S}}^{LI}(AF, AF', Lab)$ is defined by the first item in Definition 21, and according to Proposition 7 its output is the same as that returned by any local function *F* which enforces decomposability of **S** under *LI*.

Example 3. According to the considerations in Example 2, Proposition 11 applies to most common semantics (including admissible, complete, grounded, and preferred), while Proposition 10 applies to stable semantics.

7. Discussion and Conclusion

In this paper, we have investigated the construction of local functions to locally compute labellings, adopting the general model introduced in [18] for studying the decomposability of argumentation semantics. Among the many future directions of this work, a first issue is to identify for the semantics available in the literature the canonical local function, or a reduced canonical local function, in an explicit form. This will be useful for studying decomposability under different local information functions and, possibly, determining the minimal local information sufficient to guarantee decomposability. This may in turn provide a solid basis for mixing different argumentation semantics adopted in different subframeworks. More specifically, decomposability may be a necessary requirement in the specific case where all semantics coincide. In this regard, using less information relaxes the tie between local computations and gives more flexibility in the mixing strategy.

References

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

- [2] P. Baroni, M. Giacomin, On principle-based evaluation of extension-based argumentation semantics, Artificial Intelligence 171 (2007) 675–700.
- [3] B. Liao, H. Huang, Partial semantics of argumentation: basic properties and empirical results, J. of Logic and Computation 23 (2013) 541–562.
- [4] R. Baumann, Splitting an argumentation framework, in: Proc. of LPNMR 2011 11th Int. Conf. on Logic Programming and Nonmonotonic Reasoning, 2011, pp. 40–53.
- [5] R. Baumann, G. Brewka, R. Wong, Splitting argumentation frameworks: An empirical evaluation, in: Theory and Applications of Formal Argumentation - First Int. Workshop (TAFA 2011). Revised Selected Papers, volume 7132 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 17–31.
- [6] F. Cerutti, M. Giacomin, M. Vallati, M. Zanella, A SCC recursive meta-algorithm for computing preferred labellings in abstract argumentation, in: Proc. of the 14th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2014), 2014.
- [7] F. Cerutti, I. Tachmazidis, M. Vallati, S. Batsakis, M. Giacomin, G. Antoniou, Exploiting parallelism for hard problems in abstract argumentation, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence, volume 29, 2015.
- [8] B. Liao, L. Jin, R. C. Koons, Dynamics of argumentation systems: A division-based method, Artificial Intelligence 175 (2011) 1790–1814.
- [9] R. Baumann, G. Brewka, Analyzing the equivalence zoo in abstract argumentation, in: Proc. of the 14th Int. Workshop on Computational Logic in Multi-Agent Systems (CLIMA XIV), 2013, pp. 18–33.
- [10] D. M. Gabbay, Fibring argumentation frames, Studia Logica 93 (2009) 231-295.
- [11] S. Villata, G. Boella, L. van Der Torre, Argumentation patterns, in: Proc. of ARGMAS 2011 8th Int. Workshop on Argumentation in Multi-Agent Systems, 2011, pp. 133–150.
- [12] P. Baroni, G. Boella, F. Cerutti, M. Giacomin, L. W. N. van der Torre, S. Villata, On the input/output behavior of argumentation frameworks, Artificial Intelligence 217 (2014) 144–197.
- [13] T. Rienstra, A. Perotti, S. Villata, D. Gabbay, L. van Der Torre, G. Boella, Multi-sorted argumentation frameworks, in: Theory and Applications of Formal Argumentation - First Int. Workshop (TAFA 2011). Revised Selected Papers, volume 7132 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 231–245.
- [14] M. Giacomin, Handling heterogeneous disagreements through abstract argumentation (extended abstract), in: Proc. of PRIMA 2017, 2017, pp. 3–11.
- [15] P. Baroni, M. Giacomin, Some considerations on epistemic and practical reasoning in abstract argumentation, in: Proc. of the 2nd Workshop on Advances In Argumentation In Artificial Intelligence, 2018, pp. 1–5.
- [16] M. Giacomin, P. Baroni, F. Cerutti, Towards a general theory of decomposability in abstract argumentation, in: Proc. of 4th International Conference on Logic and Argumentation (CLAR 2021), 2021, p. To appear.
- [17] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, The Knowledge Engineering Review 26:4 (2011) 365–410.
- [18] C. Cayrol, F. Dupin de Saint-Cyr, M. Lagasquie-Schiex, Change in abstract argumentation frameworks: adding an argument, Journal of Artificial Intelligence Research 38 (2010) 49–84.