A First Approach to Argumentation Label Functions

Marcos Cramer [a] and Jérémie Dauphin [b]
[a] *TU Dresden*
[b] *University of Luxembourg*

Abstract. An important approach to abstract argumentation is the labeling-based approach, in which one makes use of labelings that assign to each argument one of three labels: `in`, `out` or `und`. In this paper, we address the question, which of the twenty-seven functions from the set of labels to the set of labels can be represented by an argumentation framework. We prove that in preferred, complete and grounded semantics, eleven label functions can be represented in this way while sixteen label functions cannot be represented by any argumentation framework. We show how this analysis of label functions can be applied to prove an impossibility result: Argumentation frameworks extended with a certain kind of weak attack relation cannot be flattened to the standard Dung argumentation frameworks.

Keywords. knowledge representation, abstract argumentation, argumentation semantics, labelings, flattening

1. Introduction

Abstract argumentation frameworks (AFs) [12] are reasoning structures where one aims at extracting sets of jointly acceptable arguments. One of the central methods to do so is the labeling-based approach [2], in which one derives labelings which assign to each argument one of three labels: `in`, `out` or `und`. The arguments that are labeled `in` represent the arguments that are jointly acceptable, while the arguments that are `out` represent the ones that are defeated by those. The last label, `und` (*undecided*), represents the cases where one cannot, or decides with proper justification, not to assign either of these two labels. One advantage of the labeling approach is that to verify that an argument is correctly labeled, one only needs to check the labels of its direct ancestors. This allows for a more local evaluation, which is still equivalent to other global approaches such as the extension-based approach.

Many enrichments of abstract argumentation frameworks have been studied, e.g. with bipolar argumentation frameworks which add a second relation of support [9], or with argumentation frameworks with recursive attacks (AFRA) [3] in which attacks may also target other attacks. One methodology for evaluating such enriched frameworks while staying coherent with the basic framework is the flattening approach [6], where the enrichments added to the abstract argumentation frameworks are expressed in terms of extra arguments and attacks, allowing one to evaluate them as abstract argumentation frameworks. An essential concern in the flattening approach is whether the extra arguments and attacks produce the same behavior as the one intended by the enrichment they flatten. This raises a question: Which relations connecting two arguments can be expressed in terms of arguments and attacks alone?

In this paper we propose to address this research question by studying the representability of label functions, i.e. of functions which map each of the three labels to one of these labels. We prove that in preferred, complete and grounded semantics, eleven label functions can be represented by an AF while sixteen label functions cannot be represented by any AF. We show how this analysis of label functions can be applied to prove an impossibility result: Argumentation frameworks extended with a certain kind of weak attack relation cannot be flattened to the standard Dung argumentation frameworks. Furthermore we also briefly discuss representability of label functions with respect to the stable semantics.

The structure of the paper is as follows: in Section 2 we formally define the notion of label function and what it means to represent them as abstract argumentation frameworks. In Section 3 we show which of the twenty-seven label functions are representable and which ones are unrepresentable in the context of the complete, grounded and preferred semantics, and briefly mention the case of the stable semantics. In Section 4 we discuss the implications of these impossibility results for the flattening of a particular relation: a weak attack relation that does not propagate the undecided label. We then discuss related work in Section 5 and future work in Section 6. We provide a short conclusion in Section 7.

Due to space limitations, we assume the reader to be familiar with the labeling approach for abstract argumentation [2]. A summary of the required existing notions as well as the proofs of the results of this paper are presented in a technical report [10].

2. Label Functions

In this section we define the basic notions of a label function, an input-output argumentation framework and the representability of a label function. We write *Labs* for the set of possible labels $\{\texttt{in}, \texttt{out}, \texttt{und}\}$.

Definition 1. *A* label function *LF is a function from Labs to Labs.*

Definition 2. *Let* LF_1 *and* LF_2 *be two label functions. Then* $LF_1 \circ LF_2$ *denotes the composition of these two label functions that is defined as* $LF_1 \circ LF_2(L) = LF_1(LF_2(L))$.

We use the triplet $(LF(\texttt{in}), LF(\texttt{out}), LF(\texttt{und}))$ to refer to LF in a concise way. For example, the triplet $(\texttt{out}, \texttt{und}, \texttt{in})$ denotes the label function that maps `in` to `out`, `out` to `und` and `und` to `in`.

Definition 3. *An* input-output argumentation framework *(*I/O AF*) is a tuple* $(\mathscr{A}, \mathscr{R}, i, o)$, *where* $(\mathscr{A}, \mathscr{R})$ *is an argumentation framework and* $i, o \in \mathscr{A}$.

Definition 4. *Given an input-output argumentation framework* $G = (\mathscr{A}, \mathscr{R}, i, o)$, *with an argument* $b \notin \mathscr{A}$ *and a label* $L \in Labs$, *the* standard argumentation framework w.r.t. G and L *– denoted* $F_{st}(G, L)$ *– is the argumentation framework* $(\mathscr{A}', \mathscr{R}')$, *where* $\mathscr{A}'$ *and* $\mathscr{R}'$ *are defined through the following case distinction:*

- *If* $L = \texttt{in}$, *then* $\mathscr{A}' = \mathscr{A}$ *and* $\mathscr{R}' = \mathscr{R}$.

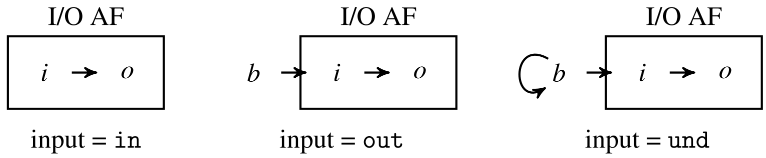

Figure 1. The three standard AFs for the I/O AF that cgp-represents the label function (out, in, und).

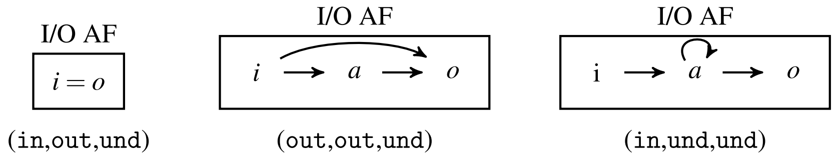

Figure 2. cgp-representation of three label functions.

- *If* $L = \texttt{out}$, *then* $\mathscr{A}' = \mathscr{A} \cup \{b\}$ *and* $\mathscr{R}' = \mathscr{R} \cup \{(b,i)\}$.
- *If* $L = \texttt{und}$, *then* $\mathscr{A}' = \mathscr{A} \cup \{b\}$ *and* $\mathscr{R}' = \mathscr{R} \cup \{(b,b),(b,i)\}$.

Definition 5. *Let* σ *be an argumentation semantics. An input-output argumentation framework* G represents *a label function* LF *w.r.t.* σ *iff for every* $L \in Labs$, $\sigma(F_{st}(G,L)) \neq \emptyset$ *and for every labeling* $\texttt{Lab} \in \sigma(F_{st}(G,L))$, $\texttt{Lab}(i) = L$ *and* $\texttt{Lab}(o) = LF(L)$.

Definition 6. *Let* σ *be an argumentation semantics. A label function* LF *is called* σ-representable *iff there is some input-output argumentation framework* G *that represents* LF *w.r.t.* σ.

In this work, we shall focus on three of the most well-known semantics, namely complete, grounded and preferred. The principles that these semantics satisfy make them the most appropriate to start with.

Definition 7. *We define* cgp *to be the set of semantics* {*complete, grounded, preferred*}. *If a label function can be* σ*-represented for every* $\sigma \in$ cgp, *we say that the function is* cgp*-representable. Similarly, if a label function cannot be* σ*-represented for any* $\sigma \in$ cgp, *we say that the function is* cgp*-unrepresentable.*

Example 1. *Consider the label function* (out, in, und) *which maps* in *to* out *and vice-versa, leaving* und *as it is. This function can be* cgp*-represented as depicted in Fig. 1. By having the input directly attack the output, when the input is* in, *it forces the output to be* out. *Conversely, when the input is* out, *there is no attacker of the output left, so it must be* in. *And finally when the input is* und, *the undecided label propagates to the output.*

Example 2. *Fig. 2 depicts three I/O AFs that* cgp*-represent the label functions* (in, out, und), (out, out, und) *and* (in, und, und) *respectively. Note that the I/O AF that represents the identity function* (in, out, und) *consists only of a single argument, so that the input argument* i *and the output argument* o *are the same argument.*

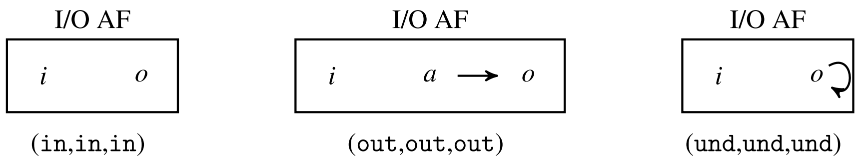

Figure 3. cgp-representation of the three constant label functions.

We now define how two input-output argumentation frameworks can be composed into a single one. The intuitive idea is that the output of the first I/O AF is used as input for the second I/O AF.

Definition 8. *Let $G_1 = (\mathscr{A}_1, \mathscr{R}_1, i_1, o_1)$ and $G_2 = (\mathscr{A}_2, \mathscr{R}_2, i_2, o_2)$ be two input-output argumentation frameworks with $\mathscr{A}_1 \cap \mathscr{A}_2 = \emptyset$, and let $c \notin \mathscr{A}_1 \cup \mathscr{A}_2$. Then we define $G_1 \oplus G_2$ to be the input-output argumentation framework $(\mathscr{A}_1 \cup \mathscr{A}_2 \cup \{c\}, \mathscr{R}_1 \cup \mathscr{R}_2 \cup \{(o_1, c)\} \cup \{(c, i_2)\}, i_1, o_2)$.*

The following theorem establishes that composed AFs represent composed label functions with respect to the complete, grounded and preferred semantics.

Theorem 1. *Let LF_1 and LF_2 be representable label functions, and let $G_1 = (\mathscr{A}_1, \mathscr{R}_1, i_1, o_1)$ and $G_2 = (\mathscr{A}_2, \mathscr{R}_2, i_2, o_2)$ be input-output argumentation frameworks that represent LF_1 and LF_2 respectively. Then $G_1 \oplus G_2$ cgp-represents $LF_2 \circ LF_1$.*

The following corollary directly follows from Theorem 1

Corollary 1. *If LF_1 and LF_2 are cgp-representable, then $LF_1 \circ LF_2$ is cgp-representable.*

3. Representability of Label Functions

In this section, we will categorize the twenty seven label functions into eleven functions that are cgp-representable and sixteen functions that are not cgp-representable.

As we will show below, a label function is cgp-representable iff it is either a constant function or maps und to und. This motivates the following definition:

Definition 9. *We define the set* Rep *as the following set of label functions:*

$$\texttt{Rep} = \{(\texttt{in}, \texttt{in}, \texttt{in}), (\texttt{out}, \texttt{out}, \texttt{out})\} \cup \{(l, l', \texttt{und}) \mid l, l' \in \mathit{Labs}\}$$

Theorem 2. *Every function in* Rep *is* cgp-*representable.*

The following theorem establishes that the sixteen label functions not included in Rep are actually cgp-unrepresentable.

Theorem 3. *The sixteen label functions not in* Rep *are* cgp-*unrepresentable.*

Aside from the widely used semantics included in the set cgp, the stable semantics is another well-known semantics which is also complete-based. Notice however that the stable semantics does not allow for any und arguments, and thus no framework could stable-represent a label function as defined in Def. 5, since having und as input would automatically mean there is no extension in the corresponding standard AF, so no output could be given. We can however define a similar notion over 2-valued labelings, i.e. restricting the functions to only two possible inputs and outputs: in and out.

This restriction leaves us with only four different possible label functions, and an interesting small result is that all of these are stable-representable. $(\texttt{out},\texttt{in})$ is stable-represented by the I/O AF in Figure 1 and $(\texttt{in},\texttt{out})$ by the I/O AF on the left in Figure 2. $(\texttt{in},\texttt{in})$ and $(\texttt{out},\texttt{out})$ are stable-represented by the I/O AFs in Figure 3, respectively on the left and in the middle.

Proposition 1. *The four 2-valued label functions* $(\texttt{in},\texttt{out}),(\texttt{out},\texttt{in}),(\texttt{in},\texttt{in})$ *and* $(\texttt{out},\texttt{out})$ *are all stable-representable.*

4. Impossibility of Flattening Weak Attacks

Various extensions of argumentation frameworks have been studied in the literature. One fruitful approach to studying such extensions is the flattening methodology, in which extensions of argumentation frameworks are mapped to standard argumentation frameworks through a flattening function that is faithful with respect to the semantics of the extended argumentation frameworks. Some explanations about this flattening approach and existing work applying it to argumentation frameworks with a support relation can be found in the technical report.

In this section we show how the theory of label functions can be used prove impossibility results concerning flattenings of certain extensions of argumentation frameworks, namely frameworks with a weak attack relation additionally to the standard attack relation. Note that for the formal definition of an extended framework, it is irrelevant whether the second relation that gets added to the standard attack relation is a relation of support or a second attack relation. This motivates the following definitions:

Definition 10. *A* two-relation framework *is a triple* $(\mathscr{A},\mathscr{R},\mathscr{T})$ *such that* $\mathscr{R}\subseteq\mathscr{A}\times\mathscr{A}$ *and* $\mathscr{T}\subseteq\mathscr{A}\times\mathscr{A}$.

Definition 11. *A* two-relation semantics *is a function* σ *that maps any two-relation framework* $B=(\mathscr{A},\mathscr{R},\mathscr{T})$ *to a set* $\sigma(B)$ *of labelings of* B*. The elements of* $\sigma(B)$ *are called* σ*-labelings of* B.

Definition 12. *Let* σ *be an argumentation semantics and let* σ' *be a two-relation semantics. We say that* σ' extends σ *iff for every two-relation framework* $B=(\mathscr{A},\mathscr{R},\mathscr{T})$ *with* $\mathscr{T}=\emptyset$, $\sigma'(B)=\sigma((\mathscr{A},\mathscr{R}))$.

We want flattenings to be defined in a local way, which we formalize as follows:

Definition 13. *Let* $B=(\mathscr{A},\mathscr{R},\mathscr{T})$ *be a two-relation framework, and let* $G=(\mathscr{A}',\mathscr{R}',i,o)$ *be an I/O AF. The* G-flattening of B *is the AF* $flat_G(B)=(\mathscr{A}^*,\mathscr{R}^*)$*, where* $\mathscr{A}^*:=\mathscr{A}\cup\{(a,b,c)\mid(a,b)\in\mathscr{T}\text{ and }c\in\mathscr{A}'\setminus\{i,o\}\}$ *and* $\mathscr{R}^*:=\mathscr{R}\cup\{((a,b,c),(a,b,c'))\mid(a,b)\in\mathscr{T},(c,c')\in\mathscr{R}'\text{ and }c,c'\notin\{i,o\}\}\cup\{(a,(a,b,c))\mid(a,b)\in\mathscr{T}\text{ and }(i,c)\in\mathscr{R}'\}\cup\{((a,b,c),a)\mid(a,b)\in\mathscr{T}\text{ and }(c,i)\in\mathscr{R}'\}\cup\{(b,(a,b,c))\mid(a,b)\in\mathscr{T}\text{ and }(o,c)\in\mathscr{R}'\}\cup\{((a,b,c),b)\mid(a,b)\in\mathscr{T}\text{ and }(c,o)\in\mathscr{R}'\}$.

Definition 14. *Let σ be an argumentation semantics and let σ' be a two-relation semantics that extends σ. We say that σ'* admits a uniform local flattening w.r.t. *σ iff there exists an I/O AF G such that for every two-relation argumentation framework B, $\sigma'(B) = \sigma(flat_G(B))$.*

We now consider a way of interpreting two-relation frameworks in which the second relation is not a support relation, but rather a *weak attack* relation. The intention behind our notion of a weak attack is that when an argument *a* is weakly attacked by an argument *b*, one can accept *a* without being able to defend *a* against the weak attack from *b*, but that in all other respects (such as conflict-freeness), weak attacks behave like the standard attacks of abstract argumentation, which we from now on call *strong attacks* to distinguish them clearly from weak attacks. In the labeling-based approach this can be formalized as follows (the abbreviation "s/w" stands for "strong/weak"):

Definition 15. *Let $B = (\mathscr{A}, \mathscr{R}, \mathscr{T})$ be a two-relation framework, and let* `Lab` *be a labeling of B.*

- *An argument $a \in \mathscr{A}$ is called* s/w-legally `in` w.r.t. `Lab` *iff every argument that strongly attacks a is labeled* `out` *by* `Lab` *and every argument that weakly attacks a is labeled either* `out` *or* `und`.
- *An argument $a \in \mathscr{A}$ is called* s/w-legally `out` w.r.t. `Lab` *iff some argument that strongly or weakly attacks a is labeled* `in` *by* `Lab`.
- *An argument $a \in \mathscr{A}$ is called* s/w-legally `und` w.r.t. `Lab` *iff no argument that strongly or weakly attacks a is labeled* `in` *by* `Lab` *and some argument that strongly attacks a is labeled* `und` *by* `Lab`.

Now we define the semantics for two-relation frameworks with strong and weak attacks analogously as for standard AFs:

Definition 16. *Let $B = (\mathscr{A}, \mathscr{R}, \mathscr{T})$ be a two-relation framework, and let* `Lab` *be a labeling of B.*

- `Lab` *is an* s/w-complete labeling *of B iff every argument that* `Lab` *labels* `in` *is s/w-legally* `in` *w.r.t.* `Lab`*, every argument that* `Lab` *labels* `out` *is s/w-legally* `out` *w.r.t.* `Lab`*, and every argument that* `Lab` *labels* `und` *is s/w-legally* `und` *w.r.t.* `Lab`.
- `Lab` *is an* s/w-grounded labeling *of B iff* `Lab` *is an s/w-complete labeling of B in which the set of* `in`*-labeled arguments is minimal w.r.t. set inclusion.*
- `Lab` *is an* s/w-preferred labeling *of B iff* `Lab` *is an s/w-complete labeling of B in which the set of* `in`*-labeled arguments is maximal w.r.t. set inclusion.*

One can easily see that these three semantics extend the corresponding semantics of standard AFs.

The following theorem establishes that the weak attack relation cannot be flattened to the strong attack relation in a uniform local way:

Theorem 4. *Let $\sigma \in$* `cgp`*. Then s/w-σ does not admit a uniform local flattening w.r.t. σ.*

5. Related Work

In the work of Baroni et al. [1], a similar methodology is introduced, where argumentation frameworks are partitioned, allowing for partitions to be evaluated locally. This local evaluation function needs to condition on the potential statuses of attackers from outside the partition, but does not need to consider the whole rest of the framework. From their results on decomposability of semantics, one could derive a result similar to our Theorem 1 but restricted to finite argumentation frameworks. We however chose to consider infinite argumentation frameworks as well in our work, as it grants more weight to the unrepresentability result derived in Section 3.

The work of Rienstra et al. [14] considers the partitioning of argumentation frameworks such that different semantics are applied to different partitions. In these cases, when evaluating the acceptance status of arguments within a partition, only the outside arguments which are the source of an attack targeting an argument inside that partition need to be considered, using a similar input/output methodology.

Enrichments of argumentation frameworks, such as the AFRA [3] and the BAF [9] have been interpreted in some cases using a flattening approach [7,6] which expresses higher-level relations in terms of auxiliary arguments and attacks, which can replace the original relation in a local fashion. Our results would prove useful when devising flattenings for existing or future enrichments, or showing no such flattening is possible.

6. Future Work

In future work, one could generalize the concept of a label function by dropping the requirement that the output argument always has the same label; these generalized label functions would therefore have a set of possible labels as their output value. Additionally one could drop the distinction between input argument and output argument, thus allowing an external effect on both arguments and looking at the set of label pairs that these two arguments may take over the different extensions. This would yield to a generalized theory of binary relations between arguments that have a local effect expressible in the 3-label approach. While there are only 27 label functions, the number of such different relations between arguments is 2^{36}, so the classification according to their representability is likely to be much more complex. Such a classification would allow one to extend the impossibility result from Section 4 to other enrichments of abstract argumentation frameworks, or provide insights on how to flatten new enrichments.

Another line of future work would be to investigate the representability with respect to other semantics such as semi-stable [8], stage [15], stage2 [13], CF2 [4], and the more recent SCF2 [11] and weakly complete [5]. Some preliminary findings for representability with respect to the semi-stable semantics can be found in a technical report [10].

7. Conclusion

In this paper, we formally introduce argumentation label functions, and address the question of which functions are representable with an argumentation framework, focusing on the complete, grounded and preferred semantics, for which the labeling approach has been widely studied. We provide a proof that two representations of label functions can be composed to yield the composed label function, and use this finding to categorize the twenty seven label functions into eleven label functions that are representable and sixteen that are unrepresentable with respect to these three semantics. We also briefly investigate the case of the stable semantics, which is quite straightforward since it only allows for two different labels. We then discuss how the label function approach can be used to prove an impossibility result about the flattening approach for enrichments of abstract argumentation frameworks.

References

[1] Pietro Baroni, Guido Boella, Federico Cerutti, Massimiliano Giacomin, Leendert Van Der Torre, and Serena Villata. On the input/output behavior of argumentation frameworks. *Artificial Intelligence*, 217:144–197, 2014.

[2] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[3] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. Afra: Argumentation framework with recursive attacks. *International Journal of Approximate Reasoning*, 52(1):19–37, 2011.

[4] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168(1):162–210, 2005.

[5] Ringo Baumann, Gerhard Brewka, and Markus Ulbricht. Revisiting the foundations of abstract argumentation–semantics based on weak admissibility and weak defense. *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (2020)*, 2020.

[6] Guido Boella, Dov M Gabbay, Leendert van der Torre, and Serena Villata. Meta-argumentation modelling I: Methodology and techniques. *Studia Logica*, 93(2-3):297–355, 2009.

[7] Guido Boella, Dov M Gabbay, Leendert WN van der Torre, and Serena Villata. Support in Abstract Argumentation. *COMMA*, 216:111–122, 2010.

[8] Martin Caminada. Semi-stable semantics. In *Proceedings of the 2006 conference on Computational Models of Argument: Proceedings of COMMA 2006*, pages 121–130. IOS Press, 2006.

[9] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 378–389. Springer, 2005.

[10] Marcos Cramer and Jérémie Dauphin. Argumentation Label Functions - Technical Report. https://orbilu.uni.lu/bitstream/10993/43078/1/report.pdf. University of Luxembourg, 2020.

[11] Marcos Cramer and Leendert van der Torre. SCF2 – an argumentation semantics for rational human judgments on argument acceptability. *Proceedings of the 8th Workshop on Dynamics of Knowledge and Belief (DKB-2019) and the 7th Workshop KI and Kognition (KIK-2019)*, 2019.

[12] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[13] Wolfgang Dvořák and Sarah Alice Gaggl. Stage semantics and the scc-recursive schema for argumentation semantics. *Journal of Logic and Computation*, 26(4):1149–1202, 2014.

[14] Tjitze Rienstra, Alan Perotti, Serena Villata, Dov M Gabbay, and Leendert van der Torre. Multi-sorted argumentation. In *International Workshop on Theorie and Applications of Formal Argumentation*, pages 215–231. Springer, 2011.

[15] Bart Verheij. Two Approaches to Dialectical Argumentation: Admissible Sets and Argumentation Stages. In *In Proceedings of the biannual International Conference on Formal and Applied Practical Reasoning (FAPR) workshop*, pages 357–368. Universiteit, 1996.