

Odd or Even: Handling N-lemmas in a Dynamic Argumentation Framework

Kazuko Takahashi^{1,*}

¹Kwansei Gakuin University, 1 Gakuen Uegahara, Sanda, 669-1330, Japan

Abstract

This paper discusses a dynamic argumentation framework (AF) that consists of odd-length cycles. When an AF has no stable extension, we investigate the position to which an argument is added so that it has a stable extension. We prove the unstability of an AF that consists of only odd-length cycles with some constraints, show the conditions for repairability using a reduced AF, and identify the positions. In contrast to existing studies on dynamic argumentation, we use a topological approach. In addition, we introduce a reduced form of an AF, which provides a proof that is easier to understand and reduces the computational burden.

Keywords

dynamic argumentation framework, semantics, enforcing problem, graph topology

1. Introduction

There have been lots of studies of computational argumentation [1] since Dung first introduced an abstract argumentation framework (AF) [2]. Among these studies, dynamic argumentation that investigates the changes in accepted arguments by adding or removing arguments or attacks based on various semantics, has attracted substantial attention [3, 4, 5, 6, 7, 8]. The possibility of modification such that a desired set of arguments becomes a subset of an extension was introduced as an *enforcing problem* and was first discussed in [9]. Subsequently, considerable work has been done on this problem [10].

We often use argumentation to reach an agreement or to persuade others, and it is preferable to obtain a justified result. In this paper, we consider argumentation that becomes stuck and yields no outcome because of its cyclic structure, such as the structure in Figure 1, in which nodes represent arguments and edges represent counter-arguments. In this situation, an agent usually wants to persuade other agents to accept their claim by making one new counter-argument.

In Figure 1(a), if an agent wants their argument a to be accepted, they attack an argument c with a new argument, then c is defeated by the new argument, and as a result, a and the new argument are accepted. This is a unique solution (with respect to stable semantics), and it is easy to find such an attack position. In Figure 1(b), not only c but also d can be regarded as a position

SAFA'22: Fourth International Workshop on Systems and Algorithms for Formal Argumentation 2022, September 13, 2022, Cardiff, Wales, United Kingdom

*Corresponding author.

✉ ktaka@kwansei.ac.jp (K. Takahashi)

🌐 <https://ist.ksc.kwansei.ac.jp/~ktaka/LABO/> (K. Takahashi)



© 2022 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)

so that an argument a is accepted. Therefore, even if an agent does not find a new argument that attacks c , they can achieve their goal if they find a new argument that attacks d . However, it is difficult to rapidly identify the position when the AF is large. We identify such a position by considering the topology of an AF.

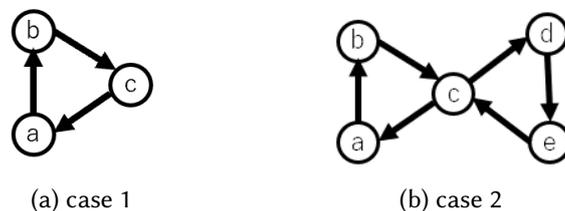


Figure 1: Examples of stuck argumentation.

We consider a problem to resolve the stuck argumentation that consists of n -lemmas (i.e., odd-length cycles), through the addition of one new argument. We mainly discuss the position in which a new argument is added.

We discussed this problem in the case of a trilemma in our previous paper [11]. There, we showed that the topology of the connectors (i.e., nodes that are attacked by more than one argument) plays an important role. In this paper, we extend the discussion to general n -lemmas. In contrast to the case of trilemma, there are more types of configurations of connectors when handling n -lemmas. We introduce the concept of a reduction, in which an AF is summarized with the semantics of the connectors preserved; we discuss the stability and reparability of a given AF, based on its reduced AF. Then, we derive other solutions, namely, positions that do not appear in the reduced AF.

This paper is organized as follows. In Section 2, we describe the basic concepts. In Section 3, we introduce the concept of reduction and consider the stability of the reduced AF. In Section 4, we explore the positions to which a counter-argument is added to a given AF, based on the solution on its reduced AF. In Section 5, we compare our approach with related work. Finally, in Section 6, we present our conclusions and discuss future work.

2. Preliminaries

The abstract AF, proposed by Dung [2], is a representation of an argumentation structure that ignores its content.

Definition 1 (argumentation framework (AF)). An Argumentation Framework (AF) \mathcal{F} is defined as a pair $\langle \mathcal{A}, \mathcal{R} \rangle$ where \mathcal{A} is a set of arguments and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$.

A pair $(A, B) \in \mathcal{R}$ is called an *attack*, and it is said that A attacks B .

An AF can be represented as a graph in which each node corresponds to an argument, and each edge corresponds to an attack. In this paper, we consider a finite AF.

Definition 2 (path, cycle). For a pair of arguments A_0 and A_n of an AF, if there exists a sequence of attacks $(A_0, A_1), (A_1, A_2), \dots, (A_{n-1}, A_n)$ where $A_i \neq A_j$ ($\forall i, j; 0 \leq i \neq j \leq n, n \neq 0$),

then $\langle A_0, \dots, A_n \rangle$ is called a path from A_0 to A_n . If $A_n = A_0$ ($n \neq 0$), then it is called a cycle from A_0 to A_0 .

Definition 3 (odd/even-length-far). For a path $\langle A_0, \dots, A_n \rangle$, if i ($0 < i \leq n$) is odd, A_i is called a node that is odd-length-far from A_0 in the path; and if i is even, a node that is even-length-far from A_0 in the path.

For an abstract AF, semantics is defined either by an extension or labeling, which has a one-to-one relation [12].

A labeling is a total function from a set of arguments to a set of labels $\{in, out, undec\}$.

Definition 4 (complete labeling). Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an AF. Labeling \mathcal{L} is said to be complete if the following conditions are satisfied for any argument $A \in \mathcal{A}$.

- $\mathcal{L}(A) = in$ iff $\forall B \in \mathcal{A}; (B, A) \in \mathcal{R} \Rightarrow \mathcal{L}(B) = out$.
- $\mathcal{L}(A) = out$ iff $\exists B \in \mathcal{A}; \mathcal{L}(B) = in \wedge (B, A) \in \mathcal{R}$.

Hereafter, the term “labeling” denotes a complete labeling unless otherwise indicated.

The set $\{A \mid A \in \mathcal{A}, \mathcal{L}(A) = in\}$ is a set of accepted arguments corresponding to an extension in the extension-based semantics. The label of an argument is determined only by the labels of its direct attackers.

We use the following notations hereafter.

For a label L , we define $\neg L$ as follows:

$$\neg L = \begin{cases} in & (\text{if } L = out) \\ out & (\text{if } L = in) \\ undec & (\text{if } L = undec) \end{cases}$$

For labels L_1 and L_2 , we define $L_1 \wedge L_2$ as follows:

$$L_1 \wedge L_2 = \begin{cases} in & (\text{if } ((L_1 = in \wedge L_2 = out) \vee (L_1 = out \wedge L_2 = in) \vee (L_1 = in \wedge L_2 = in))) \\ out & (\text{if } L_1 = out \wedge L_2 = out) \\ undec & (\text{otherwise}) \end{cases}$$

In this paper, we adapt semantics by labeling and regard stable semantics as the most natural semantics for our purpose. An AF may not have a stable extension when it includes an odd-length cycle, and some semantics were introduced to solve this problem. For example, CF2 semantics [13] gives three extensions $\{a\}, \{b\}, \{c\}$ to the AF shown in Figure 1(a). However, in some ways, stable semantics most closely reflects the situation in which the other attendants agree in a practical argumentation, considering that each argument not in a stable extension is attacked by an argument in a stable extension.

Definition 5 (stable labeling). Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an AF. For a complete labeling \mathcal{L} , if $\{A \mid A \in \mathcal{A}, \mathcal{L}(A) = undec\} = \emptyset$, then it is called a stable labeling.

Definition 6 (stable AF). An AF with a stable labeling is called a stable AF, and one without it is called an unstable AF.

In addition to these concepts, we introduce several new concepts and terminology.

Definition 7 (annihilator). *For a new argument P and an attack I from P , a pair $\langle P, I \rangle$ is said to be an annihilator¹.*

Definition 8 (entrance). *For an AF \mathcal{F} , the argument to which an annihilator is added is called an entrance of \mathcal{F} .*

Definition 9 (repair). *For an unstable AF, the act of revision by adding a single annihilator to obtain a stable AF is called repair², and the original AF is said to be repairable. An AF that cannot be repaired by taking any argument as an entrance is said to be unrepairable.*

An entrance is always labeled *out* in the repaired AF, according to the definition of stable labeling.

3. Stability and Repairability

3.1. A target AF

We investigate an AF that consists of odd-length cycles.

Definition 10 (unit, n-lemma). *An AF of the form $\langle \{A_1, A_2, \dots, A_{2n-1}\}, \{(A_1, A_2), (A_2, A_3), \dots, (A_{2n-2}, A_{2n-1}), (A_{2n-1}, A_1)\} \rangle$ where $n \geq 1$ is called a unit. For a specific number n , it is called an n-lemma.*

An AF $\langle \{A\}, \{(A, A)\} \rangle$ is a 1-lemma, which contains a single node with a self-loop.

When an AF includes several units, some arguments are attacked by more than one argument. Such arguments play important roles in judging repairability.

Definition 11 (connector). *Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF. For an argument $C \in \mathcal{A}$, if there exists more than one argument A such that $(A, C) \in \mathcal{R}$, then C is said to be a connector of \mathcal{F} ; if there exists a unique argument A such that $(A, C) \in \mathcal{R}$, then C is said to be a non-connector of \mathcal{F} .*

Definition 12 (source/terminal/isolated connector). *Let C be a connector of an AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$. If there does not exist a connector $C' (\neq C)$ such that $(C', C) \in \mathcal{R}$, and there exists a connector $C'' (\neq C)$ such that $(C, C'') \in \mathcal{R}$, then C is said to be a source connector of \mathcal{F} . If there exists a connector $C' (\neq C)$ such that $(C', C) \in \mathcal{R}$, and there does not exist a connector $C'' (\neq C)$ such that $(C, C'') \in \mathcal{R}$, then C is said to be a terminal connector of \mathcal{F} . If there exists neither a connector $C' (\neq C)$ such that $(C', C) \in \mathcal{R}$, nor a connector $C'' (\neq C)$ such that $(C, C'') \in \mathcal{R}$, then C is said to be an isolated connector of \mathcal{F} .*

Example 1. *In Figure 2, C is the source connector, D is the terminal connector, E is the isolated connector, and the other nodes are non-connectors. Generally, there may be multiple terminal/source/isolated connectors, although this example has one of each.*

¹In [11], we used a name “whisker.” The term annihilator is named after Gabbay’s paper [14]. He used this term to indicate the node that breaks a loop.

²The meaning of “repair” is distinct from the meaning used in [9].

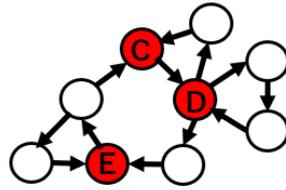


Figure 2: Connectors.

Generally, let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF and \mathcal{L} be its complete labeling. If $C \in \mathcal{A}$ is a connector that is attacked by k arguments, that is, $(A_1, C), (A_2, C) \dots, (A_k, C) \in \mathcal{R}$, then $\mathcal{L}(C) = \neg(\dots (\mathcal{L}(A_1) \wedge \mathcal{L}(A_2)) \wedge \dots) \wedge \mathcal{L}(A_k)$. If $D \in \mathcal{A}$ is a non-connector that is attacked by an argument E , that is, $(E, D) \in \mathcal{R}$, then $\mathcal{L}(D) = \neg \mathcal{L}(E)$.

We clarify our target AF. In a practical stuck argumentation, interactions of cyclic arguments are observed but messy attack relations are uncommon. We set the target AF to reflect this situation.

Definition 13. An AF that satisfies the following [Cond] is called a target AF.

[Cond]

1. It includes m -units ($m \geq 1$).
2. Each node is included in at least one unit.
3. Each edge is included in at least one unit.
4. It includes no even-length cycle.
5. It is uncontroversial, that is, there exist no arguments A and B connected by two different paths of even-length and odd-length.
6. No cycle consists only of connectors.

Note that a target AF does not always consist of the same n -lemmas. It may include, for example, 3-lemma and 5-lemma simultaneously. In addition, because of the third condition, we have only one stable labeling after repair.

We seldom encounter controversial argumentation in a practical argumentation, since it is unnatural that an argument A (in)directly attacks another argument B , and at the same time A (in)directly supports B . In such a case, we can divide B into several arguments considering that A attacks some aspect of B and supports another aspect of B .

The last condition excludes any AF that has a cycle containing only connectors, since both stability and reparability of such AFs depend on the configuration of the connectors. We do not discuss this issue here as we want to focus on repair of an unstable AF.

We have proven the following properties when a target AF consists of only trilemmas [11].

Theorem 1. If an AF including only trilemmas satisfies the condition [Cond], then the following properties hold.

1. It is unstable.

2. It is repairable if there is only one connector or there is only one terminal connector.
3. Each connector is labeled *out* in the repaired AF.
4. When there is one connector, the connector and the node of an odd-length-far from it that does not attack itself are the entrances.
5. When there is more than one connector, the terminal connector and the node of an odd-length-far from the terminal connector that does not attack a connector are the entrances.

3.2. Reduction

We investigate the properties corresponding to Theorem 1 in the general case of n-lemma.

In the case of a trilemma, the connectors are always connected; thus, there always exist a source connector and a terminal connector when there are multiple connectors. However, there may exist an isolated connector in the general case of an n-lemma.

It is difficult to understand the structure of a large and complicated AF, and it is computationally burdensome to explore its repairability directly. We introduce the reduced form of a target AF, while preserving the labels of connectors, because the label of the connector is the key to considering stability. Because two succeeding non-connectors in a path to a connector do not affect the label of the connector, we can reduce a sequence of non-connectors.

Definition 14 (cpath). *Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF and C, D be (possibly the same) connectors of \mathcal{F} . Then a path $\langle C, A_1, \dots, A_k, D \rangle$ where A_1, \dots, A_k are non-connectors is said to be a cpath(k) from C to D , where k represents the number of nodes between C and D .*

Definition 15 (reduction). *Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF. For each pair of connectors C_p and D_p (possibly the same) in \mathcal{F} , let p be an arbitrary cpath(k) between these connectors. We define three sets S_p , T_p , and U_p : $S_p = \{A_1, \dots, A_k\}$, $T_p = \{(C_p, A_1), (A_1, A_2), \dots, (A_k, D_p)\}$, and $U_p = \{(C_p, D_p)\}$ if k is even, $U_p = \{(C_p, E_p), (E_p, D_p)\}$ where E_p is a new argument if k is odd.*

Set $S = \bigcup_p S_p$, $T = \bigcup_p T_p$, and $U = \bigcup_p U_p$; and $\mathcal{A}' = \mathcal{A} \setminus S$, $\mathcal{R}' = (\mathcal{R} \setminus T) \cup U$. Then $\mathcal{F}' = \langle \mathcal{A}', \mathcal{R}' \rangle$ is called a reduced form of \mathcal{F} .

At least one connector in a target AF remains a connector in the reduced AF, but not all of them may remain as connectors in the reduced AF.

Example 2. *Figure 3 shows target AFs and their reduced forms. In the figures, connectors are shown as red-colored nodes. Both connectors of \mathcal{F}_1 remain connectors of \mathcal{F}'_1 (Figure 3(a)), while one connector of \mathcal{F}_2 becomes a non-connector of \mathcal{F}'_2 (Figure 3(b)).*

Proposition 1. *Let $\mathcal{F}' = \langle \mathcal{A}', \mathcal{R}' \rangle$ be a reduced form of a target AF. Then \mathcal{F}' is unstable.*

Proof. Let \mathcal{L}' be a labeling of \mathcal{F}' .

If there is one connector C in \mathcal{F}' , then only odd-length cycles exist from C to C . Therefore, $\mathcal{L}'(C) = \text{undec}$. Thus, it is unstable.

If there is more than one connector in \mathcal{F}' , let C be any terminal connector. There exists a cycle $\langle C, E, D, C \rangle$ where D is another connector and E is a non-connector. If $\mathcal{L}'(C) = \text{out}$, then $\mathcal{L}'(E) = \text{in}$, $\mathcal{L}'(D) = \text{out}$, whereas if $\mathcal{L}'(C) = \text{in}$, then $\mathcal{L}'(E) = \text{out}$, $\mathcal{L}'(D) = \text{in}$. Both are contradictions, since C is a terminal connector that has an odd-length cycle from C to C . Therefore, $\mathcal{L}'(C) = \text{undec}$. Thus, \mathcal{F}' is unstable. \square

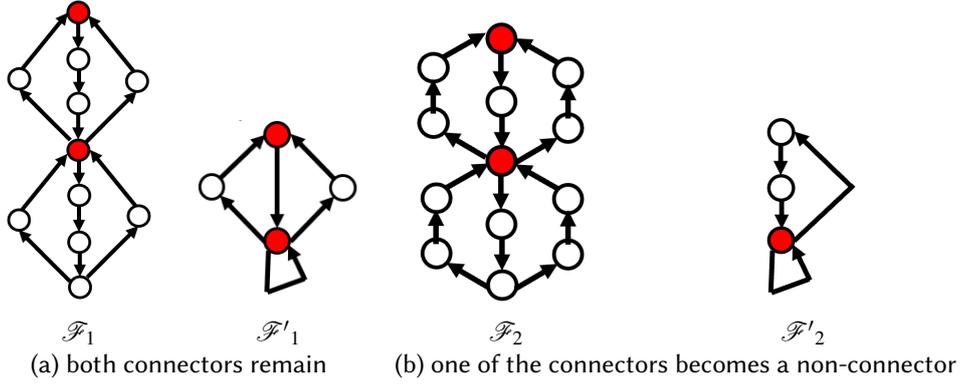


Figure 3: A target AF and its reduced form.

Proposition 2. *A target AF \mathcal{F} is unstable.*

Proof. Let \mathcal{F}' be a reduced form of \mathcal{F} , \mathcal{L} and \mathcal{L}' be labelings of \mathcal{F} and \mathcal{F}' , respectively.

Let C, C_1 and C_2 be connectors in \mathcal{F} , where C may coincide with C_1 or with C_2 .

For simplicity, assume that there exist exactly two paths to C in \mathcal{F} : an odd-length path p from C_1 and an even-length path q from C_2 . Similar discussion is available if there are more than two paths, or only an odd/even-length path exists.

Let p be $\text{cpath}(2k)$ from C_1 to C and q be $\text{cpath}(2h+1)$ from C_2 to C in \mathcal{F} , respectively. Then, p is reduced to $\langle C_1, C \rangle$ and q is reduced to $\langle C_2, E, C \rangle$ in \mathcal{F}' , where E is a new node that is a non-connector in \mathcal{F}' .

Figure 4 illustrates a reduction and labeling. In this figure, pink nodes and blue nodes denote arguments labeled *in* and *out*, respectively.

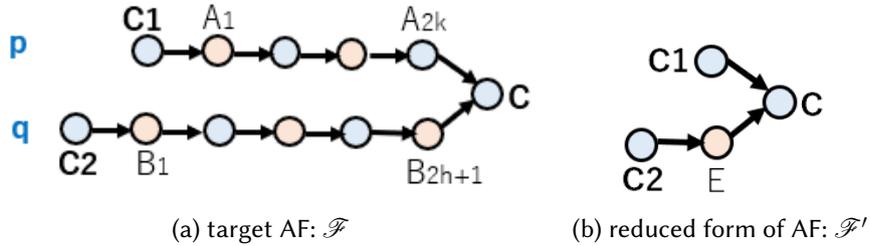


Figure 4: Reduction and labeling.

Then, $\mathcal{L}(C) = \neg(\mathcal{L}(A_{2k}) \wedge \mathcal{L}(B_{2h+1})) = \neg(\mathcal{L}(C_1) \wedge \neg\mathcal{L}(C_2))$.

On the other hand, $\mathcal{L}'(C) = \neg(\mathcal{L}'(C_1) \wedge \mathcal{L}'(E)) = \neg(\mathcal{L}'(C_1) \wedge \neg\mathcal{L}'(C_2))$.

This shows that the relationship of the label of connector C and the labels of connectors C_1 and C_2 is the same in both \mathcal{F} and \mathcal{F}' .

Since all connectors in \mathcal{F}' are labeled *undec* by Proposition 1, all connectors in \mathcal{F} are also labeled *undec*.

Therefore, \mathcal{F} is unstable. \square

Proposition 3. *Let $\mathcal{F}' = \langle \mathcal{A}', \mathcal{R}' \rangle$ be a revised AF of the reduced form of a target AF by taking a connector as an entrance. \mathcal{F}' is stable iff all its connectors are labeled out.*

Proof. Let \mathcal{L}' be a labeling of \mathcal{F}' .

(\Rightarrow) If there is only one connector in \mathcal{F}' , then it is labeled *out*. If there is more than one connector, then there exist connectors C and D , such that $(C, D) \in \mathcal{R}'$ and $(D, E), (E, C) \in \mathcal{R}'$ where E is a non-connector in \mathcal{F}' . Assume that there exists a connector labeled *in*. First, assume that $\mathcal{L}'(C) = \textit{in}$, then $\mathcal{L}'(D) = \textit{out}$, $\mathcal{L}'(E) = \textit{in}$, that is a contradiction. Next, assume that $\mathcal{L}'(D) = \textit{in}$, then $\mathcal{L}'(E) = \textit{out}$, $\mathcal{L}'(C) = \textit{out}$; in this case, there should exist a node labeled *in* that attacks C ; however, there is no such node for a source connector, which is a contradiction. As a result, all connectors in \mathcal{F}' are labeled *out*.

(\Leftarrow) Let \mathcal{L}' be a labeling of \mathcal{F}' . If $\mathcal{L}'(C_1) = \mathcal{L}'(C_2) = \textit{out}$ for all connectors C_1 and C_2 , then $\mathcal{L}'(C) = \neg(\mathcal{L}'(C_1) \wedge \neg\mathcal{L}'(C_2)) = \textit{out}$. Each connector has a path from another connector. Therefore, \mathcal{L}' is stable. \square

Proposition 4. *Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be a target AF and $\mathcal{F}' = \langle \mathcal{A}', \mathcal{R}' \rangle$ be its reduced form. When \mathcal{F}' has one connector C , \mathcal{F} is repairable by taking C as an entrance.*

Proof. When \mathcal{F}' has one connector C , C is also a connector in \mathcal{F} . Assume that C is taken as an entrance of \mathcal{F} , and let \mathcal{L} be a labeling of the revised \mathcal{F} . Then $\mathcal{L}(C) = \textit{out}$.

When \mathcal{F} includes the unique connector C , for a $\textit{cpath}(2k)$ from C to C , we label in a manner such that $\mathcal{L}(A_i) = \textit{out}$ if i ($1 \leq i \leq 2k$) is even and $\mathcal{L}(A_i) = \textit{in}$ if i is odd. Then, \mathcal{L} is stable.

When \mathcal{F} includes a connector D other than C , D becomes a non-connector in the reduction process because there is more than one $\textit{cpath}(2k)$ from C to D . Let these paths be $\langle C, A_1, \dots, A_{2k}, D \rangle$ and $\langle C, B_1, \dots, B_{2h}, D \rangle$, respectively. We label in a manner such that $\mathcal{L}(A_i) = \mathcal{L}(B_i)$ for each i ($1 \leq i \leq \min(2k, 2h)$). Then, D is attacked from A_{2k} and B_{2h} , where $\mathcal{L}(A_{2k}) = \mathcal{L}(B_{2h}) = \mathcal{L}(C) = \textit{out}$. Therefore, $\mathcal{L}(D) = \textit{in}$. Since the length of a path from D to C is odd, $\mathcal{L}(C) = \textit{out}$. Then, \mathcal{L} is stable.

Thus, \mathcal{F} is repairable by taking C as an entrance. \square

Proposition 5. *Let $\mathcal{F}' = \langle \mathcal{A}', \mathcal{R}' \rangle$ be a reduced AF and \mathcal{F}' have one terminal connector C . For each connector D other than the terminal connector, if there exists a non-connector that attacks D , then \mathcal{F}' is repairable by taking C as an entrance; otherwise, \mathcal{F}' is unrepairable.*

Proof. Let \mathcal{L}' be a labeling of the revised AF of \mathcal{F}' by taking C as an entrance. Then, $\mathcal{L}'(C) = \textit{out}$.

If $(D, C) \in \mathcal{R}'$, then $(C, E), (E, D) \in \mathcal{R}'$ where E is a non-connector; in this case, $\mathcal{L}'(E) = \textit{in}$ and $\mathcal{L}'(D) = \textit{out}$. Otherwise, there exists another connector between D and C . Assume that there exists only one connector C' in the path from D to C . Then $\mathcal{L}'(C') = \textit{out}$ from the above discussion. And we can label all the connectors in the same way. As a result, all the connectors in \mathcal{F}' are labeled *out*. Thus, \mathcal{L}' is stable from Proposition 3.

On the other hand, assume that a connector D other than a terminal connector is not attacked by a non-connector. Then it is attacked only by the connectors labeled *out*. Then, $\mathcal{L}'(D) = \textit{in}$. Thus, \mathcal{L}' is unstable from Proposition 3. \square

Proposition 6. *Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be a target AF, $\mathcal{F}' = \langle \mathcal{A}', \mathcal{R}' \rangle$ be its reduced form, and \mathcal{F}' have one terminal connector C . For each connector other than the terminal connector, if there exists a non-connector B such that $(B, C) \in \mathcal{R}'$, then \mathcal{F} is repairable by taking C as an entrance.*

This proposition is proved in a similar manner to Proposition 4.

Accordingly, we can check the repairability and entrance taking a reduced form of a target AF.

Theorem 2. *Let \mathcal{F} be a target AF.*

1. *It is unstable.*
2. *If \mathcal{F} includes one connector, then \mathcal{F} is repairable by taking that connector as an entrance.*
3. *If the reduced form of \mathcal{F} includes only one terminal connector, and for each connector other than the terminal connector, there exists a non-connector that attacks it, then \mathcal{F} is repairable by taking that terminal connector as an entrance.*

4. Entrances of AF

We have found that the (terminal) connector in the reduced AF is an entrance of the original target AF. Generally, when a target AF is repairable, there may exist several entrances that do not appear in the reduced AF. These are the nodes that are deleted in the reduction procedure but exist in the original target AF. In this section, we discuss these possibilities depending on the number of connectors.

4.1. A target AF without connectors

A target AF includes no connector when it includes a single unit. In this case, we do not consider its reduction.

Proposition 7. *A target AF without a connector is unstable. It can be repaired and each node can be taken as an entrance.*

Proof. It is trivially unstable and each node can be taken as an entrance. □

4.2. A target AF with one connector

In this case, the connector of a target AF is also a connector in the reduced AF. Several nodes that do not appear in \mathcal{F}' can also be taken as entrances.

Proposition 8. *Let \mathcal{F} be a target AF and \mathcal{F}' be its reduced form. Let \mathcal{L} be a labeling of the AF obtained from \mathcal{F} by taking a specified node as an entrance.*

When \mathcal{F} includes one connector C , let $\langle C, A_1, \dots, A_t, A_{t+1}, \dots, A_k, C \rangle$ be an arbitrary cpath(k) from C to C in \mathcal{F} , where A_1, \dots, A_t are shared with all paths from C to C .

(1) For $1 \leq i \leq k$, if A_i is odd-length-far from C , then \mathcal{F} is repaired by taking A_i as an entrance; in this case $\mathcal{L}(C) = \text{out}$.

(2) For $1 \leq j \leq t$, if A_j is even-length-far from C , then \mathcal{F} is repaired by taking A_j as an entrance; in this case $\mathcal{L}(C) = \text{in}$.

(3) If we take the other nodes as entrances, the resulting AF is not stable.

Example 3. Figure 5 shows an AF \mathcal{F} with one connector C . In the figures, pink and blue nodes show the arguments labeled in and out, respectively. There are two cycles $\langle C, a, b, d, e, C \rangle$ and $\langle C, a, b, d, f, C \rangle$. Figure 5(a) shows a stable labeling in which a node a , odd-length-far from C , is taken as an entrance; in this case $\mathcal{L}(C) = \text{out}$. Figure 5(b) shows a stable labeling in which a node b , even-length-far from C and shared with all paths from C to C , is taken as an entrance; in this case $\mathcal{L}(C) = \text{in}$. Figure 5(c) shows an unstable labeling in which a node f , even-length-far from C but not shared with all paths from C to C , is taken as an entrance.

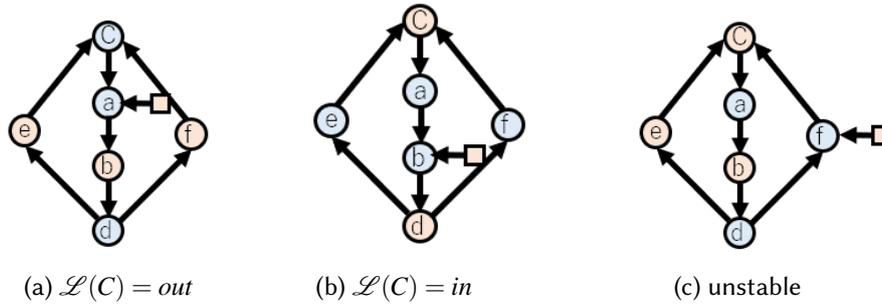


Figure 5: Position of the entrances in one-connector case (C is the connector). The rectangle with an edge denotes an annihilator.

4.3. A target AF with two connectors

Proposition 9. Let \mathcal{F} be a target AF and \mathcal{F}' be its reduced form. Let \mathcal{L} be a labeling of the AF obtained from \mathcal{F} by taking a specified node as an entrance.

Assume that \mathcal{F} includes two connectors C_1 and C_2 , where C_1 is the terminal connector in \mathcal{F}' .

- (1) When C_2 is not a connector in \mathcal{F}' , the entrances of \mathcal{F} are the same as entrances in the case of one connector.
- (2) When C_2 is a connector in \mathcal{F}' . Let $\langle C, A_1, \dots, A_k, C_1 \rangle$ be an arbitrary $\text{cpath}(k)$ from C to C_1 in \mathcal{F} , where C is either C_1 or C_2 . For $1 \leq i \leq k$, if A_i is odd-length-far from C , then \mathcal{F} is repaired by taking A_i as an entrance; in this case $\mathcal{L}(C) = \text{out}$ (Figure 6(a)(b)).
- (3) If we take the other nodes as entrances, the resulting AF is not stable (Figure 6(c)–(f)).

The proof is shown in the Appendix.

4.4. More than two connectors

Proposition 10. Let \mathcal{F} be a target AF and \mathcal{F}' be its reduced form. If \mathcal{F}' includes only one terminal connector C and for each connector C' other than the terminal connector, there exists a non-connector that attacks C' in \mathcal{F} , then an odd-length-far node A_j ($1 \leq j \leq k$) from C in an arbitrary $\text{cpath}(k)$ from C to C in \mathcal{F} , can be taken as an entrance in \mathcal{F} .

Proof. Let \mathcal{L} be a labeling of the revised AF. Then $\mathcal{L}(A_j) = \text{out}$ and $\mathcal{L}(A_k) = \text{in}$, because the length of the cycle from C to C is odd. Therefore, $\mathcal{L}(C) = \text{out}$. The labeling of the other nodes can be done similarly to the case in which C is an entrance. Thus, \mathcal{L} is stable. \square

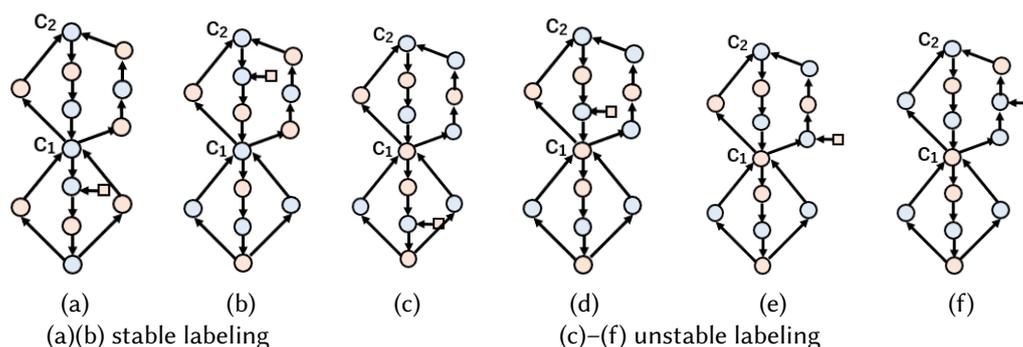


Figure 6: Position of the entrances in two-connectors case (C_1 is the terminal connector). The rectangle with an edge denotes an annihilator.

5. Related Works

Dynamic argumentation is an important and interesting topic in practical argumentation. In general, the main consideration when changing an AF is an enforcing problem [9], and there has been considerable work regarding this problem thus far. Boella et al. discussed the change in grounded semantics when adding or removing an attack relation [6, 7]. They classified the type of change and clarified its properties. Cayrol et al. expanded this discussion to several types of semantics, including stable semantics when adding or removing an argument with an attack. They first investigated a single attack and then extended the procedure to the case of multiple attacks [15, 8]. Coste-Marquis et al. addressed the revision of extension when changing an attack relation between existing arguments and when adding an argument with an attack [16]. Baumann et al. showed the minimal change required in an extension to accept a particular set of arguments [9]; specifically, they demonstrated the change in extensions under several semantics for the addition and removal of arguments and attacks [4, 5]. They investigated the changes in extensions in various cases. They also proposed the possibility of inferring syntactic information from the semantic information obtained by comparing the changes in extensions upon revision, such as the existence of an even-length cycle between a specific pair of nodes [17]. The complexities for the revision were discussed [3, 18].

These studies were concerned mainly with the problem of finding a solution to realize a minimal change in an extension, which enables a specific argument to be credulously/skeptically accepted. In contrast, we focused on the position to which a new argument is added from a topological perspective.

Cyclic arguments have attracted substantial interest with regard to AFs. Gabbay discussed a method of handling loops that appear in argumentation networks [14]. He investigated both odd-length and even-length loops and proposed a method of adding nodes called annihilators to a loop, such that an AF has an extension. However, he did not formalize the position and this is not a topological approach.

The target AFs in these related studies were more congested than ours and most allowed controversial AFs. In this paper, we ignore those topologies that are not frequently observed in

practical argumentation.

6. Conclusion

We discussed repairing AF from a topological perspective, and investigated the position to which an argument is added so that it has a stable labeling.

We proved that an AF consisting of only odd-length cycles with some constraints is unstable, then showed the conditions for repairability using a reduced AF. We also showed the entrances when it is repairable.

When a stuck argumentation is repairable, the set of accepted arguments is obtained depending on the chosen entrance. If we want to make a specific argument accepted, we should take the entrance with which the obtained set includes it. If no such entrance exists, it is impossible that the argument is accepted by adding a single counter-argument. Our approach can help to achieve this goal.

In contrast to existing work regarding dynamic argumentation, we used a topological approach. In addition, we introduced a reduced form of an AF, which provides a proof that is easier to understand and that reduces the computational burden.

In future work, we will consider the following three issues. First, we will clarify the sufficient conditions for repairability in a case of more than two connectors. Second, we will consider the complexity of finding a position that a specific argument can be accepted. Third, we will extend the discussion to other structures of AFs especially an AF including even-length cycles, as well as other semantics.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP17H06103.

References

- [1] I. Rahwan, G. R. Simari (Eds.), *Argumentation in Artificial Intelligence*, Springer, 2009.
- [2] 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. doi:10.1016/0004-3702(94)00041-X.
- [3] R. Baumann, M. Ulbricht, If nothing is accepted - repairing argumentation frameworks, in: *KR 2018*, 2018, pp. 108–117. doi:10.1613/jair.1.11791.
- [4] R. Baumann, G. Brewka, Extension removal in abstract argumentation - an axiomatic approach, in: *AAAI 2019*, 2019, pp. 2670–2677. doi:10.1609/aaai.v33i01.33012670.
- [5] R. Baumann, D. M. Gabbay, O. Rodrigues, Forgetting an argument, in: *AAAI 2020*, 2020, pp. 2750–2757. doi:10.1609/aaai.v34i03.5662.
- [6] G. Boella, S. Kaci, L. W. N. van der Torre, Dynamics in argumentation with single extensions: Abstraction principles and the grounded extension, in: *ECSQARU 2009*, 2009, pp. 107–118. doi:10.1007/978-3-642-02906-6_11.

- [7] G. Boella, S. Kaci, L. W. N. van der Torre, Dynamics in argumentation with single extensions: Attack refinement and the grounded extension, in: *AAMAS 2009*, 2009, pp. 1213–1214. doi:10.1007/978-3-642-12805-9_9.
- [8] C. Cayrol, F. D. de Saint-Cyr, M.-C. Lagasquie-Schiex, Change in abstract argumentation frameworks: Adding an argument, *J. Artif. Intell. Res.* 38 (2010) 49–84. doi:10.1613/jair.2965.
- [9] R. Baumann, G. Brewka, Expanding argumentation frameworks: Enforcing and monotonicity results, in: *COMMA 2010*, 2010, pp. 75–86. doi:10.3233/978-1-60750-619-5-75.
- [10] S. Doutre, J.-G. Maily, Constraints and changes: A survey of abstract argumentation dynamics, *Argument and Computation* 9 (2018) 223–248. doi:10.3233/AAC-180425.
- [11] K. Takahashi, T. Okubo, How can you resolve a trilemma? - a topological approach -, in: *CLAR 2021*, 2021, pp. 397–416. doi:10.1007/978-3-030-89391-0_22.
- [12] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, *Knowl. Eng. Rev.* 26 (2011) 365–410. doi:10.1017/S0269888911000166.
- [13] P. Baroni, M. Giacomin, G. Guida, Scc-recursiveness: A general schema for argumentation semantics, *Artificial Intelligence* 168 (2005) 162–210. doi:10.1016/j.artint.2005.05.006.
- [14] D. Gabbay, The handling of loops in argumentation networks, *J. of Logic and Computation* 26 (2014) 1065–1147. doi:10.1093/logcom/exu007.
- [15] C. Cayrol, F. D. de Saint-Cyr, M.-C. Lagasquie-Schiex, Revision of an argumentation system, in: *KR 2008*, 2008, pp. 124–134.
- [16] S. Coste-Marquis, S. Konieczny, J.-G. Maily, P. Marquis, Extension enforcement in abstract argumentation as an optimization problem, in: *IJCAI 2015*, 2015, pp. 2876–2882.
- [17] R. Baumann, M. Ulbricht, On cycles, attackers and supporters - a contribution to the investigation of dynamics in abstract argumentation, in: *IJCAI 2021*, 2021, pp. 1780–1786. doi:10.24963/ijcai.2021/245.
- [18] J. P. Wallner, A. Niskanen, M. Järvisalo, Complexity results and algorithms for extension enforcement in abstract argumentation, *J. Artif. Intell. Res.* 60 (2017) 1–40. doi:10.1613/jair.5415.

Appendix.

Proof for Proposition 9.

Proof. (1) There does not exist a cycle from C_2 to C_2 , since C_2 is not a connector in \mathcal{F}' . Therefore, we do not necessarily make $\mathcal{L}(C_2) = out$, and C_2 can be considered as a non-connector in \mathcal{F} .

(2-1) If A_i is an entrance, then $\mathcal{L}(A_i) = out$. Then, $\mathcal{L}(A_k) = in$, since the length of this cycle is odd. Therefore, $\mathcal{L}(C_1) = out$. Labeling of the other nodes proceeds as in the case when C_1 is an entrance. Thus, \mathcal{L} is stable (Figure 6(a)).

(2-2) If A_i is an entrance, then $\mathcal{L}(A_i) = out$. The length of a path from C_2 to C_1 in \mathcal{F} is odd, since C_1 is the terminal connector in \mathcal{F}' . Therefore, $\mathcal{L}(A_k) = in$. Then, $\mathcal{L}(C_1) = out$.

The length of any path from C_1 to C_2 in \mathcal{F} is even, since the length of any path from C_2 to C_1 in \mathcal{F} is odd. Therefore, for any cpath(h) from C_1 to C_2 in \mathcal{F} , $\mathcal{L}(B_h) = in$. Then, $\mathcal{L}(C_2) = out$.

Thus, \mathcal{L} is stable (Figure 6(b)).

(3) Let \mathcal{L} be a labeling of an AF obtained by adding an annihilator to the node that does not satisfy either condition shown in (1) or (2). We show that a contradiction arises in each case.

(3-1) Let $\langle C_1, A_1, \dots, A_k, C_1 \rangle$ be an arbitrary cpath(k) from C_1 to C_1 in \mathcal{F} .

If A_j is an entrance that is even-length-far from C_1 , then $\mathcal{L}(A_k) = \mathcal{L}(A_j) = out$.

For another cpath(h) from C_1 to C_1 without an annihilator, $\mathcal{L}(B_h) = out$, since the length of the path is odd. This means that there is no node labeled *in* that attacks C_1 in any path from C_1 to C_1 . Therefore, there should exist a path from C_2 to C_1 that includes a node labeled *in* that attacks C_1 .

Here, we consider the sub-AF \mathcal{F}_s obtained by removing all the paths from C_1 to C_1 from \mathcal{F} . Since \mathcal{F}_s is also a target AF, it is unstable. The terminal connector C_1 in \mathcal{F}' is not attacked by a node labeled *in* from the outside of \mathcal{F}_s . Therefore, \mathcal{F}_s is unstable. As a result, there is no node labeled *in* that attacks C_1 in any path.

Thus, \mathcal{L} is unstable (Figure 6(c)).

(3-2) When C_2 is a connector in \mathcal{F}' , let $\langle C_2, A_1, \dots, A_k, C_1 \rangle$ be an arbitrary cpath(k) from C_2 to C_1 in \mathcal{F} .

If A_j is an entrance that is even-length-far from C_1 , then $\mathcal{L}(A_k) = \mathcal{L}(A_j) = out$. Therefore, there should exist a path from C_1 to C_1 including a node labeled *in* that attacks C_1 ; which is impossible (Figure 6(d)).

(3-3) When C_2 is a connector in \mathcal{F}' , let $\langle C_1, A_1, \dots, A_k, C_2 \rangle$ be an arbitrary cpath(k) from C_1 to C_2 in \mathcal{F} .

If A_j is an entrance that is odd-length-far from C_1 in a cpath(k) from C_1 to C_2 in \mathcal{F} , then $\mathcal{L}(A_k) = \mathcal{L}(A_j) = out$, since C_2 is even-length-far from C_1 .

Assume that $\mathcal{L}(C_2) = in$. $\mathcal{L}(C_1) = out$, since C_1 is odd-length-far from C_2 . As C_2 is a connector, there exists either a path from C_2 to C_2 or another path from C_1 to C_2 . As the length of the path from C_2 to C_2 is odd, the node in this path that attacks C_2 is labeled *in*. As the length of another path from C_1 to C_2 is even, the node in this path that attacks C_2 is labeled *in*. Therefore, there exists a node labeled *in* that attacks C_2 in either case; which is a contradiction.

Assume that $\mathcal{L}(C_2) = out$. There should exist a node labeled *in* that attacks C_1 . As C_1 is a connector, there exists either a path from C_1 to C_1 , or another path from C_2 to C_1 . As the length of the path from C_1 to C_1 is odd, the node in this path that attacks C_1 is labeled *in*. As the length of another path from C_2 to C_1 is odd, the node in this path that attacks C_1 is labeled *out*. Therefore, there does not exist a node labeled *in* that attacks C_1 in either case; which is a contradiction (Figure 6(e)).

(3-4) When C_2 is a connector in \mathcal{F}' , let $\langle C_1, A_1, \dots, A_k, C_2 \rangle$ be an arbitrary cpath(k) from C_1 to C_2 in \mathcal{F} .

Assume that we add an annihilator to A_j which is even-length-far from C_1 in a cpath(k) from C_1 to C_2 in \mathcal{F} . This is a contradiction by the same discussion with that in (3-3) (Figure 6(f)).

(3-5) When C_2 is a connector in \mathcal{F}' , let $\langle C_1, A_1, \dots, A_k, C_2 \rangle$ be an arbitrary cpath(k) from C_1 to C_2 in \mathcal{F} .

Assume that we add an annihilator to A_j in a cpath(k) from C_2 to C_2 in \mathcal{F} . This is a contradiction by the same discussion with that in (3-3). \square