An Argumentation Model with Queries

Yu NAMBU[1] and Kazuko TAKAHASHI[1]

School of Science&Technology, Kwansei Gakuin University,
2-1, Gakuen, Sanda, 669-1337, JAPAN
ktaka@kwansei.ac.jp cdh67289@kwansei.ac.jp

Abstract. An argumentation is generally defined by a set of arguments and attacks between them. If a pair of arguments attacks each other, the result of an argumentation may not be decided. We propose an argumentation model that can resolve such mutual attacks and can reach an agreement between the participants. In this model, each argument is assumed to consist of three elements: claim, data, and warrant. The direction of attack between arguments is syntactically determined based on the refinement level of each argument. When we cannot determine the winner of an argumentation because of a mutual attack, a query is made to urge agents to present more refined arguments about the arguments in question. Agents can continue a deeper argumentation by obtaining new information as answers. Deeper argumentation enables an undecided argumentation to be decided. By incorporating refinement and query, the model can simulate more practical argumentation.
Keywords: argumentation, preference, query, refined argument, Toulmin model

1 Introduction

An argumentation is a kind of dialogue, but it differs from ordinary dialogue because a unit that constructs an argumentation is a statement with grounds, and each utterance is given with a kind of validity.

To date, most research in argumentation has been undertaken in the field of law, but recently, researchers in the fields of artificial intelligence and computer science have focused on its logical properties [13, 6]. Argumentation has many promising applications, such as decision support in a multi-agent environment, proofs for statements made in legal judgments, and support for verification of assurance cases.

Dung proposed an argumentation framework to allow abstract treatment of argumentation, and showed its relationship with logic programming and nonmonotonic reasoning [7]. Within this framework, when arguments and the relations of attacks between them are given, an argumentation can be represented as a graph, ignoring the contents of each argument. This approach has been adopted by many researchers in artificial intelligence and logic programming, who have proposed various argumentation models [13]. This technique is simple and easy to apply because most simulation and semantics problems can be reduced to those on graphs. In this framework, if a mutual attack exists, if we want to decide whether an argument is accepted in an argumentation, the result is different depending on which argument to be admitted that constitutes a mutual attack. However, in a practical argumentation, agents usually engage in challenges to resolve this mutual attack and reach the result. The phenomenon is particularly notable when the arguments that constitute a mutual attack make different claims based on the same data.

For example, assume that agent a and b want to decide whether they take umbrellas with them. Suppose that an agent a gives an argument A, "We should take umbrellas with us because the weather forecast says the chance of rain today is 30 percent." If b accepts this argument, they reach an agreement. However, if agent b gives the following counterargument B, "We need not take umbrellas with us because the weather forecast says the chance of rain today is 30 percent," then a must add the reason why he/she deduce the conclusion "taking umbrellas with us" from the probability of rain in the forecast. In practical argumentation, when a contradictory pair of claims arises from the same fact, each agent usually wants to know why his/her opponent is making the counterargument. They will ask "Why do you think so?" If agent a answers, for example, A', "In my experience, it has always rained on days when the weather forecast said that the chance of rain was 30 percent," as a subsequent argument, b may make a counterargument to this argument or he/she may give the grounds for his/her reasoning. In any case, the argumentation can proceed and may reach some result.

In this example, a mutual attack exists between A and B, but A' is not an attack on B. Rather, it is an answer to the "Why" question that gives a detailed explanation for his/her reasoning. However, an attack relation cannot capture this phenomenon and cannot represent the process of resolving a mutual attack.

In this paper, we propose an argumentation model with two binary relations over a set of arguments in addition to an attack so that we can manage the process of resolving mutual attacks. We define an argument not in an abstract way, but as an entity that has three components, and classify attacks depending on the target component of each argument. This definition of argument is based on the Toulmin model [14]. Some types of mutual attack can be resolved by posing a query. Agents can continue a deeper argumentation by obtaining new information as answers. More refined arguments can help resolve an undecided argumentation.

The goal of our model is twofold: to clarify the reasons for attacks and argumentation flow, and to make it possible to resolve an argumentation that is undecided due to mutual attacks between arguments. The first goal can be achieved by incorporating the contents of an argument; the second can be approached by encouraging agents to proceed to deeper argumentation by providing queries.

The rest of this paper is organized as follows. Section 2 introduces the concepts on which our work is based. Section 3 formalizes our argumentation model. Section 4 describes our argumentation procedure with a query phase and Section 5 presents an illustrative example. Section 6 compares our approach with other approaches. Finally, Section 7 presents our conclusions.

2 Basic Concepts

2.1 Toulmin Model

The Toulmin model is a well-known argumentation model [14], on which many studies have been based [8, 15].

In the model, an argument, which is a unit of an argumentation, consists of six components: *data, claim, warrant, backing, qualifier,* and *rebuttal.* Figure 1 shows one configuration of these components. The claim is an assertion, which is the core of an argument. Data comprise a set of facts, which act as the grounds for the claim, and the warrant is a rule that explains why the claim is deduced from the data. The backing is a universal or legal ground that supports the warrant. The qualifier shows the degree of force that the data give to the claim by the warrant. The rebuttal is a condition of exception for the argument. The relationships among these components can be interpreted as follows: the claim with the qualifier is deduced from data, based on the warrant and supported by the backing.

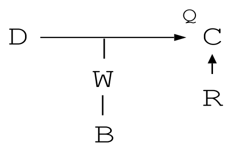

Fig. 1. Toulmin's argument diagram

Example 1 *The following is a sample argument [14].*
data (D): Harry was born in Bermuda.
qualifier (Q): presumably
warrant (W): A man born in Bermuda will generally be a British subject.
backing (B): The following status and other legal provisions.
rebuttal (R): Both his parents were aliens and he has become a naturalized American.
claim (C): Harry is a British subject.

In a practical argumentation, each argument is not always presented with these six components. An argument often appears without Q, B, and R; these parts are supplied afterwards when required.

2.2 Dung's Argumentation Framework

Whereas Toulmin focused on the use of argument to defend a claim by asserting something, Dung presented an argumentation framework that abstracted the contents of arguments away and focused mainly on the interaction between arguments and acceptance of arguments. This approach is currently the base of most argumentation models.

Definition 1 (Dung's argumentation framework) *[7]*
An argumentation framework *is defined as a pair $\langle Args, Atts \rangle$ where $Args$ is a set of arguments and $Atts$ is a binary relation over $Args$, representing attacks.*

3 Formalization

We formalize our argumentation model.

Definition 2 (consistent) *Let Ψ be a set of formulas in propositional logic. If no ψ exists that satisfies both $\psi \in \Psi$ and $\neg\psi \in \Psi$, Ψ is said to be* consistent*.*

The knowledge base $\mathbf{K}_{\mathrm{a}}$ for each agent a is a finite, consistent set of propositional formulas. An agent a participates in argumentation using elements of $\mathbf{K}_{\mathrm{a}}$. Note that $\mathbf{K}_{\mathrm{a}}$ may not be deductively closed, that is, there may be a case in which $\phi, \phi \Rightarrow \psi \in \mathbf{K}_{\mathrm{a}}$ and $\psi \notin \mathbf{K}_{\mathrm{a}}$ hold. Also note that $\neg\neg\psi$ is considered to be ψ.

An argument is defined based on the Toulmin model.

Definition 3 (argument) *Let $\mathbf{K}_{\mathrm{a}}$ be a knowledge base for an agent a.* An argument of a *is a triple $(\phi, \phi \Rightarrow \psi, \psi)$, where ϕ, ψ are formulas in $\mathbf{K}_{\mathrm{a}}$, or $(\bot, \bot, \psi)$, where ψ is a formula in $\mathbf{K}_{\mathrm{a}}$ and $\bot$ denotes that no element exists. For an argument $A = (D, W, C)$, D, W, and C are said to be* the data*,* the warrant*, and* the claim*, and they are denoted by $Clm(A), Dat(A)$, and $Wrr(A)$, respectively.*

Definition 4 (support) *For an argument $A = (D, W, C)$, if $D \neq \bot$, then we say that A* has support*, otherwise, we say that A* has no support*.*

Each argument has a level of statement. For example, "the weather forecast says the probability of rain is 30%, and in my experience, it has always rained on days when the weather forecast said that the chance of rain was 30%" is a stronger reason for the same claim "taking an umbrella" than just the fact "the weather forecast says the probability of rain is 30%." We define the level of refinement based on the number of propositions included in the data of an argument. The level of refinement of an argument without support is defined as a maximum value, `max`.

Definition 5 (level of refinement) *For an argument A, the level of refinement rLevel(A) is defined as the number of propositions in $Dat(A)$, if $Dat(A) \neq \bot$, and* `max` *if $Dat(A) = \bot$. For arguments A_a and A_b, if rLevel(A_a) $>$ rLevel(A_b), then it is said that A_a* is more refined than *A_b.*

Attack is a binary relation between arguments and it is available from an argument at a higher level to an argument at a lower or equal level.

Definition 6 (attack) *Let A_a and A_b be arguments. An attack from A_a to A_b is defined as follows.*

1. *If $Clm(A_a) \Rightarrow \neg Clm(A_b)$, $rLevel(A_a) \geq rLevel(A_b)$ and A_a has support, then (A_a, A_b) is said to be* a rebut *from A_a to A_b.*
2. *If $Clm(A_a) \Rightarrow \neg Dat(A_b)$ or $Clm(A_a) \Rightarrow \neg Wrr(A_b)$, $rLevel(A_a) \geq rLevel(A_b)$ and A_a has support, then (A_a, A_b) is said to be* an undercut *from A_a to A_b.*
3. *If $Clm(A_a) \Rightarrow \neg Dat(A_b)$ or $Clm(A_a) \Rightarrow \neg Wrr(A_b)$, $rLevel(A_a) \geq rLevel(A_b)$ and A_a has no support, then (A_a, A_b) is said to be* a force *from A_a to A_b.*
4. An attack *from A_a to A_b is either a rebut, an undercut, or a force from A_a to A_b.*

When (A_a, A_b) is an attack from A_a to A_b, then it is said that A_a attacks *A_b.*

A rebut is an attack on a claim, whereas an undercut and force are attacks on the data or the warrant. An attack from an argument without support on the claim of another argument is not allowed because mutual attacks between arguments without support generate meaningless repetition.

A force is an attack from an argument without support on the data or the warrant of another argument. Figure 2 illustrates the effect of a force. Let A_1, A_2, and A_3 be arguments, and (A_2, A_1) and (A_3, A_2) be attacks. If A_2 is a force to A_1, A_2 has no support. Therefore, an attack to A_2 should be a rebut. Because $Clm(A_2)$ is equivalent to $\neg Dat(A_1)$, $Clm(A_3)$ is equivalent to $Dat(A_1)$. This process means that the force is an attack that requires the agent to present the grounds of the data.

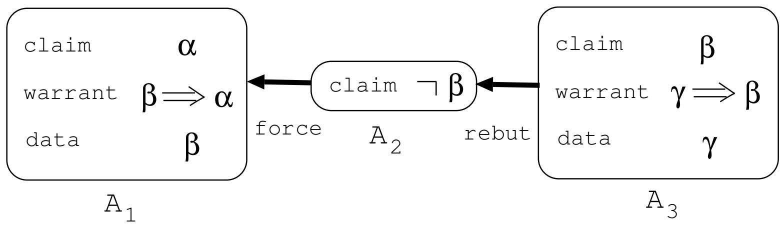

Fig. 2. An effect of a force

In addition to the attack, we define two more binary relations on a set of arguments: complementary pair of arguments and backing. A complementary pair of arguments is a pair of arguments with the same data and complementary claims, while backing means that one's claim is equivalent to the other's warrant. These two relationships are essential to represent a practical argumentation.

Definition 7 (complementary pair of arguments) *For a pair of arguments A_a and A_b, if $Dat(A_a) \equiv Dat(A_b)$ and $Clm(A_a) \equiv \neg Clm(A_b)$, then (A_a, A_b) is a* complementary pair of arguments.

Definition 8 (backing) *For a pair of arguments* A_a *and* A_b*, if* $Clm(A_a) \equiv Wrr(A_b)$*, then* (A_a, A_b) *is a* backing *from* A_a *to* A_b*.*

4 Argumentation Procedure

In this section, we present our argumentation procedure between agents P and C. We assume that P and C have their own knowledge bases from which all possible arguments are constructed.

4.1 Argumentation Framework

Definition 9 (argumentation framework) *Let* $\mathbf{K}_{\mathrm{P}}$ *and* $\mathbf{K}_{\mathrm{C}}$ *be knowledge bases for agents P and C, respectively.* An argumentation framework *between P and C,* $AF(\mathbf{K}_{\mathrm{P}}, \mathbf{K}_{\mathrm{C}})$ *is defined as a five-tuple* $\langle Arg_P, Arg_C, Atts, CPs, BKs\rangle$*, where* Arg_P *and* Arg_C *are sets of all possible arguments of* P *and* C*, respectively, and* $Atts$*,* CPs*, and* BKs *are the binary relations on* $Arg_P \cup Arg_C$*, which represent attacks complementary pairs of arguments and backings, respectively.*

Let AF be an argumentation framework. Then, from the consistency of knowledge bases, the following propositions hold.

Proposition 1 *(1) For each element* $(A_a, A_b) \in Atts$*, if* $A_a \in Arg_P$*, then* $A_b \in Arg_C$*; and if* $A_a \in Arg_C$*, then* $A_b \in Arg_P$*.*
(2) $CPs \subseteq Atts$*.*
(3) For each element $(A_a, A_b) \in BKs$*, if* $A_a \in Arg_P$*, then* $A_b \in Arg_P$*; and if* $A_a \in Arg_C$*, then* $A_b \in Arg_C$*.*
(4) There is no cycle of attacks that consists of an odd number of arguments.

Example 2 *Let P and C be agents, and let* $\mathbf{K}_{\mathrm{P}} = \{\alpha, \beta, \gamma, \delta, \beta \Rightarrow \alpha, \gamma \Rightarrow \beta, \delta \Rightarrow (\gamma \Rightarrow \beta)\}$ *and* $\mathbf{K}_{\mathrm{C}} = \{\neg\beta, \gamma, \gamma \Rightarrow \neg\beta\}$ *where* $\alpha, \beta, \gamma,$ *and* δ *are propositions. Then, three arguments* $P_1 = (\beta, \beta \Rightarrow \alpha, \alpha)$*,* $P_2 = (\gamma, \gamma \Rightarrow \beta, \beta)$ *and* $P_3 = (\delta, \delta \Rightarrow (\gamma \Rightarrow \beta))$ *are generated from* $\mathbf{K}_{\mathrm{P}}$*, and one argument* $C_1 = (\gamma, \gamma \Rightarrow \neg\beta, \neg\beta)$ *is generated from* $\mathbf{K}_{\mathrm{C}}$*.* $rLevel(P_1) = rLevel(P_2) = rLevel(P_3) = rLevel(C_1) = 1$*. In this case,* $AF_0 = \langle Arg_P, Arg_C, Atts, CPs, BKs\rangle$ *is defined as follows:* $Arg_P = \{P_1, P_2, P_3\}, Arg_C = \{C_1\}$*,* $Atts = \{(C_1, P_1), (P_2, C_1), (C_1, P_2)\}$*,* $CPs = \{(C_1, P_2)\}$*,* $BKs = \{(P_3, P_2)\}$ *as shown in Figure 3.*

Definition 10 (related complementary pair)
Let $AF = \langle Arg_P, Arg_C, Atts, CPs, BKs\rangle$ *be an argumentation framework. For an argument* A *and complementary pair of arguments* $(A', A'') \in CPs$*, if there exists a sequence of arguments* $A_0, \ldots, A_n$*, where* $A_0 = A$*,* $A_n = A'$*, and* $(A_i, A_{i-1}) \in Atts$ *for each* i $(1 \le i \le n)$*, then this complementary pair of arguments is* related to A*. Additionally, a set of complementary pairs of arguments that are related to* A *is denoted by* $CPs(A)$*.*

Example 3 *(Cont'd) For the argumentation framework* AF_0*,* $CPs(P_1) = \{(C_1, P_2)\}$ *and* $CPs(P_3) = \emptyset$*.*

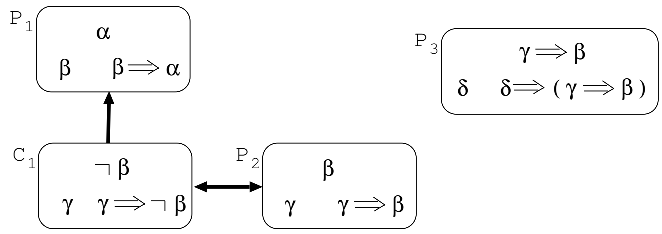

Fig. 3. Example of an argumentation framework AF_0

4.2 Extensions

Here, we introduce the notion of using an *extension* defined by Dung [7], to determine win or loss of an argumentation. An extension is an acceptable set of arguments within a given argumentation framework.

Definition 11 (conflict-free, admissible) *Let* $\langle Arg_P, Arg_C, Atts, CPs, BKs\rangle$ *be an argumentation framework. For* $A, B \in Arg_P \cup Arg_C$*, and* $\mathcal{S} \subseteq Arg_P \cup Arg_C$*,*
(1) $\mathcal{S}$ *is* conflict-free *iff there are no elements* $A, B \in \mathcal{S}$ *such that* A *attacks* B*.*
(2) $\mathcal{S}$ defends A *iff* $\mathcal{S}$ *attacks each argument that attacks* A*.*
(3) $\mathcal{S}$ *is* admissible *iff* $\mathcal{S}$ *is conflict-free and defends all of the elements.*

There are several definitions of acceptability, and different extensions are possible for each acceptability. Here, we adopt a preferred extension.

Definition 12 (preferred extension) *A maximal admissible set with respect to* $\subseteq$ *is* a preferred extension*.*

Example 4 *(Cont'd) For the argumentation framework* AF_0*, its preferred extensions are* $\{P_1, P_2, P_3\}$ *and* $\{C_1, P_3\}$*.*

4.3 Argumentation Procedure with A Query Phase

In an argumentation procedure, we want to decide whether an agent proposing some issue wins or loses an argumentation.

Definition 13 (acceptance) *For an argumentation framework* $AF = \langle Arg_P, Arg_C, Atts, CPs, BKs\rangle$ *and an argument* $A \in Arg_P \cup Arg_C$*, if* $CPs(A) = \emptyset$ *and all preferred extensions include* A*, then* A is accepted in AF*; if* $CPs(A) = \emptyset$ *and if there exists a preferred extension that does not include* A*, then* A is non-accepted in AF*; and if* $CPs(A) \neq \emptyset$*,* A *is* undecided in AF*. The acceptance of* A *in* AF *is denoted by* $judgeAcc(A, AF)$*.*

Example 5 *(Cont'd) For the argumentation framework* AF_0*,* $judgeAcc(P_1, AF) = undecided$*,* $judgeAcc(C_1, AF) = undecided$ *and* $judgeAcc(P_3, AF) = accepted$*.*

For an argumentation framework $AF = \langle Arg_P, Arg_C, Atts, CPs, BKs \rangle$ and an argument $A \in Arg_P \cup Arg_C$, an argumentation procedure with a query phase proceeds as follows.

First, we check whether A is accepted. When it is *undecided*, then a complementary pair of arguments exists in AF. We assume that different reasons are used in the processes through which a complementary pair of claims is deduced from the same data. In this case, we want to urge each agent to present the reason why his/her claim is deduced, that is, the grounds of his/her warrant. This can be done by invoking an additional argumentation in the query phase. The additional argumentation starts with the argument of which the claim is such a warrant.

The purpose of a query phase is to eliminate mutual attacks in the main argumentation. We feed the results back to the main argumentation so that it proceeds with the newly obtained information. If an agent can give a sufficient explanation against which the opponent cannot give any more counterargument, this means that the agent's warrant has sufficiently strong grounds, that is the backing. In this case, no more detailed explanation is expected and that warrant remains. Otherwise, the grounds are insufficient to make the warrant, so we withdraw that warrant from the knowledge base. It follows that a mutual attack is eliminated.

From this process, we can change an undecided argumentation into a decided one.

We present the procedure for argumentation with a query phase.

Argumentation procedure with a query phase

Let $AF = \langle Arg_P, Arg_C, Atts, CPs, BKs \rangle$ be an argumentation framework, and A_{is} be an issue argument. Let *player* be an agent who gives A_{is}. The following procedure $judgeArg(A_{is}, AF)$ determines if *player* wins or loses AF.

1. If $judgeAcc(A_{is}, AF) = accepted$,
   then terminate with the result $judgeArg(A_{is}, AF) = win$,
   which means *player* wins AF.
   If $judgeAcc(A_{is}, AF) = non\text{-}accepted$,
   then terminate with the result $judgeArg(A_{is}, AF) = lose$,
   which means *player* loses AF.
   If $judgeAcc(A_{is}, AF) = undecided$,
   then for $(A_P, A_c) \in CPs$,
   do the *existBacking* for A_P and A_C, respectively.
2. If $existBacking(A_P, AF) = true$ and $existBacking(A_C, AF) = true$,
   then set $AF' = \langle Arg_P, Arg_C, Atts, CPs - \{(A_P, A_C)\}, BKs \rangle$,
   and go to 6.
3. If $existBacking(A_P, AF) = false$,
   then set $\mathbf{K}_{\mathrm{P}}{}'$ to be $\mathbf{K}_{\mathrm{P}} - \{Wrr(A_P)\}$.
4. If $existBacking(A_C, AF) = false$,
   then set $\mathbf{K}_{\mathrm{C}}{}'$ to be $\mathbf{K}_{\mathrm{C}} - \{Wrr(A_C)\}$.
5. Set AF' to be $AF(\mathbf{K}_{\mathrm{P}}{}', \mathbf{K}_{\mathrm{C}}{}') = \langle Arg'_P, Arg'_C, Atts', CPs', BKs' \rangle$.

6. Set $AF = AF'$ and go to 1.

$existBacking$ is invoked as a query phase and defined as follows.
$existBacking(A, AF)$ is $true$ if there exists an argument A' such that $(A', A) \in BKs$ and $judgeArg(A', AF) = win$, otherwise it is $false$.

Note that this argumentation procedure terminates because the number of the arguments is finite and the number of CPs decreases every cycle.

Proposition 2
For an argumentation framework $AF = \langle Arg_P, Arg_C, Atts, CPs, BKs\rangle$ and an argument A, $judgeArg(A_{is}, AF)$ terminates with the result that the player that proposes A wins or loses AF.

Example 6 *(Cont'd) The argumentation procedure with a query phase for AF_0 and the issue node P_1 is shown below. First, $judgeAcc(P_1, AF) = undecided$, and $(C_1, P_2) \in CPs(P_1)$. Then, go to the query phase. As $existBacking(C_1, AF)$ $= false$ and $existsBacking(P_2, AF) = true$, delete $Wrr(C_1)$ from $\mathbf{K}_\mathrm{C}$ and reconstruct the argumentation framework to get $AF' = \langle\{P_1, P_3\}, \emptyset, \emptyset, \emptyset, \{(P_1, P_3)\}\rangle$. Next, $judgeAcc(P_1, AF') = accepted$, because there is no complementary pair of arguments and its preferred extension is $\{P_1, P_3\}$. Therefore, P wins AF'.*

5 Illustrative Dialogue

The following section illustrates how our argumentation procedure with a query phase works using a case study of the history of chemistry.

Example 7 *Some scientists in the 17th century believed that an entity called "phlogiston" was the cause of combustion. According to this theory, phlogiston was a substance contained in combustible bodies and released during combustion. The existence of phlogiston was disproven when Lavoisier showed that combustion requires a gas that has weight (i.e., oxygen) and could be measured.*

The following illustrates the argumentation between scientists in the phlogiston school (P) and those in the non-phlogiston school (C). This is a typical example in which a complementary pair of claims is deduced from the same data in different knowledge bases, and where the mutual attack is ultimately resolved by a query phase.

P_1 *"Phlogiston exists because it is contained in every combustible resource."*
C_1 *"I don't think that it is contained in every combustible resource."*
P_2 *"It is contained because it is released during combustion."*
C_2 *"I don't think that it is released during combustion."*
P_3 *"The fact that a material can burn means that phlogiston is released."*
C_3 *"When a material burns, it gets weight. Thus, it is unlikely that anything is released. So, phlogiston is not released."*
P_4 *"No. The fact that "a material gets weight on burning" itself proves that phlogiston is released."*

In C_3 and P_4, a complementary pair of claims "phlogiston is released" and "phlogiston is NOT released" from the same data "a material gets weight when burning." Then, queries are generated to ask both agents to justify their arguments.

The query from C to P is as follows.

$Q_{C\to P}$ *"Why do you think that the phlogiston is released, given that a material gets weight when burning?"*

P_5 *"Due to the fact that phlogiston has a negative weight."*

C_5 *"There is no material that has a negative weight. Therefore, phlogiston does not have a negative weight."*

In this case, P_5 is an answer to the question, and C makes a counterargument to P's answer. Then, there are no more arguments. Therefore, the warrant for P_4 is not justified.

On the other hand, the query from P to C is as follows.

$Q_{P\to C}$ *"Why do you think that phlogiston is NOT released, given that a material gets weight when burning?"*

C_4 *"Because the fact that a material burns means that it gets oxygen."*

In this case, C_4 answers the question and P makes no additional counterargument[1]*. Therefore, the warrant for C_3 is justified.*

Overall, C is the winner of this argumentation.

Here is the process of argumentation with a query phase.

Let $\mathbf{K}_\mathrm{P}$ and $\mathbf{K}_\mathrm{C}$ be knowledge bases of P and C, respectively, as shown in Figure 4.

First, we make an initial argumentation framework $AF = \langle Arg_P, Arg_C, Atts, CPs, BKs\rangle$ from $\mathbf{K}_\mathrm{P}$ and $\mathbf{K}_\mathrm{C}$ (Figure 5). Note that (C_1, P_1) and (C_2, P_2) are the forces. (C_3, P_4) is the only complementary pair of arguments. (P_5, P_4) and (C_4, C_3) are backings. In this figure, for each node that corresponds to an argument A, $Clm(A)$ appears in the upper column of the node, $Dat(A)$ appears in the lower left of the node, and $Wrr(A)$ appears in the lower right of the node. For example, $Clm(C_3)$, $Dat(C_3)$, and $Wrr(C_3)$ are $\neg relP$, $burn \Rightarrow getWeigt$ and $(burn \Rightarrow getWeight) \Rightarrow \neg relP$, respectively. Note that C_2 cannot attack P_3, since C_2 which has no support is not allowed to attack P_3's claim, and that P_3 cannot attack C_3, since C3 is more refined than P_3.

We start argumentation on AF and the issue argument P_1. $judgeAcc(P_1, AF) = undecided$, because $(C_3, P_4) \in CPs(P_1)$.

Then, a query phase starts for (C_3, P_4), which invokes additional argumentation (Figure 6).

First, we explain a query phase for P_4.

[1] Historically, this answer took place several years after the question was presented.

$phlogiston$	$includeP \Rightarrow phlogiston$
$includeP$	$relP \Rightarrow includeP$
$relP$	$burn \Rightarrow relP$
$burn$	$burn \Rightarrow getWeight$
$minus$	$(burn \Rightarrow getWeight) \Rightarrow relP$
$minus \Rightarrow ((burn \Rightarrow getWeight) \Rightarrow relP)$	

$\mathbf{K}_{\mathrm{P}}$

$\neg includeP$	$burn \Rightarrow getWeight$
$\neg relP$	$(burn \Rightarrow getWeight) \Rightarrow \neg relP$
$\neg minus$	$\neg observeMinus \Rightarrow \neg minus$
$getOxy$	
$\neg observeMinus$	
$getOxy \Rightarrow ((burn \Rightarrow getWeight) \Rightarrow \neg relP)$	

$\mathbf{K}_{\mathrm{C}}$

Fig. 4. Agents' knowledge bases

Because P_5 is the backing for P_4, a query is a bridge between P_4 and P_5. In this figure, $[\mathcal{Q}]((burn \Rightarrow getWeight) \Rightarrow relP)$ denotes the question corresponding to $Q_{C \rightarrow P}$, and $[\mathcal{A}]minus \Rightarrow ((burn \Rightarrow getWeight) \Rightarrow relP)$ denotes the answer. We examine whether P_5 is accepted in AF. Because $CPs(P_5) = \emptyset$, and P_5 is not included by any preferred extension of AF, $judgeAcc(P_5, AF) =$ *non-accepted*. Therefore, $existBacking(P_4, AF) = false$.

The query phase for C_3 is performed using a similar procedure. Because C_4 is the backing for C_3, a query is a bridge between C_4 and C_3. Because $CPs(C_4) = \emptyset$ and C_4 is included by all preferred extensions of AF, $judgeAcc(C_4, AF) =$ *accepted*, Therefore, $existBacking(C_4, AF) = true$.

Next, we update the knowledge bases and reconstruct an argumentation framework. Because $existBacking(P_4) = false$, formula $(burn \Rightarrow getWeight) \Rightarrow relP$ is eliminated from $\mathbf{K}_{\mathrm{P}}$. The resulting corresponding graph for the reconstructed argumentation framework is shown in Figure 7. It has only one *preferred extension* $\{C_1, C_2, C_3, C_4, C_5\}$ and a set of complementary pairs of arguments $CPs' = \emptyset$. Therefore, $judgeAcc(P_1, AF') = lose$, meaning that P loses the main argumentation.

6 Discussion

6.1 Relationship with the Toulmin Model

In our model, the argument structure is based on the Toulmin model. Initially, an argumentation proceeds without backing in our model. However, when the judgment for an argumentation is not decided because of a mutual attack, a more detailed explanation is required. Our model requires both agents to present

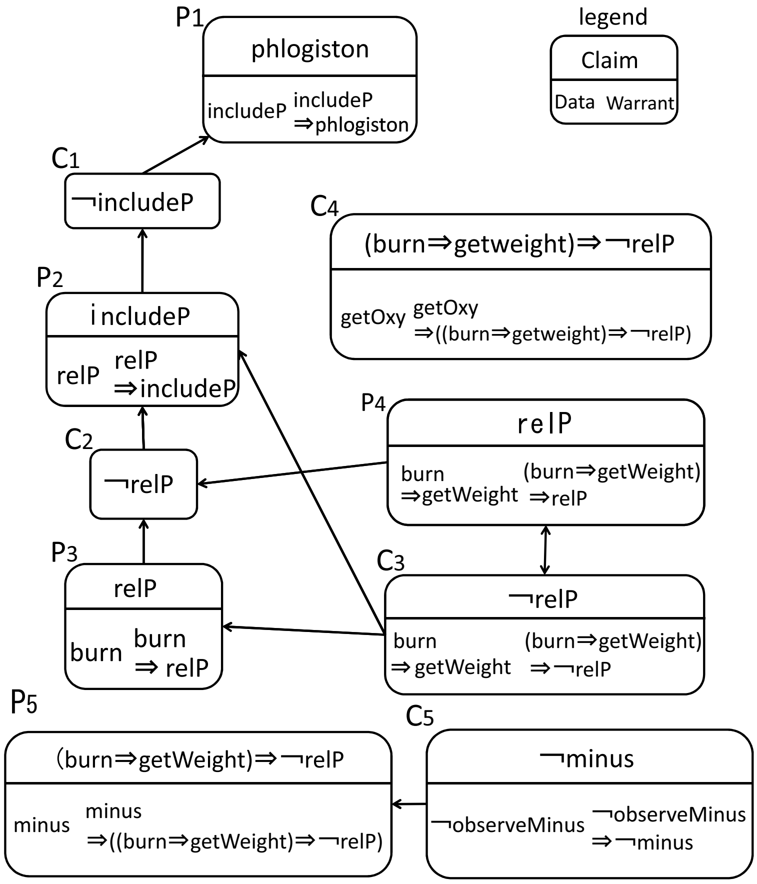

Fig. 5. Argumentation for the phlogiston example

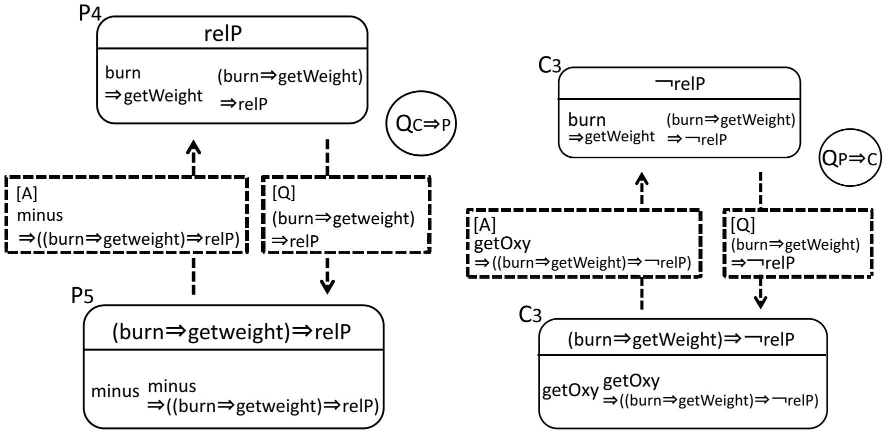

Fig. 6. Query phase

reasons for their own statements by giving a query. A query can be considered a trigger to encourage an agent to give backing to the warrant of an argument in a mutual-attack relation. This invokes an additional argumentation. Winning an additional argumentation means that the backing is given. An argument with backing is stronger than one without. This is reflected in the judgment on the reconstructed argumentation.

Another important point of comparison is the treatment of rebuttal. In the Toulmin model, attack is possible as a rebuttal. However, it is not clear which element of an argument a rebuttal is attacking. In contrast, in our model, the target element of attack is obvious.

6.2 Preference-Based Argumentation

According to Dung's abstract argumentation framework, if an argumentation framework includes a mutual attack between arguments, then an agent's win or loss is usually undecided. This makes the framework unsuitable for applications of persuasion, agreement, or legal judgment. Several extensions of this framework have been proposed to avoid mutual attacks; most have proposed a certain order over arguments such as preference [11, 1, 3] or value [4]. In these extended frameworks, preferences or values are assigned depending on the strength, certainty, or stability of formulas or the reliability of agents, so that attack is effective in a unique direction. However, this approach is unnatural because preference or value is set by a user arbitrarily. In our model, although the direction of attack relation is also determined uniquely based on the level of statement, it can be defined syntactically without user intervention. Additionally, almost all arguments have the same level of statement, and mutual attacks are not strictly prohibited. This means that mutual attacks may appear in the initial argumentation. Our

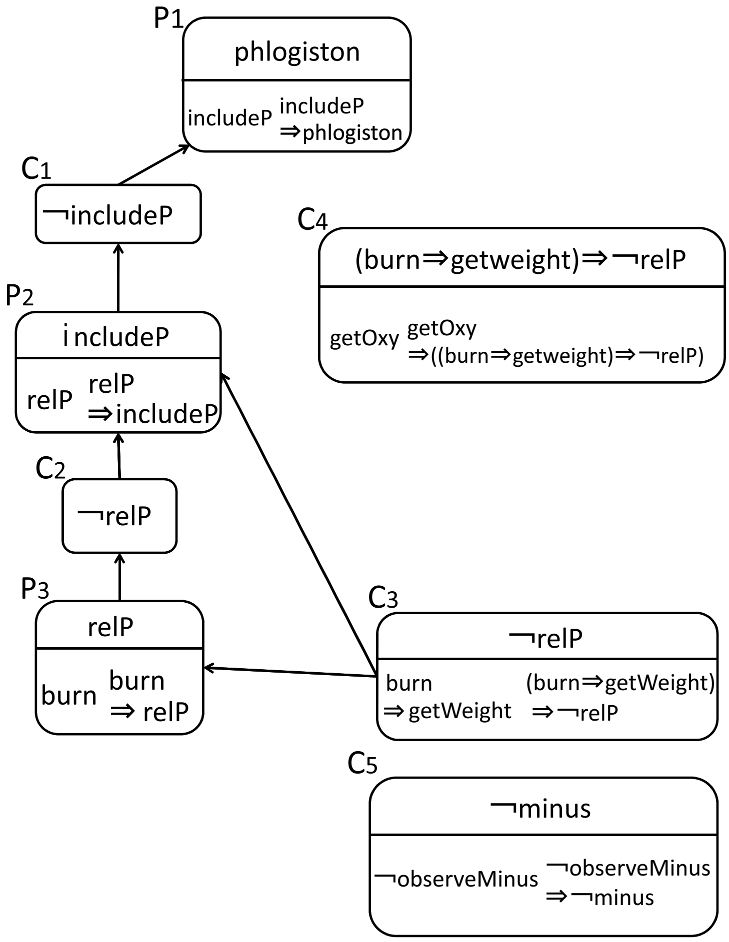

Fig. 7. Reconstructed argumentation

goal is to eliminate them based on how the argumentation proceeds by changing the knowledge bases of agents.

6.3 Extended Argumentation Framework

Modgil extended Dung's framework by introducing the meta-attack, an attack on an attack, and proposed the Extended Argumentation Framework (EAF) [9, 10]. In this framework, the relative strength of attacks is determined by the existence of meta-attacks without applying any preference. Because a meta-attack is an attack on the reason that the target attack is possible, it corresponds to an attack on the warrant in our framework. EAF differs in several ways from our framework. First, EAF ignores the contents of arguments, and it is not clear which element of an argument is being attacked. In contrast, in our framework, an argument is composed of three elements and the target element of an attack is clear. Second, in EAF, the strength of attacks in the loop can be determined by the existence of meta-attacks. In contrast, in our model, win or loss of an argumentation is decided by eliminating the loop.

6.4 Dynamic Argumentation Framework

In our model, the knowledge bases of agents change as the argumentation proceeds. Most argumentation frameworks do not take the change of knowledge bases into account, but several researchers have considered this point. Amgoud considered the knowledge base of each agent separately, as well as its revision during the exchange of arguments, for handling negotiation processes [2]. Cayrol discussed the revision of argumentation theory and investigated how acceptable arguments change when an argument is added at an abstract level [5]. Okuno proposed a dynamic argumentation framework in which agents' knowledge bases are changed during the execution of an argumentation [12]. His model explained the process of the generation of a new argument considering the contents of an argument. However, he also used preference to avoid mutual attacks, and did not discuss their elimination during an argumentation.

7 Conclusions

In this paper, we proposed a new argumentation model that includes queries to handle a practical argumentation.

In this model, an argument consists of three elements: the data, the claim, and the warrant that connects the data and the claim. An attack is classified depending on the target element. We developed an argumentation procedure based on this model. When we cannot determine the winner of an argumentation because of the existence of a pair of arguments that attack each other, a query is made to encourage agents to present detailed reasons for the warrants of the arguments in question. This model can explain why arguments occur and can also determine the winner of an argumentation when it includes mutual attacks. It is particularly useful for handling argumentations that include a pair of arguments with the same data and complementary claims.

In future research, we plan to extend the procedure to enable it to handle complementary pairs of arguments with different data and a complementary pair of claims. We are currently working to implement the system.

References

1. L.Amgoud and C.Cayrol: A reasoning model based on the production of acceptable arguments, Annals of Mathematics and Artificial Intelligence, 34(1-3), pp. 197-215, 2002.
2. L.Amgoud, Y.Dimopolos and P.Moraitis: A general framework for argumentation-based negotiation, ArgMAS2007, pp. 1-17, 2007.
3. L.Amgoud and S.Vesic: Repairing preference-based argumentation frameworks, IJCAI2009, pp. 665-670, 2009.
4. T.Bench-Capon: Persuasion in practice argument using value-based argumentation frameworks. Journal of Logic and Computation, 13(3), pp. 429-448, 2003.
5. C.Cayrol, F.de St-Cyr and M-C Lagasquie-Shiex: Change in abstract argumentation frameworks: adding an argument. Journal of Artificial Intelligence Research, 38, pp. 49-84, 2010.
6. C.Chesñevar, A.Maguitman and R.Loui: Logical models of argument. ACM Computing Surveys, 32(4), pp. 337-383, 2005.
7. P.M.Dung: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence, 77, pp. 321-357, 1995.
8. D.Hitchcock and B.Verheiji (eds.): *Arguing on the Toulmin Model*, Springer, 2006.
9. S.Modgil and T.Bench-Capon: Integrating object meta-level value based argumentation. In *Proc. of 2nd Int. Conf. on Computational Models of Argument*, pp. 240-251, 2008.
10. S.Modgil: Reasoning about preferences in argumentation frameworks, Artificial Intelligence, 173(9-10), pp. 901-1040, 2009.
11. H.Prakken and G.Sartor: Argument-based extended logic programming with defeasible priorities. Journal of Applied Non-Classical Logics, 7, pp. 25-75, 1993.
12. K.Okuno and K.Takahashi: Argumentation system with changes of an agent's knowledge base, IJCAI2009, pp. 226-232, 2009.
13. I.Rahwan and G.Simari (eds.): *Argumentation in Artificial Intelligence*, Springer, 2009.
14. S.Toulmin: *The Use of Argument*, Cambridge University Press, 1958.
15. B.Verheiji: The Toulmin Argument Model in Artificial Intelligence. In [13], 2009.