Closure and Consistency Rationalities in Logic-Based Argumentation

Phan Minh Dung Phan Minh Thang

Department of Computer Science, Asian Institute of Technology
GPO Box 4, Klong Luang, Pathumthani 12120, Thailand
dung,thangphm@cs.ait.ac.th

Abstract. Caminada and Amgoud have argued that logic-based argumentation systems should satisfy the intuitive and natural principles of logical closure and consistency. Prakken has developed this idea further for a richer logic. A question arises naturally whether a general structure guaranteeing the logical closure and consistency properties could be identified that is common for all underlying logics. We explore this question by first defining a logic-based argumentation framework as combination of an abstract argumentation framework with a monotonic Tarski-like consequence operator. We then demonstrate that the logical closure and consistency properties are rested on a simple notion of a base of arguments from which the argument could be constructed in an indefeasible way (using the monotonic consequence operator) and the only way to attack an argument is to attack its base. We show that two natural properties of structural closure and consistency covering based on the idea of bases of arguments indeed guarantee the logical closure and consistency properties. We demonstrate how the properties of structural closure and consistency covering are captured naturally in argumentation systems of Caminada, Amgoud and Prakken as well as in assumption-based argumentation.

1 Introduction

How do we know whether an argumentation framework is appropriate for its purposes ? Caminada and Amgoud [3] have argued that for logic-based systems, they should at least satisfy two intuitive and natural principles of logical closure and consistency. Prakken [8] has developed this idea further for a richer logic. But as there are many logics, Caminada, Amgoud and Prakken's results do not cover all of them. As argumentation is charaterized by arguments and the attack relation between them, a natural question is whether the logical closure and consistency principles could be captured in abstract argumentation without associating to a specific logic?

Logic-based abstract argumentation is viewed as abstract argumentation equipped with a general Tarski-like (monotonic) consequence operator. We develop in this paper two general principles of structural closure and consistency covering in logic-based abstract argumentation and show that they indeed capture the intuitions of the logical closure and consistency principles. The principles of structural closure and consistency covering rest on a simple notion of a base of arguments from which the argument could be constructed in an indefeasible way (using the monotonic consequence operator) and the only way to attack an argument is to attack its base. In other words the principle of logical closure boils down to the idea that if an argument is considered to be a "indefeasible logical consequence" of a set of arguments then the former must be acceptable wrt the later. The principle of consistency covering reduces the logical consistency to a kind of a conflict-freeness.

2 Logic-Based Abstract Argumentation Theories

Argumentation is a form of reasoning, that could be viewed as a dispute resolution, in which the participants present their arguments to establish, defend, or attack certain propositions. An abstract argumentation framework [5] is defined simply by a pair $(\mathcal{AR}\ , att)$ where $\mathcal{AR}$ is a set of arguments and att is a binary relation over $\mathcal{AR}$ representing the relation that an argument A *attacks* an argument B for $(A, B) \in att$. The semantics of abstract argumentation is determined by the acceptability of arguments and various associated notions of extensions. For the purpose of this paper, we introduce two of them. A set of argument S attacks an argument A if some argument in S attacks A; S is *conflict-free* if it does not attack itself. An agument A is *acceptable* wrt set of arguments S if S attacks each attack against A. S is *admissible* if S is conflict-free and it counter-attacks each attack against it. The *characteristic function* $\mathcal{F}$ assigns to each set of arguments S the set of arguments that are acceptable wrt S. As $\mathcal{F}$ is monotonic, $\mathcal{F}$ has a least fixed point. A *complete extension* is defined as a fixed point of $\mathcal{F}$ while the *grounded extension* is the least fixed point of $\mathcal{F}$. A *stable extension* is defined as a conflict-free set of arguments that attacks every argument not belonging to it. It is well-known that each stable extension is a complete extension but not vice versa. Stable extensions generalize the stable and answer set semantics of [6, 7].

Intuitively, an argument is a proof of some conclusion. In many cases, such proofs are constructed following some proof theory of some formal logics. Such logics could be nonmonotonic. The notion of closure is then defined accordingly as the set of consequences following from the monotonic parts of the underlying logics. For illustration, we use an example borrowed from [3].

Example 1. The logics in consideration consists of a set of strict rules $\mathcal{R}_0 = \{\rightarrow wr;\ \rightarrow go;\ b \rightarrow \neg hw;\ m \rightarrow hw\}$ and a set of defeasible rules $\mathcal{D} = \{wr \Rightarrow m;\ go \Rightarrow b\}$[1]. The monotonic logic is defined by the set of strict rules $\mathcal{R}_0$. There are 6 arguments[2]:

$A_1 : \rightarrow wr,\ \ A_3 : \rightarrow wr \Rightarrow m,\ \ A_5 : \rightarrow wr \Rightarrow m \rightarrow hw.$

[1] wr = "John wears something that looks like a a wedding ring", m = "John is married", hw = "John has a wife", go = "John often goes out until late", b = "John is a bachelor".

[2] For a precise definition see defintion 7

$A_2 : \to go, \quad A_4 : \to go \Rightarrow b, \quad A_6 : \to go \Rightarrow b \to \neg hw.$

Arguments A_3, A_5, A_4, A_6 are also often written as $A_1 \Rightarrow m$, $A_3 \Rightarrow hw$, $A_2 \Rightarrow b$ and $A_4 \Rightarrow \neg hw$ respectively.

Attack relation: A_5 attacks A_6 and vice versa. There are no other attacks. Let $att_0 = \{(A_5, A_6)\}$. The grounded extension contains arguments A_1, A_2, A_3, A_4. Hence the conclusions of the arguments in the grounded extension are not consistent wrt (monotonic) logic defined by the set of strict rules $\mathcal{R}_0$.

There are two preferred extensions $\{A_1, A_2, A_3, A_4, A_5\}$ and $\{A_1, A_2, A_3, A_4, A_6\}$. The conclusions of the arguments of neither is closed wrt (monotonic) logic defined by $\mathcal{R}$.

In this paper, we are interested in argumentation frameworks whose arguments could be intuitively understood as proofs of some (possibly nonmonotonic) underlying logic over a language $\mathcal{L}$. The monotonic part of the underlying logic is assummed to be represented by a Tarski-like consequence operator $CN(X)$ for set of sentences X over $\mathcal{L}$ such that following properties are satisfied:

1. $X \subseteq CN(X)$
2. $CN(X) = CN(CN(X))$
3. $CN(X) = \bigcup\{CN(Y) \mid Y \subseteq X \text{ and } Y \text{ is finite}\}$
4. A notion of contradictory is introduced by a set $CONTRA$ of subsets of $\mathcal{L}$ ($CONTRA \subseteq 2^{\mathcal{L}}$) such that if $S \in CONTRA$ then each superset of S also belongs to $CONTRA$. A set belonging to $CONTRA$ is said to be *contradictory.*
   The set $CN(\emptyset)$ is not contradictory, i.e. $CN(\emptyset) \notin CONTRA$.

A set of sentences X is said to be **inconsistent** wrt CN if its closure $CN(X)$ is contradictory. X is said to be **consistent** if it is not inconsistent. X is closed if it coincides with its own closure.

The language in example 1 consists of literals whose atoms occur in the (strict and defeasible) rules. The consequence operator CN_0 is defined by the set $\mathcal{R}_0$ of strict rules, namely $CN_0(X)$ is the smallest (wrt set inclusion) set of literals satisfying the propositions that $X \subseteq CN_0(X)$ and for any strict rule $r \in \mathcal{R}_0$, if the premises of r are contained in $CN_0(X)$ than the head of r also belongs to $CN_0(X)$. For example $CN_0(\{m\}) = \{wr, go, m, hw\}$ and $CN_0(\{m, b\}) = \{wr, go, m, hw, b, \neg hw\}$ and $CN_0(\emptyset) = \{wr, go\}$. A contradictory set is any set containing a pair of literals $\{l, \neg l\}$. Hence the set $CN_0(\{m, b\})$ is contradictory while the set $\{m, b\}$ is inconsistent wrt CN_0.

Definition 1. *A logic-based abstract argumentation framework over a language $\mathcal{L}$ is a triple (AF, CN, Cnl) where AF is an abstract argumentation framework, CN is a Tarski-like consequence operator over $\mathcal{L}$ and for each argument A, $Cnl(A)$ is the conclusion of A.*

For a set S of arguments, $Cnl(S)$ denotes the set of the conclusions of the arguments in S. The Tarski-consequence operator has been employed in [1] to give a general definition of a logic-based argument. In contrast, we use a Tarski-like consequence operator to only specify the logical consequences of arguments without any hint about how an argument is constructed.

From now on until the end of this section, we assume an arbitrary but fixed logic-based abstract argumentation framework (AF, CN, Cnl) and often simply refer to it as an argumentation framework.

Definition 2. *Let (AF, CN, Cnl) be a logic-based abstract argumentation framework.*

1. *AF is said to satisfy the* **logical closure-property** *if for each complete extension E, $Cnl(E)$ is closed.*
2. *AF is said to satisfy the* **logical consistency-property** *if for each complete extension E, $Cnl(E)$ is consistent.*

Example 2. (Continuation of example 1) The grounded extension of the argumentation framework in example 1 is $GE = \{A_1, A_2, A_3, A_4\}$. $Cnl(GE) = \{wr, go, m, b\}$ and $CN_0(Cnl(GE)) = Cnl(GE) \cup \{hw, \neg hw\}$. Hence the considered argumentation framework satisfies neither the logical consistency- nor the closure-property.

It turns out that the properties of logical closure and consistency of argumentation is based on an intuitive idea of a base of an argument.

Definition 3. *Given a logic-based argumentation framework (AF, CN, Cnl). We say that a set of arguments S is a* **base of an argument** *A if the conclusion of A is a consequence of the conclusions of S (i.e. $Cnl(A) \in CN(Cnl(S))$) and each attack against A is an attack against S and vice versa.*

We say that a set of arguments S is a **base of a set of arguments** *R if $Cnl(R) \subseteq CN(Cnl(S))$ and each attack against R is an attack against S and vice versa.*

We say that an argument A is **based** *in a set of arguments S if S contains a base of A.*

In example 1, though $Cnl(A_5) \in CN_0(Cnl(A_3))$, the set $\{A_3\}$ is not a base of A_5 since A_6 attacks A_5 but A_6 does not attack A_3. Note that $\emptyset$ is a base of A_1 and A_2 and the set $\{A_1, A_2\}$.

Example 3. Consider a modified version of the example 1 where the set $\mathcal{R}_1$ of strict rules is obtained by adding to $\mathcal{R}_0$ two more strict rules $\neg hw \to \neg m$; and $hw \to \neg b$. The corresponding consequence operator is denoted by CN_1. There are two more arguments: $A_7 : A_5 \to \neg b$ and $A_8 : A_6 \to \neg m$. The attack relation is defined by $att_1 = \{(A_7, A_4), (A_7, A_6), (A_7, A_8), (A_8, A_3), (A_8, A_5), (A_8, A_7)\}$. $\{A_3\}$ is now a base of A_5 and A_7 and $\{A_5, A_7\}$. It is also easy to see that $\{A_3, A_4\}$ is a base of $\{A_5, A_6\}$.

Lemma 1. *Let E be a complete extension of a logic-based argumentation framework LAF. Further let S be a base of a subset of E. Then $S \subseteq E$*

Proof As each attack against S is an attack against E, each argument in S is acceptable wrt E. Hence $S \subseteq E$.

The imposition of the closure and consistency-properties on an argumentation framework wrt consequence operator suggests intuitively that if a sentence σ follows from the conclusions of a set of arguments S wrt consequence operator CN then there exists an argument A with conclusion σ constructed from some arguments in S using the rules of the underlying monotonic logic. In other words, argument A is acceptable wrt S.

Definition 4. *We say that a logic-based argumentation framework* (AF, CN, Cnl) *is* **structurally closed** *if for each set of arguments S, for each sentence* $\alpha \in CN(Cnl(S))$ *there exists an argument A based in S such that* $Cnl(A) = \alpha$*.*

The argumentation framework in example 1 is not structurally closed since although $hw \in CN_0(Cnl(A_3))$ and A_5 is the only argument with conclusion hw, A_5 is not based in $\{A_3\}$ as A_6 attacks A_5 but A_6 does not attack A_3. In contrast, the argumentation framwork generated by the set of strict rules $\mathcal{R}_1$ in example 3 together with the defeasible rules in $\mathcal{D}$ is structurally closed.

Lemma 2. *Suppose a logic-based abstract argumentation framework* $LAF = (AF, CN, Cnl)$ *is structural closed. Then* LAF *satisfies the logical closure-property.*

Proof. Let E be a complete extension. Let $\alpha \in CN(Cnl(E))$. Therefore from the structural closure, there is an argument A based in E such that $Cnl(A) = \alpha$ such that each attack against A is an attack against E. Because admissibility of E, E attacks each attack against A. Hence A is acceptable wrt E, i.e. $A \in E$. E is hence closed wrt CN.

We say that an argument A is **generated** by a set S of arguments if A is based in a base of S.

In example 3, $\{A_3\}$ is a base of both A_5 and A_7. Hence A_7 is generated by $\{A_5\}$.

We say that a set of arguments S **implicitly attacks** an argument B if there is an argument A generated by S such that A attacks B. S is said to **implictly attack itself** if S implicitly attacks an argument in S.

Consider again example 3. A base of A_3, A_6 is $\{A_3, A_4\}$. As A_7 is generated by $\{A_3, A_4\}$ and A_7 attacks A_4, $\{A_3, A_6\}$ implicitly attacks itself.

Definition 5. *A logic-based argumentation framework is said to be* **consistency covering** *if for each set of arguments S such that* $Cnl(S)$ *is inconsistent, S implicitly attacks itself.*

In the argumentation framework in 1, for the grounded extension $GE = \{A_1, A_2, A_3, A_4\}$, $Cnl(GE) = \{wr, go, b, m\}$ is inconsistent. I is not difficlt to see that $\{A_3, A_4\}$ is a base of $\{A_1, A_2, A_3, A_4\}$ and the arguments generated by $\{A_3, A_4\}$ are $\{A_1, A_2, A_3, A_4\}$. Hence GE does not implicitly attack itself. The consistency covering property is not satisfied for this framework.

In contrast, in example 3, a base for $S = \{A_1, A_2, A_3, A_4\}$ is also $\{A_3, A_4\}$, and a base of (A_7) is $\{A_3\}$. It follows that A_7 is based in S. Because A_7 attacks A_4, S implicitly attacks itself. Overall, the logic-based argumentation framework of this example satisfies the consistency covering property (see section 4 for precise proof).

Theorem 1. *Let LAF be a structural closed and consistency covering argumentation framework. Then LAF satisfies both the logical closure- and consistency-properties.*

Proof From lemma 2, we need to prove only the consistency property. Let E be a complete extension of LAF. Suppose E is inconsistent. From the consistency covering of LAF, it follows that there is an argument A generated by E attacking some argument B in E. Therefore A attacks E. E hence attacks A. Since any base of E is a subset of E (lemma 1), A is based in E. Hence any attack against A is an attack against E. E hence attacks itself. Contradiction.

In the next sections, we present two different argumentation systems slightly generalizing similar systems from the literature to demonstrate how to capture the structural-closedness and consistency covering property.

3 Abstract Assumption-based Argumentation

We assume a language $\mathcal{L}$, a set of assumptions $\mathcal{A} \subseteq \mathcal{L}$, a contrary operator $\overline{(.)} : \mathcal{A} \longrightarrow \mathcal{L}$, and a Tarski-like consequence operator CN with a set CONTRA of contradictory sets. Note that we do not assume that sets containing both α and $\overline{\alpha}$ for an assumption $\alpha \in \mathcal{A}$ belong to CONTRA. In case of normal logic programming [6, 2, 5], CONTRA is empty while for extended logic programming [7] CONTRA contains sets containing pair of literals $\{l, \neg l\}$ where $\neg$ is explicit negation[3].

An argument is a pair (X, σ) where X is a finite set of assumption X and $\sigma \in CN(X)$. An argument (X, σ) attacks an argument (Y, δ) if $\sigma = \overline{\alpha}$ for some $\alpha \in Y$.

The just defined logic-based argumentation framework is referred to in the rest of thic section as abstract assumption-based argumentation AAA.

Example 4. For illustration, consider the famous bird-fly example. Let CN be the consequence operator defined by the following set of strict rules $\{\rightarrow p;\ \ p \rightarrow b;\ \ p, np \rightarrow \neg f;\ \ b, nb \rightarrow f;\ \ p \rightarrow \neg nb\}$ with np, nb (for normal penguin and normal bird respectively) being assumptions and $\overline{np} = \neg np$ and $\overline{nb} = \neg nb$. Let $A_1 = (\{np\}, \neg f)$, $A_2 = (\{\}, \neg nb)$, $A_3 = (\{nb\}, f)$. A_2 attacks A_3.

[3] Negation as failure is denoted by not-l

Definition 6. *The consequence operator CN is said to be* **assumption-discriminate** *if for each inconsistent set of assumptions $X \subseteq \mathcal{A}$, there is $\alpha \in X$ such that $\overline{\alpha} \in CN(X)$.*

The argumentation framework in example 4 is assumption-discriminate. For illustration, the set $X = \{np, nb\}$ is inconsistent and $\neg nb \in CN(X)$.

Lemma 3. *Suppose CN is assumption-discriminate. Then the abstract assumption-based argumentation framework is structurally closed and consistency-covering.*

Proof We first show the structural closure. Let S be a set of arguments and $\sigma \in CN(Cnl(S))$. Let X be a minimal subset of $Cnl(S)$ such that $\sigma \in CN(X)$. Further let S_X be a minimal set of arguments from S whose conclusions belong to X. Let $A = (Y, \sigma)$ such that $Y = \bigcup\{Z \mid (Z, \delta) \in S_X\}$. It is obvious that A is an argument. We show now that S_X is a base of A. Suppose B is an argument attacking S_X. Then there is $(X, \delta) \in S_X$ such that $Cnl(B) = \overline{\alpha}$ for some $\alpha \in X$. Hence B attacks A. Suppose now that B attacks A. Then $Cnl(B) = \overline{\alpha}$ for some $\alpha \in Y$. Hence there is $(X, \delta) \in S_X$ such that $Cnl(B) = \overline{\alpha}$ for some $\alpha \in X$. B therefore attacks S_X.

We have proved that that the abstract assumption-based argumentation framework is structurally closed. We show now that it is consistency covering. We need some new notations. For an argument $A = (X, \sigma)$, let $NB(A) = \{(\{\alpha\}, \alpha) \mid \alpha \in X\}$. It is easy to see that $NB(A)$ is a base of A. Similarly, for a set S of arguments, $NB(S) = \bigcup\{NB(A) \mid A \in S\}$ is a base of S.

Let S be a set of arguments such that $Cnl(S)$ is inconsistent. Let $Y = NB(S)$ Hence Y is inconsistent. From the assumption-discrimination of CN, it follows that there is $Z \subseteq Y$ such that $A = (Z, \overline{\alpha})$ is an argument. As $NB(S)$ is a base of S, A is generated by S. Since A attacks each argument having α as an assumption, A attacks S. Hence S implicitly attacks itself.

It follows immediately from lemma 3 and theorem 1

Theorem 2. *Suppose CN is assumption-discriminate. Then the abstract assumption-based argumentation framework satisfies both the logical closure- and consistency-properties.*

4 Argumentation with Strict and Defeasible Rules

In this section, we apply our results developed in previous section on a defeasible logic similar to the one studied by [3, 8, 10].

The language $\mathcal{L}$ is a set of literals. A set of literals is said to be contradictory if it contains a pair $\{l, \neg l\}$. The set of all contradictory sets is denoted by CONTRA. Arguments are built from strict rules and defeasible rules. The set of strict rules is denoted by $\mathcal{R}_s$ while the set of defeasible rules by $\mathcal{R}_d$. Strict rules are of the form $l_1, \ldots, l_n \longrightarrow h$ and defeasible rules of the form $l_1, \ldots, l_n \Rightarrow h$ where $l_1, \ldots, l_n, h$ are literals from $\mathcal{L}$.

Definition 7. *Let $\alpha_1, \ldots, \alpha_n \to \alpha$ (respectively $\alpha_1, \ldots, \alpha_n \Rightarrow \alpha$) be a strict (respectively defeasible) rule. Further suppose that $A_1, \ldots, A_n$, $n \geq 0$, are arguments with $\alpha_i = Cnl(A_i)$, $1 \leq i \leq n$. Then $A_1, \ldots, A_n \to \alpha$ (respectively $A_1, \ldots, A_n \Rightarrow \alpha$) is an argument with conclusion α.*

Arguments of the form $A_1, \ldots, A_n \to \alpha$ or $A_1, \ldots, A_n \Rightarrow \alpha$ are also often viewed as proof trees with the root labelled by α and $A_1, \ldots, A_n$ are subtrees whose roots are children of the proof tree root.

A strict argument is an argument containing no defeasible rule.

B is a subargument of an argument A, denoted by $B \sqsubseteq A$ if $B = A$ or B is a subargument of some A_i if A is of the form $A_1, \ldots, A_n \to \alpha$ or $A_1, \ldots, A_n \Rightarrow \alpha$.

The consequence operator $CN_{\mathcal{R}_s}(X)$ (or simply $CN(X)$ if no misunderstanding is possible) is defined by the set of conclusions of strict arguments over the set of rules $\mathcal{R}_s(X) = \mathcal{R}_s \cup \{\to \alpha \mid \alpha \in X\}$.

For a strict argument A over a set of rules $\mathcal{R}_s(X)$, the set of premises of A, denoted by $Prem(A)$, is the set of literals from X labelling the leaves of A (viewed as a proof tree).

Basic arguments are arguments whose last rule is a defeasible one. For a basic argument B, the last rule of B is denoted by $Lr(B)$.

The following notion of attack is adopted but slightly modified from the ones given in [3, 8–10].

An argument A attacks a argument B if one of the following conditions is satisfied:

1. *(Undercutting)* B is basic and $Cnl(A) = \neg Oj(Lr(B))$ where for a defeasible rule r, $Oj(r)$ is a literal denoting that the rule is applicable.
2. *(Rebutting)* B is basic and $Cnl(A) = \neg Cnl(B)$
3. A attacks a basic subargument of B.

An example of argumenttion based on strict and defeasible rules is given in example 3.

Definition 8. *The consequence operator $CN_{\mathcal{R}_s}$ is said to be* **discriminate** *if for each inconsistent set X of literals, there is a literal $\sigma \in X$ such that $\neg\sigma \in CN_{\mathcal{R}_s}(X)$ holds.*

Theorem 3. *Let $\mathcal{R}_s, \mathcal{R}_d$ be sets of strict and defeasible rules respetively. Let $\mathcal{AR}$ be the arguments built from these rules and att be the associated attack relation and $AF = (\mathcal{AR}, att)$. Then the logic-based argumentation framework $LAF = (AF, CN_{\mathcal{R}_s}, Cnl)$ is structurally closed and consistency covering if the consequence operator $CN_{\mathcal{R}_s}$ is discriminate.*

Proof We first show that LAF is structurally closed. Suppose now that $\sigma \in CN(Cnl(S))$. Let X be a minimal subset of $Cnl(S)$ such that $\sigma \in CN(X)$. Hence there is a strict argument A_0 over $\mathcal{R}_s(X)$ with conclusion σ. Further let S_X be a minimal set of arguments from S whose conclusions belong to X. Let A be the argument obtained by replacing each leave in A_0 (viewed as a proof tree) labelled by a literal α from X by an argument with conclusion α from S_X. It is obvious that the conclusion of A is σ. We show now that S_X is a base of A. Suppose B is an argument attacking S_X. Then it is obvious that B attacks A. Suppose now that B attacks A. Then B attacks a basic argument of A. Since A_0 is a strict argument over $\mathcal{R}_s(X)$, B must attacks a basic subargument of some argument in S_X. Hence B attacks S_X.

We have proved that that the logic-based argumentation framework LAF is structurally closed. We show now that it is consistency covering. We need some new notations.

Among the bases of arguments, a special kind of base called normal base plays a key role. The **normal base** of argument A is defined by $NB(A) = \{B \mid B$ is a basic subargument of A and for each argument C, if $C \neq B$ and $B \sqsubseteq C \sqsubseteq A$ then C is strict $\}$. For a set S of arguments, $NB(S)$ is the union of the normal bases of elements of S. The following lemma shows that a normal base is indeed a base.

Lemma 4. *For any argument A, $NB(A)$ is a base of A.*

Proof of Lemma 4 From the definition of NB(A), it is obvious that $Cnl(A) \in CN_{\mathcal{R}_f}(NB(A))$. It is also obvious that each argument attacking $NB(A)$ also attacking A. Let B be an argument attacking A. From the definition of attack, B attacks a basic subargument C of A. From the definition of $NB(A)$, there is an argument $C' \in NB(A)$ such that $C \sqsubseteq C'$. Hence B attacks C' and therefore $NB(A)$.

Continuation of Proof of Theorem 3 It is obvious that NB(S) is also a base of set of arguments S. Suppose now that $Cnl(S)$ is inconsistent. It follows immediately that the set $Cnl(NB(S))$ is also inconsistent. Let $X = Cnl(NB(S))$. From the definition 5, it follows that there is $\alpha \in X$ such that $\neg\alpha \in CN_{\mathcal{R}_s}(X)$. Since $\alpha \in X$, there is a basic argument $B \in NB(S)$ with conclusion α. From the structural closure of LAF, there is an argument A with conclusion $\neg\alpha$ based in $NB(S)$. Hence A is generated by S and A attacks B. As $B \in NB(S)$, there is $C \in S$ s.t. $B \in NB(C)$. Hence A attacks C. Therefore S implicitly attacks itself.

It follows immediately from theorem 1

Theorem 4. *Let $\mathcal{R}_s, \mathcal{R}_d$ be sets of strict and defeasible rules respectively. Then the associated logic-based abstract argumentation framework $LAF = (AF, CN_{\mathcal{R}_s})$ satisfies the logical closure- and consistency-properties if $CN_{\mathcal{R}_s}$ is discriminate.*

We next show that theorem 4 generalizes the results in Caminada and Amgoud [3], Prakken [8].

A set of strict rules $\mathcal{R}_s$ is said to be closed under transposition if for each rule $\alpha_1, \ldots, \alpha_n \to \sigma$ in $\mathcal{R}_s$, all the rules of the form

$\alpha_1, \ldots, \alpha_{i-1}, \neg\sigma, \alpha_{i+1}, \alpha_n \to \neg\alpha_i$ also belong to $\mathcal{R}_s$.

A set of strict rules $\mathcal{R}_s$ is said to satisfy the contraposition-property if for each set of literals X, for each argument A (with conclusion σ) wrt $\mathcal{R}_s(X)$ and for each

$\alpha \in Prem(A)$, there is an argument whose premises is $Prem(A) - \{\alpha\} \cup \{\neg\sigma\}$ and conclusion is $\neg\alpha$.

Theorem 5. *$CN_{\mathcal{R}_s}$ is discriminate if the set of strict rules $\mathcal{R}_s$ is closed under transposition or satisfies the contraposition-property.*

Proof We first prove the following assertion.
Assertion: Let A be a strict argument over $\mathcal{R}_s(X)$ whose conclusion is σ and $\emptyset \neq Prem(A) \subseteq X$. Then there is an argument B with premises in $Prem(A) \cup \{\neg\sigma\}$ and conclusion $\neg\alpha$ for some $\alpha \in Prem(A)$.
Proof of Assertion The assertion holds immediately if $\mathcal{R}_s$ satisfies the contraposition-property.

Supoose that $\mathcal{R}_s$ is closed under transposition. We prove by induction on the height of A (as a proof tree).

If the heitght of A is 1, the theorem is obvious.

Suppose A is of the form $A_1, \ldots, A_n \to \sigma$ where $Cnl(A_i) = \alpha_i$. Suppose $Prem(A_n) \neq \emptyset$. From the closure under transposition, $\alpha_1, \ldots, \alpha_{n-1}, \neg\sigma \to \neg\alpha_n$ also belongs to $\mathcal{R}_s$. Let A_0 be the argument $A_1, \ldots, A_{n-1}, \neg\sigma \to \neg\alpha_n$.

From the induction hypothesis, there is a tree C whose premises in $Prem(A_n) \cup \{\neg\alpha_n\}$ and whose conclusion is $\neg\alpha$ for some $\alpha \in Prem(A_n)$.

Let B be the tree obtained from C by replacing each occurence of premise $\neg\alpha_n$ by the argument A_0. It is clear that $Prem(B) \subseteq Prem(A) \cup \{\neg\sigma\}$ and $Cnl(B) = \neg\alpha$. Note $\alpha \in Prem(A)$.

Continuation of Proof of Theorem 5. Let X be an inconsistent set of literals. Hence there are two arguments A_0, A_1 with premises in X and conclusions $\sigma, \neg\sigma$ respectively. From the above assertion, it follows that there exists an argument B with conclusion $\neg\alpha$ for some $\alpha \in Prem(A_0)$ and $Prem(B) \subseteq Prem(A_0) \cup \{\neg\sigma\}$. Let A be the argument obtained by replacing leaves labelled by $\neg\sigma$ in B by trees A_1. It is clear that $Prem(A) \subseteq X$ and the conclusion of A is labelled by $\neg\alpha$ for some $\alpha \in X$.

5 Conclusion

In general, an argumentation system could take a large number of arguments, many of them could be redundant. For efficiency, many of the redundant arguments should be avoided. In [4], principles for dealing with redundant arguments have been studied. It would be interesting to see how such principles could be integrated with the concepts of structural closure and consistency covering for modeling practical argumentation.

6 Acknowledgements

Many thanks to Marcello Balduccini and Tran Cao Son for allowing us to contribute to this event.

References

1. Amgoud, L., Besnard, P. "Bridging the gap between abstract argumentation systems and logic", Proceedings of SUM 2009, LNCS 5785, Springer Verlag.
2. Bondarenko, A., Dung P.M., Kowalski R., Toni F. "An abstract argumentation-theoretic approach to default reasoning", Artificial Intelligence, 93, 63-101, 1997.
3. Caminada, M., Amgoud, L. "On the evaluation of argumentation formalisms", Artificial Intelligence, 2007, 171, 286-310
4. Dung, P.M., Toni F., Mancarella P. "Some design guidelines fo practical argumentation systems", Third International Conference on Computational Models of Argument, Italy, 2010
5. Dung, P. M. "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games", Artificial Intelligence, 77, 321-357, 1995.
6. Gelfond M., Lifschitz, V. "The stable model semantics of logic programs", Proc of ICLP 1988, MIT Press.
7. Gelfond M., Lifschitz, V. "Logic programs with classical negation", Proc of ICLP 1990, MIT Press.
8. Prakken, H. An abstract framework for argumentation with structured arguments, Journal of Argumentation and Computation, Vol 1, 2010.
9. Pollock, J. "Defeasible Reasoning", Cognitive Science, 11, 481-518, 1987
10. Prakken, H., Sartor, G. "Argument-based extended logic programming with defeasible priorities", J. of Applied Non-Classical Logics, 7, 25-75, 1997