Risk assessment as an argumentation game

Henry Prakken[1,2], Dan Ionita[3], and Roel Wieringa[3]

[1] Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands,
h.prakken@uu.nl
[2] Faculty of Law, University of Groningen, Groningen, The Netherlands
[3] Department of Computer Science, University of Twente, Enschede, The Netherlands,
d.ionita@student.utwente.nl,r.j.wieringa@ewi.utwente.nl

Abstract. This paper explores the idea that IT security risk assessment can be formalized as an argumentation game in which assessors argue about how the system can be attacked by a threat agent and defended by the assessors. A system architecture plus assumptions about the environment is specified as an *ASPIC*$^+$ argumentation theory, and an argument game is defined for exchanging arguments between assessors and hypothetical threat agents about whether the specification satisfies a given security requirement. Satisfaction is always partial and involves a risk assessment of the assessors. The game is dynamic in that the players can both add elements to and delete elements from the architecture specification. The game is shown to respect the underlying argumentation logic in that for any logically completed game 'won' by the defender, the security requirement is a justified conclusion from the architecture specification at that stage of the game.

1 Introduction and motivation

This paper explores the idea that IT security risk assessment can be formalized as an argumentation game in which assessors alternate between playing the role of defenders and attackers of the system, arguing how the system can be defended and attacked, respectively. Our long-term goal is that such a formalization is used to develop tool support for human assessors during a risk assessment, to keep track of the arguments for and against a security architecture. Two characteristics of IT security risk assessment (RA) as it happens in practice are that the time available for doing the assessment is limited, and that the resources of the defender to protect a system, and of the attacker to attack the system, are limited too. Assessors have limited, qualitative information about the system, its vulnerabilities, and threats. In addition, malice and accident have to be taken into account [5]. As a consequence, the information involved in risk assessments is highly defeasible and cannot be easily quantified, which motivates an argumentation approach to RA instead of, for example, Bayesian or model-checking approaches.

Some current risk assessment frameworks also provide tool support for security discussions, but they are mostly geared toward communication between the stakeholders and the risk analysts or towards dissemination of the *results* of the risk assessments. One example is the CORAS tool [8], which uses UML-based diagrams on top of which several kinds of elements and relationships are defined as to allow a visual representation of the risk assessment. Their "risk diagrams" and "treatment diagrams" describe

the possible attacks and mitigations that were discussed during the assessment. We ultimately also want to provide support for the *process* of RA, to keep track not only of the final output of an RA but also of the assumptions and design decisions that were made during the assessment. Another reason for supporting the process of RA is the limited time and resources available for the risk assessors. This puts constraints on the assessment process that call for efficient and effective assessment. For this reason we take a dialogue game approach, since dialogue systems for argumentation are recognised in the literature as a way to promote effective and efficient debate [7].

We also want to contribute to the literature by taking a formal but non-quantative approach. Current RA practice is fully informal and uses checklists to assess threats to a system. Current formalizations assume quantitative information, an assumption that is often not warranted. The only approach that uses argumentation, uses Toulmin argument diagrams and is still informal [5, 4]. By formalising the RA argumentation process in state-of the-art AI formalisms, we aim to give a precise semantics to the use of argumentation in RA and to make well-founded computational tools available for the support of the RA process.

In more detail, our idea is that an RA game starts with a defeasible argument by the defenders that the current system architecture is sufficient to guard against attacks; the argument is defeasible because it will make assumptions about the vulnerability of some system components and about capabilities, resources or risk appetite of attackers. In an attacker round, the (assessors playing the role of) attackers defeat some arguments of the (assessors playing the role of) defenders by attacking the defenders' assumptions, rebutting defeasible conclusions or by undercutting a defeasible inference made by the defender. After an attacker round, the architecture of the system may be changed by the defender to falsify some assumptions made by the attacker, and they may change their assumptions about the attackers. Then they will play the defender round again with the new system architecture, by renewing their argument for the security of the system. The renewed argument is still defeasible, for the same reasons as indicated above. The new argument may even contain parts of their old argument that have been undermined, undercut or rebutted by the attacker. This depends on the defenders' risk assessment and risk appetite. If there is time left, more attacker rounds, redesigns and defender's rounds are played. The game ends when time is up, and the goal is to end it in a state where the defenders estimate the arguments of the defense stronger than the arguments of the attackers, given the defender's assumptions about the environment and risk appetite.

Our primary goal in this paper is to test the feasibility of the idea of modelling risk assessment as an argumentation game by giving a first formalization. A special feature of our argumentation game is that the arguments are not simply constructed from a given theory, but that the theory is itself dynamically constructed during the RA: the players can add new elements to the theory (such as descriptions of system elements, preferences or assumptions about the environment) and they can also delete or change elements from the theory (for example, if the system specification has to be changed because of an attack that exposes a risk). This is what risk assessors do in practice and our game can therefore not just be a logical argument game for testing the acceptability status of an argument in a given information state, but must allow for changes in the information state. This is another reason why we take a dialogue game approach to

RA, since dialogue games allow for such dynamics. The logical part of the game will be an instantiation of the *ASPIC*$^+$ framework of [15, 11]. This choice is in order to profit from the logical consistency and closure properties of *ASPIC*$^+$ and since the application requires explicit preferences and defeasible rules. The dialogical part of our game combines the framework of [13, 14] with some new elements.

In Section 2 we present the logical background, in the form of an instantiation of the *ASPIC*$^+$ framework. In Section 3 we sketch how this instantiation can be used to specify an architecture and to express security risk assessment arguments. In Section 4 we present our formal dialogue game and prove a correspondence property with the underlying logic. In Section 5 we illustrate the game with an example, and we conclude with a discussion of related work and future research.

2 The Formal Setting

An *abstract argumentation framework* (AF) is a pair $\langle \mathcal{A}, defeat \rangle$, where $\mathcal{A}$ is a set arguments and $defeat \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation. The theory of AFs then addresses how sets of arguments (called *extensions*) can be identified which are internally coherent and defend themselves against defeat. A key notion here is that of an argument being *acceptable with respect to*, or *defended by* a set of arguments: $A \in \mathcal{A}$ is defended by $S \subseteq \mathcal{A}$ if for all $A \in S$: if $B \in \mathcal{A}$ attacks A, then some $C \in S$ attacks B. Then relative to a given AF various types of extensions can be defined. In this paper we focus on the grounded extension, which is defined as follows :

- $E \subseteq \mathcal{A}$ is the *grounded extension* if E is the least fixpoint of operator F, where $F(S)$ returns all arguments defended by S.

A proof procedure in the form of a logical argument game between a proponent and an opponent can be used to test whether a given argument is in the grounded extension. Informally, the proponent starts a game with the argument to be tested and then the players take turns, trying to defeat the previous move of the other player. In doing so, the proponent must strictly defeat the opponent's arguments. A game is terminated if the player to move has no arguments to play and a game is won by the player who moves last. Then an argument is proven to be justified if the proponent has a winning strategy for it, that is, if he can make the opponent run out of moves whatever choice the opponent makes. A winning strategy is in fact a tree with as root the argument to be tested and then at even depth all defeaters of the parent node while at odd depth one strict defeater of the parent node.

Our reason for using grounded semantics is that we want to build a logical argument game into our dialogue game for argumentation, since this is a natural way to make the outcome of an argumentation dialogue agree with the underlying logic. Since we ultimately intend to provide support tools for human risk assessors, the tool must be simple and intuitive, and grounded semantics has, as just explained, a particularly simple and intuitive logical argument game. However, in our future research we want to investigate generalisation to other semantics.

ASPIC$^+$ [15, 11] is a general framework for structured argumentation. It defines the notion of an *argumentation system*, which consists of a logical language $\mathcal{L}$ with

a binary contrariness relation $^-$ and two sets of inference rules $\mathcal{R}_s$ and $\mathcal{R}_d$ of *strict* and *defeasible inference rules* defined over $\mathcal{L}$, written as $\varphi_1, \ldots, \varphi_n \rightarrow \varphi$ and $\varphi_1, \ldots, \varphi_n \Rightarrow \varphi$. Informally, that an inference rule is strict means that if its antecedents are accepted, then its consequent must be accepted *no matter what*, while that an inference rule is defeasible means that if its antecedents are accepted, then its consequent must be accepted *if there are no good reasons not to accept it*. An argumentation system also contains a function n which for each defeasible rule in $\mathcal{R}_d$ returns a formula in $\mathcal{L}$. Informally, $n(r) \in \mathcal{L}$ expresses that $r \in \mathcal{R}$ is applicable.

In the present paper we assume argumentation systems in which $\mathcal{L}$ consists of first-order predicate-logic literals (i.e., atomic formulas or their negation) and its contrariness relation corresponds to classical negation, and in which the n function should be obvious from the examples.

ASPIC$^+$ arguments chain applications of the inference rules from AS into inference trees, starting with elements from a *knowledge base* $\mathcal{K}$. In this paper we assume that all premises are so-called *axiom premises*, that is, they are not attackable. In what follows, for any argument A, $\texttt{Prem}$ returns all the formulas of $\mathcal{K}$ (*premises*) used to build A, $\texttt{Conc}$ returns A's conclusion, $\texttt{Sub}$ returns all of A's sub-arguments, $\texttt{Rules}$ and $\texttt{DefRules}$ respectively return all rules and all defeasible rules in A, and $\texttt{TopRule}(A)$ returns the last rule applied in A.

Definition 1. *An **ASPIC**$^+$**argument** A on the basis of a knowledge base $\mathcal{K}$ in an argumentation system $(\mathcal{L}, ^-, \mathcal{R}, n)$ is:*

1. *φ if $\varphi \in \mathcal{K}$ with:* $\texttt{Prem}(A) = \{\varphi\}$; $\texttt{Conc}(A) = \varphi$; $\texttt{Sub}(A) = \{\varphi\}$; $\texttt{Rules}(A) = \emptyset$; $\texttt{TopRule}(A)$ = *undefined.*
2. *$A_1, \ldots A_n \not\rightarrow\!\!\!\!\Rightarrow \psi$ if $A_1, \ldots, A_n$ are finite arguments such that there exists a strict/defeasible rule $\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \not\rightarrow\!\!\!\!\Rightarrow \psi$ in $\mathcal{R}_s$/$\mathcal{R}_d$.*
   $\texttt{Prem}(A) = \texttt{Prem}(A_1) \cup \ldots \cup \texttt{Prem}(A_n)$, $\texttt{Conc}(A) = \psi$,
   $\texttt{Sub}(A) = \texttt{Sub}(A_1) \cup \ldots \cup \texttt{Sub}(A_n) \cup \{A\}$.
   $\texttt{Rules}(A) = \texttt{Rules}(A_1) \cup \ldots \cup \texttt{Rules}(A_n) \cup \{\texttt{Conc}(A_1), \ldots, \texttt{Conc}(A_n) \not\rightarrow\!\!\!\!\Rightarrow \psi\}$,
   $\texttt{DefRules}(A) = \{r | r \in \texttt{Rules}(A), r \in \mathcal{R}_d\}$,
   $\texttt{TopRule}(A) = \texttt{Conc}(A_1), \ldots \texttt{Conc}(A_n) \not\rightarrow\!\!\!\!\Rightarrow \psi$

An argument A is strict *if* $\texttt{DefRules}(A) = \emptyset$ *and* defeasible *if* $\texttt{DefRules}(A) \neq \emptyset$.

Example 1. Consider a knowledge base in an argumentation system with

$\mathcal{R}_s = \{p, q \rightarrow s;\ u, v \rightarrow w\}$, $\mathcal{R}_d = \{p \Rightarrow t;\ s, r, t \Rightarrow v\}$
$\mathcal{K} = \ \{q, p, r, u\}$

An argument for w and its subarguments are written as follows:

A_1: p $\quad A_2$: q $\quad A_5$: $A_1 \Rightarrow t$ $\qquad\qquad A_6$: $A_1, A_2 \rightarrow s$
A_3: r $\quad A_4$: u $\quad A_7$: $A_5, A_3, A_6 \Rightarrow v$ $\quad A_8$: $A_7, A_4 \rightarrow w$

We have that

$\texttt{Prem}(A_8) = \{p, q, r, u\}$; $\texttt{Conc}(A_8) = w$
$\texttt{Sub}(A_8) = \{A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8\}$
$\texttt{DefRules}(A_8) = \{p \Rightarrow t;\ s, r, t \Rightarrow v\}$; $\texttt{TopRule}(A_8) = v, u \rightarrow w$

An argumentation system and a knowledge base are combined with an *argument ordering* into an *argumentation theory*. The argument ordering could be defined in any way. In this paper we assume a so-called last-link ordering defined in terms of a total preorder on $\mathcal{R}_d$. Informally, the last-link ordering compares arguments on their last-used defeasible rules. For the formal definition see [11].

Definition 2. *[Argumentation theories] An* argumentation theory *is a triple* $AT = (AS, \mathcal{K}, \preceq)$ *where* AS *is an argumentation system,* $\mathcal{K}$ *is a knowledge base in* AS *and* $\preceq$ *is the last-link ordering in the sense of [11] on the set of all arguments that can be constructed on the basis of* $\mathcal{K}$ *in* AS*, assuming a total preordering* $\leq$ *on* $\mathcal{R}_d$*. That* $A \preceq B$ *means that* B *is at least as preferred as* A*. The symbols* $\prec$*,* $<$ *and* $\approx$ *are defined as usual. All this is likewise for* $\leq$*.*

In the present instantiation of *ASPIC*$^+$ arguments can be attacked in two ways: by attacking a conclusion of a defeasible inference (rebutting attack) or by attacking the defeasible inference itself (undercutting attack). To define how a defeasible inference can be attacked, the function n of an AS can be used, which assigns to each element of $\mathcal{R}_d$ a well-formed formula in $\mathcal{L}$. Recall that informally, $n(r)$ (where $r \in R_d$) means that r is applicable.[4]

Definition 3. *[attacks]* A attacks B *iff* A undercuts *or* rebuts B*, where:*

- A undercuts *argument* B *(on* B'*) iff* $\texttt{Conc}(A) = -n(r)$ *for some* $B' \in \texttt{Sub}(B)$ *such that* B'*'s top rule* r *is defeasible.*
- A rebuts *argument* B *(on* B'*) iff* $\texttt{Conc}(A) = -\varphi$ *for some* $B' \in \texttt{Sub}(B)$ *of the form* $B''_1, \ldots, B''_n \Rightarrow \varphi$*.*

Example 2. In Example 1 argument A_8 can be (indirectly) rebutted on its subargument A_5 with an argument for $\neg t$ and on its subargument A_7 with an argument for $\neg v$, because both A_5 and A_7 have a defeasible top rule. Whether these rebuttals are symmetric depends on whether the rebutting arguments use a strict or defeasible top rule. If the argument for $\neg t$ uses a defeasible top rule, then it is in turn rebutted by A_5; likewise, if the argument for $\neg v$ uses a defeasible top rule, then it is in turn rebutted by A_7. However, A_8 itself does not rebut these arguments for $\neg t$ and $\neg v$. Note that a direct rebuttal of A_5 indirectly rebuts not just A_8 but also A_7. Note also that A_8 cannot be rebutted (on A_8) with an argument for $\neg w$ or (on A_2) with an argument for $\neg s$, since both A_2 and A_8 have a strict top rule. For the same reason A_8 cannot be undercut on A_2 or A_8. It can be undercut, however, on its subarguments A_5 and A_7, with arguments for, respectively, the conclusions $\neg n(p \Rightarrow t)$ and $\neg n(s, r, t \Rightarrow v)$. Again, an undercutter of A_5 indirectly undercuts not just A_8 but also A_7.

Attacks combined with the preferences defined by an argument ordering yield two kinds of defeat.

Definition 4. *[Successful rebuttal and defeat]*

- A successfully rebuts B *if* A *rebuts* B *on* B' *and* $A \not\prec B'$*.*

[4] Henceforth $-\neg\varphi$ denotes φ, while if φ does not start with a negation, $-\varphi$ denotes $\neg\varphi$.

– A defeats B *iff* A *undercuts or successfully rebuts* B.

The success of rebutting attacks thus involves comparing the conflicting arguments at the points where they conflict. For undercutting attack no preferences are needed to make it succeed, since undercutters state exceptions to the rule they attack.

ASPIC$^+$ thus defines a set of arguments with a binary relation of defeat, that is, it defines abstract argumentation frameworks in the sense of [3]. Formally:

Definition 5. *[Argumentation framework] An* abstract argumentation framework (AF) corresponding to an argumentation theory AT *is a pair* $<\mathcal{A},$ Def$>$ *such that:*

– $\mathcal{A}$ *is the set of arguments on the basis of* AT *as defined by Definition 1,*
– Def *is the relation on* $\mathcal{A}$ *given by Definition 4.*

Thus any semantics for abstract argumentation can be applied to *ASPIC*$^+$. As noted above, in this paper we will use grounded semantics. A formula φ from $\mathcal{L}$ is then *justified* on the basis of AT if the grounded extension of the AF corresponding to AT contains an argument with conclusion φ.

3 Architecture specification in *ASPIC*$^+$

In this section we present a motivating example and describe how it can be formalized in terms of *ASPIC*$^+$.

3.1 An example with a PIN entry device

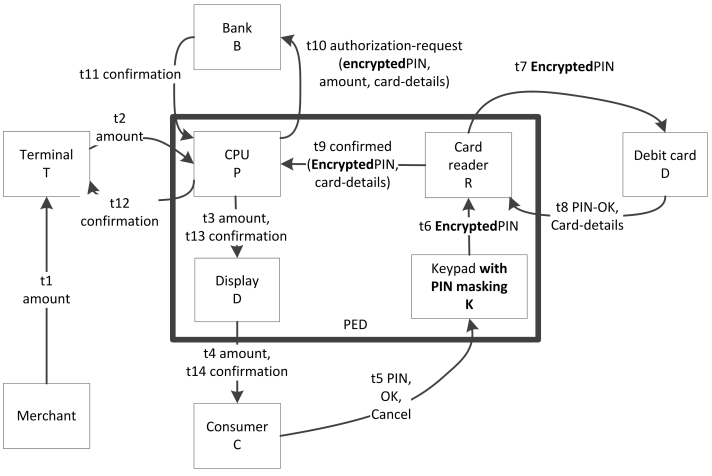

Fig. 1. Architecture of a Pin Entry Device (PED) and its context. The properties in bold are absent from the original architecture and have been added in the second round of the argument game. The labels are for ease of reference.

Our running example is a design for a Pin Entry Device (PED) that can be used by merchants in shops and restaurants. Figure 1 shows the architecture of a fixed PED, which is connected to a terminal and receives the amount to be paid from the terminal. The core functional requirement for the PED is

- FR1 Consumers can pay with a PED using a PIN.

Figure 1 shows the architecture of the PED and part of the context. Consider first the architecture without the bold annotations. The top-level informal argument for functional correctness of the architecture is given by tracing the interctions between components through the architecture roughly in the order in which we numbered them. This argument assumes that all components are implemented correctly according to their specification, that all interactions between components are reliable and that no other interactions, invisible in the diagram, occur.

Attackers keep the assumption that all components are implemented correctly, but violate the other two: They will try to change the interactions in the architecture or context (for example by changing the communication with the bank to their advantage) or will try to add additional interactions (for example by reading the PIN remotely). To make this less likely to happen, we require that the PED and its context satisfy the following properties:

SR1 PIN shall remain confidential during payment transactions
SR2 PIN communicated between nodes of the network shall remain accurate during transactions

There is no way to justify that the original architecture of figure 1 satisfies these properties. The defenders now change the architecture a bit (the bold annotations in figure 1) and make additional assumptions about the context (for example that the Consumer keeps PINs secret). The job of satisfying properties SR1 and SR2, and hence the responsibility for mitigating the risk of violating SR1 and SR2, is thus divided over the PED and its context. With the improved architecture and the additional context assumptions, the defenders can refute the argument of the attacker and reason that SR1 and SR2 are now satisfied.

3.2 Formalizing the example in *ASPIC*$^+$

We formalize this example as follows. Our general idea is that input-output relations between the components of a system are formalised as defeasible rules, while assumptions about the environment are stated as facts, which for convenience we represent a defeasible rules with empty antecedent. An argument that the system satisfies the security requirement then applies the defeasible rules to the assumptions, and is thus of a hypothetical character.

First, we represent the architecture in *ASPIC*$^+$ by a set of defeasible rules of the form $\texttt{C1!m} \Rightarrow \texttt{C2?m}$, meaning that if `C1` outputs message `m`, `C2` receives message `m`. These rules claim that communication in the system is reliable. For example, in figure 1,

(t5): $\texttt{C!PIN} \Rightarrow \texttt{K?PIN}$.

There is one such rule in $\mathcal{R}_d$ for each labeled interaction in figure 1.

Second, we assume that the assessors share defeasible beliefs about security properties of the communications between the components in the architecture. For example, in figure 1,

(conf-t5): C!confidentialPIN $\Rightarrow$ K?confidentialPIN.

This rule says that if a PIN was confidential when sent, it is still confidential when received by the keypad. These rules cannot be derived from the diagram; it is expert knowledge based on the diagram and the assessors can use it in their argumentation.

Third, we assume that the experts know the capabilities of each component. For example,

(K): K?PIN $\rightarrow$ K!encryptedPIN.

There are many of these rules for each component, and they jointly represent the knowledge that the assessors have of the capabilities of the component. This is a strict rule, as we (and the attackers) assume that each component is functioning correctly.[5]

Fourth, the experts also know how security properties are handled by each component. For example,

(conf-K): K?confidentialPIN $\rightarrow$ K!encryptedConfidentialPIN.

This rule says that if the PIN was confidential when entered in the keypad, it is still confidential after being sent in encrypted form.

Fifth, we assume that the confidentiality requirement SR1 "PIN shall remain confidential during payment transactions" is formalized as SR1

(SR1): confidentialPIN.

In general, any requirement to be verified is represented as the consequent of a defeasible rule in the architecture description and does not occur in the antecedent of any rule.

The assessors share knowledge about the meaning of the requirement in the form of a set of strict rules that for each component X,

(CRX): confidentialPIN $\rightarrow$ X!confidentialPIN.

So any non-confidential PIN transfer will violate the requirement.

Sixth, to prove a requirement, we need assumed facts, which are included in $\mathcal{R}_d$ as a set of defeasible rules with empty antecedents. Such rules are given the lowest priority in the ordering on $\mathcal{R}_d$; they are called *assumptions*.

In the first round of the game, defenders argue that the system is functionally correct, assuming that confidentialPIN is true. Attackers then try to imagine violations of SR1. For example, from the assumption that the consumer keeps her PIN confidential,

(C-keep-PIN-conf): $\Rightarrow$ C!confidentialPIN

[5] In our formalisation in Section 4 we will also include the so-called transpositions of strict rules, in order to inherit the logical closure and consistency results proven in [15, 11] about the *ASPIC*$^+$ framework.

defenders derive that the PIN received by the keypad is confidential, K?confidentialPIN using rule (conf-t5).

There are many ways in which the assumed fact (C-keep-PIN-conf) can be violated, one of which is a successful social engineering attack on a consumer [4], for instance, forcing the user to reveal the PIN. Defenders and attackers know that

(Attack-C-SE): SuccessfulSocialAttack $\Rightarrow \neg$C!confidentialPIN.

Switching to the role of attackers, the assessors now add the assumption

(Successful-attack-C-SE): $\Rightarrow$ SuccessfulSocialAttack.

This gives a rebutting attack on the initial proof of PIN confidentiality, and it proves violation of the confidentiality requirement (SR1).

To be able to allocate risk to various actors, we now assume that all users of the PED payment infrastructure support the argument. The assessors can now transfer the responsibility for beating a social engineering attack to the consumer, by simply stating that it does not occur:

(No-successful-attack-C-SE): $\neg$SuccessfulSocialAttack.

This is not a change in the architecture but a change in assumptions (this time unattackable) about the environment that reinstates the original security argument.

To illustrate how responsibility for guarding against a security requirement violation can be shifted to the PED, consider the following. In the original architecture, the PED had no PIN masking device (a cover that hides the keypad from view). If this is expressed as an assumption, then in that architecture, the attacker can rebut (conf-t5):

(not-conf-t5-masking): $\Rightarrow \neg$KwithMasking?confidentialPIN
(not-K-keep-PIN-conf): $\neg$KwithMasking?confidentialPIN $\Rightarrow \neg$K?confidentialPIN

Defenders will then change the architecture by adding PIN masking, expressed by adding the following fact to $\mathcal{K}$:

(masking): KwithMasking?confidentialPIN

and by changing (conf-t5) into

(conf-t5-masking): C!confidentialPIN, KwithMasking?confidentialPIN
$\Rightarrow$ K?confidentialPIN.

So far, we have shown that simple security arguments can be represented in an argumentation theory that is partly represented in an architecture model and partly in the knowledge and beliefs of the assessors. To play the risk argumentation game, we need to extend the argumentation theory with a dialogue game. We introduce such a game in the next section.

4 An argument game

4.1 Ideas

We now informally sketch a dialogue game for argumentation between a defender and an attacker of a design, who want to test whether a given safety or security requirement

SR is satisfied by the design. The players exchange arguments and counterarguments and during the dialogue dynamically build a joint *ASPIC*$^+$ argumentation theory describing a design and its environment. The defender's task is to ensure that the theory expresses a design that satisfies SR, while the attacker's task is to produce successful attacks on the defender's security arguments. Despite this dialectical setting, the players are cooperative in that they both want a good design that meets the requirements: their real goal is not to win but to collaborate on creating a design by critically discussing its pros and cons. For this reason we will not build rules into our dialogue game that would prevent 'selfish' players from playing moves just to obstruct the other player (such as nonrepetition moves).

The game starts with an initial argumentation theory as described in Section 3. In the first move the defender presents an argument for SR based on the initial theory and assumptions about the world. Then the players take turns after each move. The attacker's task is to defeat defender's 'current' argument for SR, after which the defender must either show that the attacker's attack is flawed (by in turn strictly defeating it) or by modifying the design in such a way that again an undefeated argument can be built that SR is satisfied. The defender can modify a design by deleting existing rules and (if needed) adding new rules as part of a new argument. The attacker cannot delete rules from the theory, because it cannot modify the design, but it can add new rules just as the defender can. Moreover, both players can add new rule priorities to make their rebutting arguments (strictly) defeat their target (but they must in doing so respect that properties of a preorder). Likewise, they can add new rule names to $\mathcal{L}$ to express undercutting defeaters. Another requirement is that each move must succeed in the mover's dialectical goal: after each defender move an argument for SR must be dialogically acceptable or *in* (in a sense to be defined below) while after each attacker move all arguments for SR must be dialogically *out* (also a sense to be defined below).

4.2 The game defined

We now define a dialogue game for a single security requirement SR. Throughout this section the logical language $\mathcal{L}$ is assumed fixed but all other elements of an AT can vary. Unless specified otherwise, the following definitions leave implicit that arguments, priorities, rules and rule names belong to some given argumentation theory with logical language $\mathcal{L}$. In our examples $\mathcal{L}$ consists of propositional literals but we stress that our game does not in any way depend on a particular logical language.

Definition 6. *A* move *is a tuple* $m = (i, A, pr, ns, del, t)$ *where:*

- $i \in \mathbb{N}$ *is the move identifier;*
- A *is an argument;*
- pr *is a set of priority statements about defeasible rules;*
- ns *is a set of ordered pairs* (r, l)*, where* $r \in \mathcal{R}_d$ *and* $l \in \mathcal{L}$*; (an assignment of names to defeasible rules, as part of the* n *function on* $\mathcal{R}_d$*)*[6]
- del *is a set of rules (to be deleted from the 'current' architecture specification)*
- $t \in \mathbb{N}$ *is the move target, that is, the move to which the move replies.*

[6] In the remainder we will for ease of notation represent the n function as a set of ordered pairs.

Below we will leave set elements of a move that are empty implicit. To indicate an element of a move m we will often write $i(m)$, $A(m)$ and so on.

Definition 7. *A* dialogue *is a finite sequence of moves $m_1, \ldots, m_n$ such that $t(m_1) = 0$ and for all j such that $1 < j \leq n$ it holds that $i(m_j) = j$ and $t(M_j)$ is some x such that $1 \leq x < j$.*

Below d_n is shorthand for dialogue $m_1, \ldots, m_n$, where d_0 is the empty dialogue. We call $m_i \in d_n$ a *defender move* if i is odd and an *attacker move* otherwise.

Definition 8. *The* argumentation theory AT_i relative to a dialogue d_i *is defined as follows:*

1. *AT_0 is any argumentation theory describing a system architecture where $\mathcal{R}_s^0$ is closed under transposition and $\leq = \{r \approx r \mid r \in \mathcal{R}_d\}$.*
2. *AT_i for $i > 0$ is such that:*
   (a) $\mathcal{R}_s^i = (\mathcal{R}_s^{i-1} \setminus del) \cup Cl_{tr}((\texttt{StrictRules}(A(m_i)))$[7]
   (b) $\mathcal{R}_d^i = (\mathcal{R}_d^{i-1} \setminus del) \cup \texttt{DefRules}(A(m_i))$
   (c) $\leq^i = \leq^{i-1} \cup pr \cup \{r < r' \mid r, r' \in \mathcal{R}_d^i$ *and r has but r' does not have an empty antecedent*$\} \cup \{r \approx r \mid r \in \mathcal{R}_d^i\}$
   (d) $n^i = n^{i-1} \cup ns_i$.
3. $\mathcal{K}_n^i = \mathcal{K}_n^{i-1} \cup \texttt{Prem}(A(m_i))$

The 'current winner' of a dialogue can be defined by adapting [13, 14]'s notion of dialogical status of a move:

Definition 9.

- *Move m is* in *iff all replies to m are* out*;*
- *Move m is* out *if either it has a retracted rule or it has a reply that is* in*.*

Note that since the reply structure on the game moves induces a tree, the dialogical status of a move is always uniquely defined.

We now adapt [13, 14]'s notion of relevance as follows.

Definition 10. *A defender move m_i is* relevant *iff exactly one defender move m_j ($j \leq i$) such that $\texttt{Conc}(A(m_j) = SR$ is in. An attacker move m_i is relevant iff all defender moves m_j ($j \leq i$) such that $\texttt{Conc}(A(m_j) = SR$ are out.*

We next define when a move is legal in a dialogue.

Definition 11. *A dialogue $d = m_1, \ldots, m_n$ is* legal *iff for all $m_i \in d$ it holds that m_i is legal in $m_1, \ldots, m_{i-1}$ (or in the empty dialogue if $d = m_1$).*

A move m_i is legal *in dialogue d_{i-1} iff the following conditions are satisfied.*

1. *If m_i is a defender (attacker) move, then $t(m_i)$ is an attacker (defender) move.*
2. *m_i is relevant.*
3. *$pr(m_i)$ leaves $\leq_i$ a total preorder.*
4. *$ns(m)$ leaves n_i a (partial or total) function from $\mathcal{L}$ to $\mathcal{R}_d^i$.*

[7] $Cl_{tr}(S)$ yields for any set S of strict rules its closure under transposition as defined in [11].

5. m_1 *is such that*
   (a) $\texttt{Conc}(A(m_1)) = SR$*; and*
   (b) $\texttt{Prem}(A(m_1)) \subseteq \mathcal{K}^0$*; and*
   (c) $\texttt{StrictRules}(A(m_1)) \subseteq \mathcal{R}_s^0$*; and*
   (d) $\texttt{DefRules}(A(m_1)) \subseteq \mathcal{R}_d^0 \cup \{\Rightarrow \varphi \mid \varphi$ *is an antecedent of a rule in* $\mathcal{R}_s^0$ *or* $\mathcal{R}_d^0\}$.
6. *If* $i > 1$ *and* m_i *is an attacker move, then*
   (a) $A(m_i)$ *defeats* $A(t(m_i))$ *on the basis of* AT_i*;*
   (b) $del(m_i) = \emptyset$.
7. *If* $i > 1$ *and* m_i *is a defender move, then*
   (a) $A(m_i)$ *has a subargument that strictly defeats* $A(t(m_i))$ *on the basis of* AT_i*; and*
   (b) *If* $A(m_i)$ *does not itself strictly defeat* $A(t(m_i))$ *on the basis of* AT_i*, then* $\texttt{Conc}(A(m_i)) = SR$*;*
   (c) del *does not contain rules from arguments in attacker moves in* d_{i-1}*;*
   (d) *If* $t(m_i) \neq i - 1$ *then* $AT_i = AT_{i-1}$.

Condition 1 states that the players may not respond to their own moves. Condition 2 makes that a dialogue is focussed on what it is meant for, namely, the critical testing whether the design meets requirement SR. Conditions 3-4 are to ensure that the argumentation theory constructed during a dialougue is well-defined, while Condition 5 regulates how the defender can start the game with an argument for SR. Condition 6 says that an attacker move must defeat a defender move without deleting rules.

Condition 7a requires the defender to move an argument with a subargument that strictly defeats the target argument of the attacker. Defeat must here be strict, since the 'burden of proof' is on the defender to show that SR is satisfied. Note that since an argument is a subargument of its own, the defeating subargument of $A(m_i)$ may be $A(m_i)$ itself. Recall that Condition 2 in effect requires that after the defender's move exactly one argument for SR is justified on the basis of AT_i. If m_i does not delete any rules from AT_{i-1} then this argument will be the one that is 'reinstated' by $A(m_i)$'s strict defeat of $A(t(m_i))$, otherwise this argument will be $A(m_i)$ itself. These last observations were illustrated in the final part of Section 3. Condition 7b says that defeating defender arguments can be extended to an argument for SR. The definition of relevance implies that such an extension is only legal if the move does not make an old argument for SR *in*. Condition 7c forbids the defender from deleting rules from the attackers arguments. This requirement is reasonable since it is defender's responsibility to build and modify the design through his own moves; the attacker does not contribute to the design but only criticises it. Finally, condition 7d says that the defender must always reply to the last move of the attacker, except if the defender makes a move that leaves the argumentation theory unchanged. Such 'logical' backtracking moves must be allowed to ensure that a dialogue can be logically completed.

4.3 Correspondence result

Definition 12. *A dialogue* d_i *is* logically completed *if no legal moves* m_{i+1} *exist such that* $AT_i = AT_{i+1}$.

In a logically completed dialogue, all logically possible legal moves on the basis of the current argumentation theory have been made. This means that every allowed continuation of the dialogue would change the argumentation theory. We now want for any logically completed dialogue that, if an argument for SR is dialogically *in*, then it is also justified on the basis of the 'current' argumentation theory. We are not so much interested in formal termination criteria for dialogues, since we assume that the players, being in essence cooperative, will agree to terminate a dialogue at a sensible moment. We now prove that our game has this property. The practical value of this result is that, to agree with the underlying logic, we do not need to restart an entire logical argument game after each move (as we would have to do if, for example, the protocol checked after each move whether the current AT justifies SR). Note also that, since all dialogue moves must be relevant, a dialogue will only in exceptional cases not be logically completed. For this reason, the restriction of Theorem1 to logically completed dialogues is not a severe practical limitation.

Theorem 1. *Let d_i be any logically completed dialogue with a defender move m_i that is in and such that $\texttt{Conc}(A(m_i)) = SR$. Then $A(m_i)$ is justified on the basis of AT_i.*

Proof. The reply relations on the moves in d_i induce a dialogue tree with as root m_1. Let T_i be its subtree with root m_i. Since m_i is *in*, by Proposition 23 of [14] there exists a 'winning part' W_i of T_i in the sense of Definition 22 of [14], i.e., a subtree of T_i that for each set of defender sibblings in T_i contains one element and contains all its attacker replies from T_i, and such that all defender moves in W_i are *in* while all attacker moves in W_i are *out*. Now let G_i be the tree obtained from W_i by replacing each move m in W_i with $A(m)$ (below written as A_i). We need to show that G_i is a winning strategy for A_i in the argument game for grounded semantics on the basis of AT_i.

First, all arguments in G_i are constructible on the basis of AT_i: if not, then G_i contains a defender argument A_j with a deleted rule, but then the node m_j in W_i from which it is derived is *out*: contradiction.

Second, it must be shown that G_i contains the correct defeat relations. Note first that by Definition 11(3) for any d_j it holds that $\leq_j$ is a total preorder, so defeat relations are preserved under addition of rule preferences. Then by Definition 11(6a) each argument at even depth defeats its parent. Note next that by definition of relevance of moves and the fact that all defender moves in W_i are *in*, G_i contains exactly one argument for SR, namely, A_i. Then by Definition 11(7a) each argument at odd depth except A_i itself strictly defeats its parent.

Next, since d_i is logically completed and Definition 11 imposes no conditions on logically completing attacker moves other than that their arguments must defeat the argument of their target, all defeaters on the basis of AT_i of any defender argument in G_i are in G_i.

This suffices to show that G_i is a winning strategy for A_i on the basis of AT_i. It follows that A_i is in the grounded extension of AT_i and so is justified. □

5 Example

We illustrate the definition of the game with the example from Section 3.1. In listing a move we will leave its identifier i obvious from the index of m and we will specify

del only for defender moves. Moreover, we will leave ns obvious from the subscripts of the rules in the moved argument. In specifying AT_i we will, overloading notation, indicate defeasible rules with their names in $\mathcal{L}$, and in specifying $\leq^i$ we will only list the explicitly stated $<$ priorities and leave priorities between assumptions and other rules and priorities that are required to leave $\leq^i$ a total preorder implicit. We will also leave the transpositions of strict rules implicit. Figure 2 shows the state of the following dialogue after move M5.

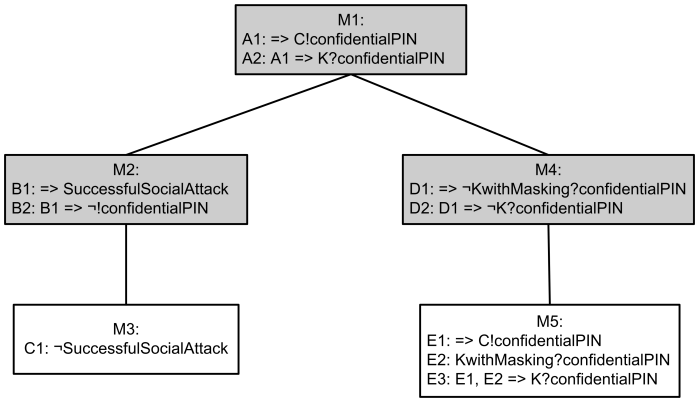

Fig. 2. State of the dialogue after move M5. White boxes are *in*while grey boxes are *out*.

– AT_0 is such that $\mathcal{K} = \mathcal{R}_s = \emptyset$ and $\mathcal{R}_d = \{conf\text{-}t5\}$.

– The defender starts with a move m_1 such that $A(m_1) =$

A_1: $\Rightarrow_{C-Keep-PIN-conf}$ `C!confidentialPIN`
A_2: $A_1 \Rightarrow_{conf-t5}$ `K?confidentialPIN`

Here $pr(m_1) = del(m_1) = \emptyset$ and $t(m_1) = 0$. As a result, AT_1 is such that $\mathcal{K} = \mathcal{R}_s = \emptyset$, $\mathcal{R}_d = \{conf\text{-}t5, C - Keep - PIN - conf\}$. Clearly, M_1 is currently *in* since it has no replies or retracted rules or premises.

– At m_2 the attacker attacks A_2 by directly attacking A_1 with $A(m_2) =$

B_1: $\Rightarrow_{Successful-attack-C-SE}$ `SuccesfulSocialAttack`
B_2: $B_1 \Rightarrow_{Attack-C-SE}$ $\neg$`C!confidentialPIN`

Here $t(m_2) = 1$. As a result, AT_2 is such that $\mathcal{K} = \mathcal{R}_s = \emptyset$, $\mathcal{R}_d = \{conf\text{-}t5, C-Keep-PIN-conf, Successful\text{-}attack\text{-}C\text{-}SE, Attack\text{-}C\text{-}SE\}$. On the basis of AT_2 we have that B_2 defeats A_2 on A_1, since the last defeasible rule of B_2 is $Attack$-C-SE, the last defeasible rule of A_1 is $C-Keep-PIN-conf$ and we have that $C-Keep-PIN-conf < Attack\text{-}C\text{-}SE$ since $C-Keep-PIN-conf$ is an assumption. Moreover, in the game we have that m_2 is *in* since it has no

replies, so m_1 is now *out* since it has a reply that is *in*.

– At m_3 the defender moves the following basic argument stating just a fact:

C_1: $\neg$`SuccesfulSocialAttack`

where $t(m_3) = 2$ and $pr(m_3) = del(m_3) = \emptyset$. Defender's move M_3 adds fact $No\text{-}successful\text{-}attack\text{-}C\text{-}SE$ to $\mathcal{K}$ and leaves the rest of AT_3 as in AT_2. On the basis of AT_3 we have that C_1 strictly defeats B_2 on B_1, since, unlike B_1, C_1 has no defeasible rules. We now have that m_3 is *in* since it has no replies, so m_2 is now *out* since it has a reply that is *in*: but then m_1 is *in* again since all its replies are *out* and it has no retracted premises or rules. Therefore, m_3 did not need to contain a new argument for SR, since $a(m_1)$ has conclusion SR.

– The attacker move m_4 now backtracks to m_1 (that is, $t(m_4) = 1$), this time attacking argument A_2 by directly attacking it on A_1 with

D_1: $\Rightarrow_{not-conf-t5-masking}$ $\neg$`KwithMasking`?`confidentialPIN`
D_2: $B_1 \Rightarrow_{not-K-keep-PIN-conf}$ $\neg$`K`?`confidentialPIN`

Moreover, the attacker states the priority $pr(m_4) = \{conf\text{-}t5 < not\text{-}K\text{-}keep\text{-}PIN\text{-}conf\}$. As a result, AT_4 is such that $\mathcal{K} = \{No\text{-}successful\text{-}attack\text{-}C\text{-}SE\}, \mathcal{R}_s = \emptyset, \mathcal{R}_d = \{conf\text{-}t5, C-Keep-PIN-conf, not\text{-}conf\text{-}t5\text{-}masking, Successful\text{-}attack\text{-}C\text{-}SE, not\text{-}K\text{-}keep\text{-}PIN\text{-}conf, Attack\text{-}C\text{-}SE\}$ and $conf\text{-}t5 < not\text{-}K\text{-}keep\text{-}PIN\text{-}conf$. On the basis of AT_4 we have that D_2 defeats A_2 on A_2, since the last defeasible rule of B_2 is $not\text{-}K\text{-}keep\text{-}PIN\text{-}conf$, the last defeasible rule of A_2 is $conf\text{-}t5$ and we have that $conf\text{-}t5 < not\text{-}K\text{-}keep\text{-}PIN\text{-}conf$. In the game we now have that m_4 is *in* so m_1 is now *out* since it has a reply that is *in*.

– At m_5 the defender moves the following argument in reply to m_4:

E_1: $\Rightarrow_{a1}$ `C!confidentialPIN`
E_2: `KwithMasking`?`confidentialPIN`
E_3: $E_1, E_2 \Rightarrow_{conf-t5-masking}$ `K`?`confidentialPIN`

Argument E_2 strictly defeats argument D_2 on $D-1$ since E_2 consists of a fact while D_1 consists of an assumption. Defender's move m_5 adds fact $masking$ to $\mathcal{K}$ and replaces rule $conf\text{-}t5$ in $\mathcal{R}_d$ with $conf\text{-}t5\text{-}masking$. This is effected by making $del(m_5) = \{conf\text{-}t5\}$. So AT_5 is such that $\mathcal{K} = \{No\text{-}successful\text{-}attack\text{-}C\text{-}SE, masking\}, \mathcal{R}_s = \emptyset$, $\mathcal{R}_d = \{conf\text{-}t5\text{-}masking, C-Keep-PIN-conf, not\text{-}conf\text{-}t5\text{-}masking, Successful\text{-}attack\text{-}C\text{-}SE, not\text{-}K\text{-}keep\text{-}PIN\text{-}conf, Attack\text{-}C\text{-}SE\}$, and $conf\text{-}t5 < not\text{-}K\text{-}keep\text{-}PIN\text{-}conf$. On the basis of AT_5 we have that argument E_2 strictly defeats argument D_2 on $D-1$ since E_2 consists of a fact while D_1 consists of an assumption.
Note that m_5 contains a new argument for SR, since the old argument A_2 is not constructible on the basis of AT_5. At this stage M_5 is clearly *in* so M_4 is now *out* since it has a reply that is *in*. However, M_1 remains out for the remainder of the game, since it has a retracted rule.

To illustrate Theorem 1, suppose the attacker makes no new move so that the game terminates. On the basis of AT_5 the attacker would have had no further legal move, so the game is logically completed. Now some move with an argument for the SR is *in*, namely, move M_5 with argument E_3; moreover, E_3 is trivially justified on the basis of AT_5, since it has no defeaters. So the defender's 'winning part' consists of just m_5.

6 Conclusion

This paper shows that it is feasible to reconstruct the security risk assessment dialog of experts as a formal argumentation game in *ASPIC*$^+$. The game is dynamic in that the players can both add elements to and delete elements from the architecture specification. The game was shown to respect the underlying argumentation logic in that for any logically completed game 'won' by the defender, the security requirement is a justified conclusion from the architecture specification at that stage of the game.

The idea to formalize risk assessment in argumentation logic is not new. Two early papers have suggested the use of argumentation in medical risk assessment [12, 6]. These proposals are preliminary and specific to the medical domain. There is more recent work on the use of argumentation in firewall policy specification and analysis [2, 1]. These papers focus on the logical representation of arguments about whether firewall policies satisfy certain properties and do not focus on dynamic or dialogical aspects. The current paper was based on earlier attempts to use informal Toulmin-style arguments to support IT security risk assessment [5, 4]. Those attempts did not use ideas from defeasible logic dialog games.

This paper raises a number of questions that we will investigate in the near future. An important long-term goal of our research is to provide tool support for argumentation-based risk assessment, and for this it is needed to find informal but precise representations of a risk argumentation game that can be understood by security experts but have a formal grounding in defeasible logic and dialogue games. We therefore want to investigate samples of actual RA dialogues to identify common dialogue patterns that can be exploited by the support tool to give suggestions to the risk assessors. We will here in particular explore the similarity between argumentation trees and attack trees [9], which are a familiar representation and reasoning structure for risk assessors and therefore warrant some confidence that an argumentation-based RA support tool will be natural for them. A further topic for future research is to analyze the role of qualitative risk assessments made in practice, where uncertainty and impact of events are estimated on ordinal scales such as (low, medium, high). Finally, we want to investigate the lifting of our current assumption that rule priorities are uncontroversial. Although in our experience this assumption holds for a fair number of risk assessments, this may not be so in general. One way to deal with this is to replace the current version of *ASPIC*$^+$ with [10]'s version that allows for argumentation about the argument ordering.

Acknowledgements We thank the reviewers for their stimulating and useful comments. Roel Wieringa has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 318003 (TREsPASS). This publication reflects only the author's views and the Union is not liable for any use that may be made of the information contained herein.

Bibliography

[1] A. Applebaum, K. Levitt, J. Rowe, and S. Parsons. Arguing about firewall policy. In B. Verheij, S. Woltran, and S. Szeider, editors, *Computational Models of Argument. Proceedings of COMMA 2012*, pages 91–102. IOS Press, Amsterdam etc, 2012.

[2] A.K. Bandara, A.C. Kakas, E.C. Lupu, and A. Russo. Using argumentation logic for firewall policy specification and analysis. In *Proceedings of the 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, pages 185–196. Springer Verlag, 2006.

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

[4] V.N.L. Franqueira, T.T. Tun, Y. Yu, R. Wieringa, and B. Nuseibeh. Risk and argument: a risk-based argumentation method for practical security. In *Proceedings of the 19th IEEE International Requirements Engineering Conference*, pages 239–248, Trento, Italy, 2011.

[5] C. Haley, R. Laney, J. Moffett, and B. Nuseibeh. Security requirements engineering: A framework for representation and analysis. *IEEE Transactions on Software Engineering*, 34(1):133–153, 2008.

[6] P. Krause, J. Fox, and Ph. Judson. An argumentation-based approach to risk assessment. *IMA Journal of Mathematics Applied in Business & Industry*, 5:249–263, 1993/4.

[7] R.P. Loui. Process and policy: resource-bounded non-demonstrative reasoning. *Computational Intelligence*, 14:1–38, 1998.

[8] M.S. Lund, B. Solhaug, and K. Stølen. *Model-Driven Risk Analysis. The CORAS Approach*. Springer-Verlag, Berlin Heidelberg, 2011.

[9] S. Mauw and M. Oostdijk. Foundations of attack trees. In D. Won and S. Kim, editors, *Information Security and Cryptology - ICISC 2005*, volume 3935 of *Lecture Notes in Computer Science*, pages 186–198. Springer Berlin Heidelberg, 2006.

[10] S. Modgil and H. Prakken. Reasoning about preferences in structured extended argumentation frameworks. In P. Baroni, F. Cerutti, M. Giacomin, and G.R. Simari, editors, *Computational Models of Argument. Proceedings of COMMA 2010*, pages 347–358. IOS Press, Amsterdam etc, 2010.

[11] S. Modgil and H. Prakken. A general account of argumentation with preferences. *Artificial Intelligence*, 195:361–397, 2013.

[12] S. Parsons, J. Fox, and A. Coulson. Argumentation and risk assessment. In *Proceedings of the AAAI Spring Symposium on Predictive Toxicology*, 1999.

[13] H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 127:187–219, 2001.

[14] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15:1009–1040, 2005.

[15] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.