Credulous and Sceptical Argument Games for Preferred Semantics

Gerard A.W. Vreeswijk and Henry Prakken

Institute of Information and Computing Sciences
Utrecht University, The Netherlands
{gv,henry}@cs.uu.nl

Abstract. This paper presents dialectical proof theories for Dung's preferred semantics of defeasible argumentation. The proof theories have the form of argument games for testing membership of *some* (credulous reasoning) or *all* preferred extensions (sceptical reasoning). The credulous proof theory is for the general case, while the sceptical version is for the case where preferred semantics coincides with stable semantics. The development of these argument games is especially motivated by applications of argumentation in automated negotiation, mediation of collective discussion and decision making, and intelligent tutoring.

1 Introduction

An important approach to the study of nonmonotonic reasoning is that of logics for defeasible argumentation (for an overview see [25]). Within this approach, a unifying perspective is provided by the work of [9] and [4] (below called the 'BDKT framework'). It takes as input a set of arguments ordered by a binary relation of 'attack', and it produces as output one or more 'argument extensions', which are maximal (in some sense) sets of arguments that survive the competition between all input arguments. A definition of argument extensions can be regarded as an *argument-based semantics* for defeasible reasoning. BDKT have developed various alternative such semantics, and investigated their properties and interrelations. They have also shown how many nonmonotonic logics can be recast in their framework. Thus their framework serves as a unifying framework not only for defeasible argumentation but also for nonmonotonic reasoning in general.

The BDKT framework exists in two versions. The version of [9] completely abstracts from the internal structure of arguments and the nature of the attack relation, while the version of [4] is more concrete. It regards arguments as sets of assumptions that can be added to a theory formulated in a monotonic logic in order to derive defeasible conclusions, and it defines attack in terms of a notion of contrariness of assumptions.

Besides a definition of argument-extensions, it is also important to have a test for extension membership of individual arguments, i.e., to have a *proof theory* for the semantics. A natural (though not the only) form of such proof theories is the dialectical form of an *argument game* between a defender and challenger of an argument [18, 29, 8, 5, 26, 24, 14]. The defender starts with an argument to be tested, after which each player must attack the other player's arguments with a counterargument of sufficient strength. The initial argument is provable if its defender has a winning strategy, i.e., if he can make the challenger run out of moves in whatever way she attacks. The precise rules of the argument game depend on the semantics which the proof theory is meant to capture.

For [4]'s assumption-based version dialectical proof theories have been studied by [15]. However, for [9]'s abstract version only the so-called 'grounded (sceptical) semantics' has been recast in dialectical style, viz. by [8]. Grounded semantics is sceptical in the sense that it always induces a unique extension of admissible arguments: in case of an irresolvable conflict between two arguments, it leaves both arguments out of the extension. For the other semantics of [9], which in case of irresolvable conflicts all induce multiple extensions, dialectical forms must still be developed. This paper contributes to this development: it presents a dialectical argument game for perhaps the most important multiple-extension semantics of [9], so-called preferred semantics. In fact, we shall present two results: a proof theory for membership of *some* preferred extension (credulous reasoning) and the same for membership of *all* preferred extensions (*sceptical* reasoning, although only for the case where preferred semantics coincides with stable semantics).

It should be motivated why proof theories for the most abstract version of the BDKT framework are important besides their counterparts for the assumption-based version. Kakas & Toni's work is very relevant when arguments can be cast in assumption-based form. In many applications this is possible, but in other applications this is different. For instance, argumentation has been used as a component of negotiation protocols, where arguments for an offer should persuade the other party to accept the offer [16, 20]. Argumentation is also part of some recent formal models and computer systems for dispute mediation [10, 11, 6], and it has been used in computer programs for intelligent tutoring: for instance, in a system (Belvedere) that teaches scientific reasoning [27] and in systems that teach argumentation skills to law students, e.g. [1]'s CATO system and [28]'s ARGUE system. Now in many applications of these types, arguments have a structure that cannot be naturally cast in assumption-based form. For instance, they can be linked pieces of unstructured natural-language text (cf. Belvedere or Gordon's ZENO system), or they consist of analogical uses of precedents, such as CATO's arguments. It is especially for such applications that proof theories for [9]'s abstract framework are relevant.

It should also be motivated why a proof-theory for preferred semantics is important despite the pessimistic results on computational complexity recorded by [7]. To start with, these pessimistic results concern worst-case scenarios, and cases might be identified where computation of preferred semantics is still feasible. Moreover, as demonstrated by e.g. [21, 18], logics for defeasible argumentation provide a suitable basis for resource-bounded reasoning: dialogues corresponding to such logics can be interrupted at any time such that the intermediate outcome is still meaningful. Finally, there is a possible use of argument-based proof theories which does not suffer from the computational complexity, viz. in automated mediation and tutoring. In, for instance, mediation systems for negotiation or collective decision making, and also in systems for intelligent tutoring, the search for arguments and counterarguments is not performed by the computer, but by the users of the system, who input their arguments into the system during a discussion. In such applications the argument-based proof theory can be used as a protocol for dispute: it checks whether the users' moves are legal, and it determines given only the arguments constructed by the users, which of the participants in a dispute is winning. (See e.g. [23] for a logical study of this use of dialectical proof theories).

Finally, we must motivate why argument-game versions are important besides other argument-based proof theories, such as [21]'s proof theory for his system, which is based on preferred semantics. This has to do with applications in fields like mediation and tutoring. In these fields, argumentation has been used as a component of several computational dialogue systems based on speech acts, such as models of legal procedure, [10, 13, 3, 17], discourse generation systems [12], multi-agent negotiation systems [20, 2], and intelligent tutoring [19]. In our opinion, the dialectical form of an argument game is ideally suited for embedding in such dialogue systems (see [22] for a formal study of such embeddings).

The structure of this paper is as follows. In Section 2 we provide an overview of the basics of the BDKT framework. In Section 3 we discuss with the help of examples which features our argument games should have. Then we define the credulous argument game in Section 4 and the sceptical game in Section 5, after which we discuss some limitations in Section 6.

2 Definitions and known results

In this section we review the basics of the BDKT framework, as far as needed for present purposes. The input of the system is a set of arguments ordered by an attack relation.

Definition 1. *(Argument system [9]). An* argument system $\mathcal{A}$ *is a pair*

$$\mathcal{A} = \langle X, \leftarrow \rangle, \tag{1}$$

where X is a set of arguments, and $\leftarrow$ is a relation between pairs of arguments in X. The expression $a \leftarrow b$ is pronounced "a is attacked by b," "b is an attacker of a," or "b is a counterargument of a".

Example 1. The pair $\mathcal{A} = \langle X, \leftarrow \rangle$ with arguments

$$X = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, p, q\}$$

and $\leftarrow$ as indicated in Figure 1 is an (abstract) example of an argument system. It accommodates a number of interesting cases and anomalies, and will therefore be used as a running example throughout this paper.

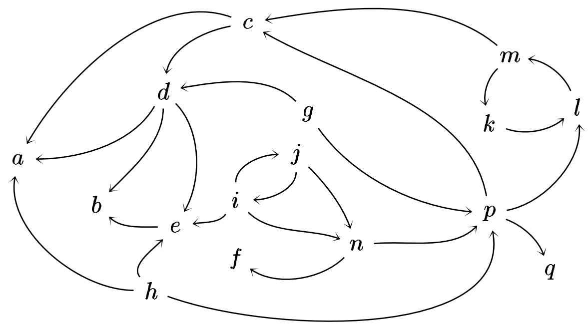


Fig. 1. Attack relations in the running example.

In practical applications it is necessary to further specify the internal structure of the arguments and the relation $\leftarrow$. See e.g. [30]. However, for the purpose of this paper it not necessary to do so; at present it suffices to know that there are arguments, and that some arguments attack other arguments.

The output of the system is one or more argument extensions, which are sets of arguments that represent a maximally defendable point of view. The different semantics of the BDKT framework define different senses of 'maximally defendable'. We list the definitions of two of them, stable and preferred semantics.

1. An argument a is *attacked* by a set of arguments B if B contains an attacker of a. (Not all members of B need attack a.)
2. An argument a is *acceptable* with respect to a set of arguments C, if every attacker of a is attacked by a member of C: for example, if $a \leftarrow b$ then $b \leftarrow c$ for some $c \in C$. In that case we say that c defends a, and also that C defends a.
3. A set S of arguments is *conflict-free* if no argument in S attacks an argument in S.
4. A conflict-free set S of arguments is *admissible* if each argument in S is acceptable with respect to S.
5. A set of arguments is a *preferred extension* if it is a $\subseteq$-maximal admissible set.
6. A conflict-free set of arguments is a *stable extension* if it attacks every argument outside it.

The following results of [9] will be used in the present paper.

Known results. *(from [9])*

1. *Each admissible set is contained in a $\subseteq$-maximally admissible set*
2. *Every stable extension is preferred.*
3. *Not every preferred extension is stable.*
4. *Stable extensions do not always exist; preferred extensions always exist.*
5. *Stable and preferred extensions are generally not unique.*

3 The basic ideas illustrated

In this section we discuss with the help of examples which features our argument games should have.

Our game for testing membership of *some* extension is based on the following idea. By definition, a preferred extension is a $\subseteq$-maximal admissible set. It is known that each admissible set is contained in a maximal admissible set, so the procedure comes down to trying to construct an admissible set 'around' the argument in question. If this succeeds we know that the admissible set, and hence the argument in question, is contained in a preferred extension.

Suppose now we wish to investigate whether a is preferred, i.e., belongs to a preferred extension. We know that it suffices to show that the argument in question is admissible. The idea is to start with $S = \{a\}$, which most likely is not admissible. (Because S is small, and small sets are usually conflict-free but not admissible.) So other arguments must be found (or constructed) in order to complete S into an admissible set.

Procedure. *(Constructing an admissible set). Let a be an argument for which we try to construct an admissible set. This task can best be divided in two subtasks:*

***Task 1:** construction.* *Let us suppose this task is performed by person **PRO**, who assumes a constructive role by trying to show that a is contained in an admissible set. To this end, **PRO** examines if there are arguments that attack his arguments constructed thus far. If there is such an argument, **PRO** tries to attack it by trying to construct an argument that attacks the original attacker (acceptability). If **PRO** has found such an argument, it must be consistent with his previous arguments (conflict-freeness).*

***PRO**'s role is purely defensive: his goal is to incorporate defenders against attacks constructed thus far — not to extend his collection of arguments per sé. To the contrary, in fact: **PRO**'s goal is to keep his collection of arguments as small as possible, because **PRO** is more vulnerable if he (or she)*[1] *has more arguments to defend.*

***Task 2:** criticism.* *This task is performed by person **CON**, who assumes a critical role by trying to find counterarguments to arguments advanced by **PRO**. In a way, **CON**'s aim is to 'make **PRO** talk' in the sense that **PRO** is more vulnerable if he has more arguments to defend.*

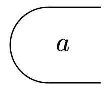
*The procedure formulated here is not necessarily adversarial: one way to look at it is to say that **CON*** helps ***PRO** by attending him to arguments that might invalidate **PRO**'s collection of admissible arguments.*

Example 2. (Straight failure). Consider the argument system that was presented at the beginning of this paper. Suppose **PRO**'s task is to show that a is preferred.

[1] From here on we will use the generic masculine form, intending no bias.

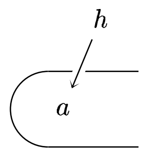
Since preferred extensions are maximally admissible sets it suffices for **PRO** to show that a is admissible, i.e., that a is contained in an admissible set.

The first action of **PRO** is simply putting forward a:



If a can't be criticized, i.e., if there are no attackers, then $S = \{a\}$ is admissible, and **PRO** succeeds. However, since $a \leftarrow h$,

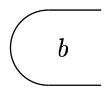
CON forwards h:



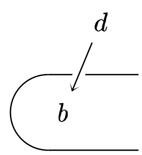
Now it is up to **PRO** to defend a by finding arguments against h. There are no such arguments, so that **PRO** fails to construct an admissible set 'around' a. So a is not admissible, hence not preferred.

Example 3. (Straight success). Suppose that **PRO** wants to show that b is admissible.

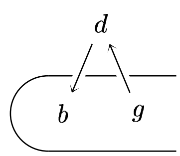
The first action of **PRO** is putting forward b:



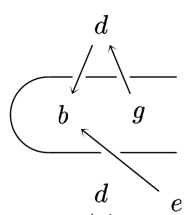
CON attacks b with d:



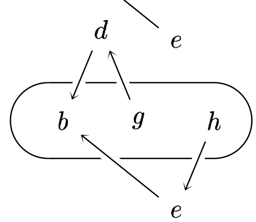
PRO defends this attack with g:



Since **CON**'s attack on b with d has failed, **CON** returns to b and attacks it again, this time with e:

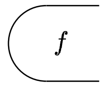


PRO defends b again, this time with h. Since **CON** is unable to find other argument against b, g or h, **PRO** may now close S:

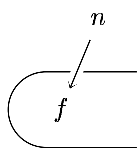


Example 4. (Even loop success). Suppose that **PRO** wants to show that f is admissible.

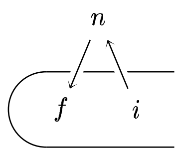
The first action of **PRO** is putting forward f:



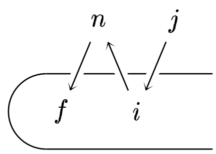
CON attacks f with n:



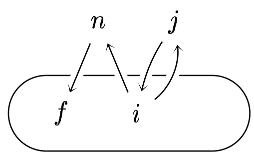
PRO defends this attack with i:



CON attacks i with j:



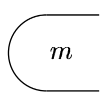
PRO defends i with i itself (so that i is self-defending). **CON** is unable to put forward other arguments that attack f or i so that **PRO** closes S:



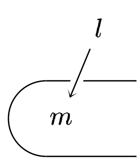
This example shows that **PRO** must be allowed to repeat his arguments, while **CON** must be forbidden to repeat **CON**'s arguments (at least in the same 'line of dispute'; see further below)

Example 5. (Odd loop failure). Suppose that **PRO** wants to show that m is admissible.

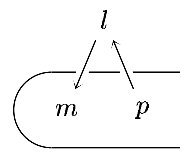
The first action of **PRO** is putting forward m:



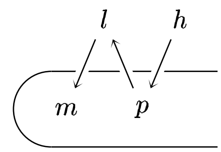
CON attacks m with l:



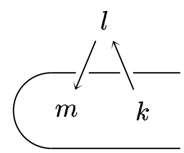
PRO defends this attack with p:



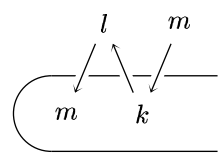
CON attacks p with h:



PRO backtracks and removes p from S. He then tries to defend l with k instead:

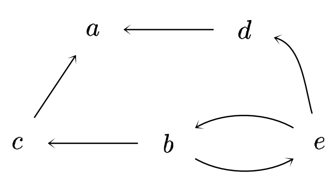


CON attacks k with m (and, as a bonus, introduces an inconsistency in S):



PRO has no other arguments in response to l and m, so that he is unable to close S into an admissible set. So m is not contained in an admissible set. Note that we cannot allow **PRO** to reply to m with l, since otherwise the set that **PRO** is constructing 'around' m is not conflict-free, hence not admissible. So we must forbid **PRO** to repeat **CON**'s moves. On the other hand, this example also shows that **CON** should be allowed to repeat **PRO**'s moves, since such a repetition reveals a conflict in **PRO**'s position.

Example 6. (The need for backtracking). Consider next an argument system with five arguments a, b, c, d and e and attack relations as shown in the graph.



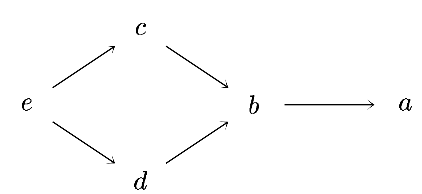
This example shows that we must allow **CON** to backtrack. Suppose **PRO** starts with a, **CON** attacks a with d, and **PRO** defends a with e. If **CON** now attacks e with b, **PRO** can defend e by repeating e itself. However, **CON** can backtrack to a, this time attacking it with c, after which **PRO**'s only move is defending a with b. Then **CON** can repeat **PRO**'s move e, revealing that **PRO**'s position is not conflict-free.

Repetition Let us summarise our observations about repetition of moves. If **PRO** can defend an argument by using one of his previous arguments that is not backtracked, then should **PRO** do that? Further, does it make sense for **PRO** to repeat arguments advanced by **CON**? The same questions can be asked for repetitions by **CON**.

i. It makes sense for **PRO** to repeat itself (if possible), because **CON** might fail to find or produce a new attacker against **PRO**'s repeated argument. If so, then **PRO**'s repetition closes a cycle of even length, of which **PRO**'s arguments are admissible.
ii. **CON** should repeat **PRO** (if possible), because it would show that **PRO**'s collection of arguments is not conflict-free.
iii. **PRO** should not repeat **CON**, because it would introduce a conflict into **PRO**'s own collection of arguments.
iv. It does not make sense if **CON** repeats itself, because **PRO** has already shown to have adequate defense for **CON**'s previous arguments.

Finally, we show that **CON** should be allowed to repeat **CON**'s arguments when they are from different 'lines' of a dispute. A *dispute line* is a dispute where each move replies to the immediately preceding move; i.e., in a dispute line no backtracking is allowed.

Example 7. (repetition from different lines)



Suppose **PRO** starts a dispute for a and **CON** attacks a with b. Then **PRO** has two alternative ways to defend a, viz. with c and with d, but **CON** must be allowed to reply to each of them with e.

4 The credulous argument game defined

We now turn to the formal definition of our argument games, starting with the credulous game. During a dispute a tree of dispute lines is constructed. This can be illustrated with the following format of disputes, taken from [29].

Example 2:

1. $|$ **PRO** : a
2. $||$ **CON** : $h\ddagger$

Example 3:

1. $|$ **PRO** : b
2. $||$ **CON** : d
3. $|||$ **PRO** : $g\dagger$
4. $||$ **CON** : e
5. $|||$ **PRO** : $h\ddagger$

Example 4:

1. $|$ **PRO** : f
2. $||$ **CON** : n
3. $|||$ **PRO** : i
4. $||||$ **CON** : j
5. $|||||$ **PRO** : i (iv)

Example 5:

1. $|$ **PRO** : m
2. $||$ **CON** : l
3. $|||$ **PRO** : p
4. $||||$ **CON** : $h\dagger$
5. $|||$ **PRO** : k
6. $||||$ **CON** : m (iii)

The vertical bars "|||" indicate the *level* of the dispute, i.e., the depth of the tree. E.g., in Ex. 3, **PRO** responded to a response of **CON** (level 3), after which **CON** backtracks (level 2) to try a new argument against b.

The "†"-symbol means that the player cannot respond to the last argument of the other player, while the "‡"-symbol means that the player is unable to respond to all arguments of the other player presented thus far. A number in the range (i-iv) means that a next move of the player would make no sense on the basis of the corresponding repetition guideline.

Rules and correspondence To establish a precise correspondence between disputes and preferred extensions, it is necessary to make the terminology more precise and to define the rules under which a dispute is conducted.

- A *move* is simply an argument (if the first move) or else an argument attacking one of the previous arguments of the other player.
- Both parties can *backtrack*.
- An *eo ipso* (meaning: "you said it yourself") is a move that uses a previous non-backtracked argument of the other player.
- A *block* is a move that places the other player in a position in which he cannot move.
- A *two-party immediate response dispute* (TPI-dispute) is a dispute in which both parties are allowed to repeat **PRO**, in which **PRO** is not allowed to repeat **CON**, and in which **CON** is allowed to repeat **CON** iff the second use is in a different line of the dispute. **CON** wins if he does an *eo ipso* or blocks **PRO**. Otherwise, **PRO** wins.

A main argument of a TPI-dispute is *defended* if the dispute is won by **PRO**.

Proposition 1. (Soundness and completeness of the credulous game). *An argument is in some preferred extension iff it can be defended in every* TPI-*dispute.*

Proof. By definition of preferred extensions it suffices to show that an argument is admissible iff it can be defended in every dispute.

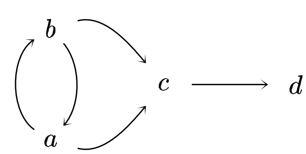
First suppose that a can be defended in every dispute. This includes disputes in which **CON** has opposed optimally. Let us consider such a dispute. Let A be the arguments that **PRO** used to defend a. (in particular $a \in A$.) If A is not conflict-free then $a_i \leftarrow a_j$ for some $a_i, a_j \in A$, and **CON** would have done an *eo ipso*, which is not the case. If A is not admissible, then $a_i \leftarrow b$ for some $a_i \in A$ while $b \not\leftarrow A$. In that case, **CON** would have used b as a winning argument, which is also not the case. Hence A is admissible.

Conversely, suppose that $a \in A$ with A admissible. Now **PRO** can win every dispute by starting with a, and replying with arguments from A only. (**PRO** can do this, because all arguments in A are acceptable wrt A.) As long as **PRO** picks his arguments from A, **CON** cannot win by *eo ipso*, because A is conflict-free. So a can be defended in dispute.

5 The sceptical argument game defined

Above, **PRO** tries to show that the main argument is contained in a preferred set. This is known as *credulous* reasoning. If **PRO** wishes to verify whether the main argument is contained in *all* preferred sets, then **PRO** does *sceptical* reasoning. Before defining an argument game for this kind of reasoning, we must first explain why for sceptical reasoning it is relevant to study preferred semantics besides [9]'s grounded semantics, which is also meant for sceptical reasoning. The reason is that grounded semantics is too weak to capture certain types of sceptical conclusions.

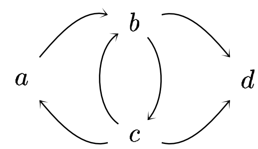
Example 8. (Floating arguments.) Consider the arguments a, b, c and d with the attack relations as shown in the picture.



Since no argument is unattacked, the grounded extension is empty. However, this example has two preferred extensions, $\{a, d\}$ and $\{b, d\}$, and both of them contain d.

Next we illustrate that there are cases where an argument system has a unique preferred extension but not all of its elements are contained in the grounded extension.

Example 9. Consider four arguments a, b, c and d with the attack relations as shown in the picture.



The unique preferred extension is $\{c\}$, so c is sceptically preferred, but the grounded extension is empty, since none of the arguments are unattacked.

We now define the sceptical argument game. A result for sceptical reasoning can be obtained by observing that a dispute is symmetric, since **CON** also may be given the task to construct an admissible set, viz. for the attackers he uses. If **CON** succeeds, he has shown that there exists at least one admissible set not including the main argument.

Proposition 2. (Soundness and completeness of the sceptical game). *In argument systems where each preferred extension is also stable, an argument is in all preferred extensions iff it can be defended in every* TPI*-dispute, and none of its attackers can be defended in every* TPI*-dispute.*

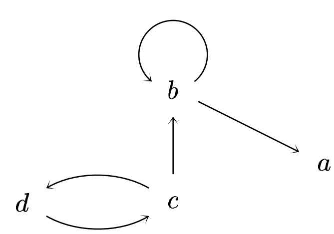
Proof. This result can be proven on the basis of the previous proposition, and by the fact that a stable extension attacks every argument outside it.

Consider any argument system where all preferred extensions are stable. For the only-if-part of the equivalence, consider any argument a that is in all preferred extensions. Then (by assumption that these extensions are also stable) all attackers of a are attacked by all such extensions, so by conflict-freeness of preferred extensions, none of these attackers is in any such extension. But then none of a's attackers is credulously provable.

For the if part, Let a be any argument that is credulously provable and such that none of a's attackers are credulously provable. Then none of these attackers is in any preferred extension, so (by assumption that these extensions are also stable) they are attacked by all such extensions. But then a is defended by all these extensions, so they all contain a.

The following example shows that this result does not hold in general.

Example 10. Consider:



a is contained in one preferred extension, viz. $E_1 = \{a, c\}$, but not in the other preferred extension, which is $E_2 = \{d\}$. Note that the self-attacking argument b prevents a from being a member of E_2 although b is not itself a member. The problem is that E_2 does not attack b so that a is not acceptable with respect to E_2. This situation cannot arise when all preferred extensions are stable, since then they attack all arguments outside them.

6 Discussion

The present paper has provided simple and intuitive argument games for both credulous and sceptical reasoning in preferred semantics. However, there are still some limitations and drawbacks.

A limitation is, of course, that the sceptical game is not sound and complete in general. A first drawback is the fact that the sceptical game actually consists of two parallel games, which is less elegant in applications in mediation and tutoring systems. In future research we hope to improve the games in both respects.

Another drawback is that in some cases proofs are infinite. This is obvious when an argument has an infinite number of attackers, but even otherwise some proofs are infinite, as in the following example.

Example 11. (Infinite attack chain.) Consider an infinite chain of arguments $a_1, \ldots, a_n, \ldots$ such that a_1 is attacked by a_2, a_2 is attacked by a_3, and so on.

$$a_1 \longleftarrow a_2 \longleftarrow a_3 \longleftarrow a_4 \longleftarrow a_5 \longleftarrow \cdots$$

PRO can win a game for a_1 (or for any other argument) since **CON** is never able to move a block, but **PRO** neither has a blocking move available.

Nevertheless, it is easy to verify that with a finite set of arguments all proofs are finite.

References

[1] V. Aleven and K.D. Ashley. Evaluating a learning environment for case-based argumentation skills. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, pages 170–179, New York, 1997. ACM Press.

[2] L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In *Proceedings of the Fourth International Conference on MultiAgent Systems*, Boston, MA, 2000.

[3] T.J.M. Bench-Capon. Specification and implementation of Toulmin dialogue game. In *Legal Knowledge-Based Systems. JURIX: The Eleventh Conference*, pages 5–19, Nijmegen, 1998. Gerard Noodt Instituut.

[4] A. Bondarenko, P.M. Dung, R.A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1-2):63–101, 1997.

[5] G. Brewka. A reconstruction of Rescher's theory of formal disputation based on default logic. In A. Cohn, editor, *Proceedings of the 11th European Conference on Artificial Intelligence*, pages 366–370. John Wiley & Sons, Ltd., 1994.

[6] G. Brewka. Dynamic argument systems: a formal model of argumentation processes based on situation calculus. *Journal of Logic and Computation*, 2000. To appear.

[7] Y. Dimopoulos, B. Nebel, and F. Toni. Preferred arguments are harder to compute than stable extensions. In T. Dean, editor, *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI99)*, pages 36–41. Morgan Kaufmann, 1999.

[8] P.M. Dung. Logic programming as dialog game. Unpublished paper, Division of Computer Science, Asian Institute of Technology, Bangkok, 1994.

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

[10] T.F. Gordon. *The Pleadings Game: An Artificial Intelligence Model of Procedural Justice.* Kluwer, Dordrecht, 1995. Revised edition of the author's Ph.D. thesis (same title), Technische Hochschule, Darmstadt, 1993.

[11] T.F. Gordon, N. Karaçapilidis, H. Voss, and A. Zauke. Computer-mediated cooperative spatial planning. In H. Timmermans, editor, *Decision Support Systems in Urban Planning*, pages 299–309. E & FN SPON Publishers, London, 1997.

[12] F. Grasso, A. Cawsey, and R. Jones. Dialectical argumentation to solve conflicts in advice giving: a case study in the promotion of healthy nutrition. *International Journal of Human-Computer Studies*, 2000. To appear.

[13] J.C. Hage, R.E. Leenes, and A.R. Lodder. Hard cases: a procedural approach. *Artificial Intelligence and Law*, 2:113–166, 1994.

[14] H. Jakobovits and D. Vermeir. Dialectic semantics for argumentation frameworks. In *Proceedings of the Seventh International Conference on Artificial Intelligence and Law*, pages 53–62, New York, 1999. ACM Press.

[15] A.C. Kakas and F. Toni. Computing argumentation in logic programming. *Journal of Logic and Computation*, 9:515–562, 1999.

[16] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104:1–69, 1998.

[17] A.R. Lodder. *DiaLaw: On Legal Justification and Dialogical Models of Argumentation.* Kluwer, Dordrecht, 1999.

[18] R.P. Loui. Process and policy: Resource-bounded nondemonstrative reasoning. *Computational Intelligence*, 14(1):1–38, 1998.

[19] N. Maudet and D. Moore. Dialogue games for computer-supported collaborative argumentation. In *Proceedings of the Workshop on Computer-Supported Collaborative Argumentation for Learning Communities*, Stanford, 1999.

[20] S. Parsons, C. Sierra, and N.R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8:261–292, 1998.

[21] J.L. Pollock. *Cognitive Carpentry. A Blueprint for How to Build a Person.* MIT Press, Cambridge, MA, 1995.

[22] H. Prakken. On dialogue systems with speech acts, arguments, and counterarguments. 2000. These proceedings.

[23] H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 2000. To appear in special issue on New Perspectives in Dialogical Logics.

[24] H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics*, 7:25–75, 1997.

[25] H. Prakken and G.A.W. Vreeswijk. Logical systems for defeasible argumentation. To appear in D. Gabbay (ed.), Handbook of Philosophical Logic, 2nd edition. Kluwer Academic Publishers, Dordrecht etc., 2000.

[26] G.R. Simari, C.I. Chesñevar, and A.J. Garcia. The role of dialectics in defeasible argumentation. In *Proceedings of the XIV International Conference of the Chilean Computer Science Society*, 1994.

[27] D. Suthers, A. Weiner, J. Connelly, and M. Paolucci. Belvedere: engaging students in critical discussion of science and public policy issues. In *Proceedings of the Seventh World Conference on Artificial Intelligence in Education*, pages 266–273, 1995.

[28] H.B. Verheij. Automated argument assistance for lawyers. In *Proc. of the Seventh Int. Conf. on Artificial Intelligence and Law*, pages 43–52, New York, 1999. ACM.

[29] G.A.W. Vreeswijk. Defeasible dialectics: A controversy-oriented approach towards defeasible argumentation. *The Journal of Logic and Computation*, 3(3):3–27, 1993.

[30] G.A.W. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90:225–279, 1997.