Procedural arguments

PROCEDURAL ARGUMENTS

Arno R. Lodder
Universiteit Maastricht - Faculty of Law - Department of Metajuridica
P.O. Box 616 - 6200 MD Maastricht
arno.lodder@metajur.unimaas.nl
http://www.metajur.unimaas.nl/~arno

Although many authors claim that argumentation is a process, in most models the procedural side of argumentation is restricted to the definition of a procedural framework that is able to compare arguments. The process of building arguments is neglected, while attack of arguments is restricted.

In procedural models elements of the language with a special structure are called arguments, in this paper: *structural* arguments. Besides analyzing structural arguments (how they can be attacked, justified, built, and whether they necessarily have underlying rules), the phenomenon of *procedural* arguments is introduced.

1. Introduction

If the product of argumentation is studied, general structures of support or justification between sets of premises and conclusions are defined. The product of argumentation is static, just like a 'check mate' position in chess. The procedure of argumentation, on the other hand, is dynamic, just as a strategy path that leads to check mate. For a chess player it is interesting to know under what circumstances a position is check mate. Even more interesting is what procedure must be followed to check mate the opponent.

Nowadays it is generally accepted in the AI and Law community that the study of legal argumentation should not only concern the product of argumentation, but that the process or procedure is an important subject too. Only five years ago Hage, Span and Lodder [1992] could open their contribution to the JURIX-conference with the following sentence:

> "Although most legal reasoning takes place in the context of legal disputes in which two parties argue their positions in a dialogue with the other party, formal models of legal reasoning usually resemble monological proofs."

Things have radically changed since then. At this moment our characterization of formal models of legal reasoning can hardly be defended anymore. The number of papers that reported on formal models of legal reasoning taking a procedural, i.e. a dialogical approach is impressive [e.g., Gordon 1993, Loui *et al.* 1993, Hage *et al.* 1994, Lodder and Herczog 1995, Farley and Freeman 1995, Prakken and Sartor 1996, Nitta and Shibasaki 1997].

In argumentative dialogs arguments can appear in different ways. In this paper a distinction is made between structural and procedural arguments. The former are arguments due to their specific structure, the latter due to the procedure in which they are adduced. Most procedural models represent only an exchange of structural arguments. This paper analyses structural

arguments and argues that there is more to argumentation than structural arguments, namely procedural arguments.

The paper is structured as follows. First, the terminology as used in the paper, viz. statements, structural and procedural arguments is introduced. Then follows an analysis of structural arguments. The elements recognized in the present analysis of structural arguments are used to explain how these arguments can be supported, how they can be attacked, and how the elements can be used to model the process of building structural arguments. Subsequently it is argued that not necessarily all elements of structural arguments have to be present, and moreover, that structural arguments are not always necessary and desirable. This paves the way for the introduction of procedural arguments. After a characterization is given of the model DiaLaw [Lodder and Herczog 1995], that allows to model both procedural and structural arguments, related literature is briefly discussed.

2. Terminology

There is a difference between a statement and an argument. Suppose we have two statements: 1. Jeffrey is punishable, and; 2. Jeffrey has murdered someone else. An argument shows a relation between statements. Arguments indicate that a statement supports or justifies another statement. An example is the argument that Jeffrey is punishable, *because* he has murdered someone else. I call this type of argument a structural argument, because the reason it is an argument is its specific structure. Although an argument can be regarded as a special type of statement, namely one that relates two other statements, for reasons of clarity structural arguments are excluded if statements are mentioned.

Most models do have structural arguments. For instance, in the Pleadings Game [Gordon 1995] arguments are represented as `argument(A p)`, where `A` is a set of formulas that supports the formula `p`. In Reason-Based Logic [e.g. Verheij 1996, Hage 1997], as well as in DiaLaw [Lodder and Herczog 1995][1], there are basically two types of structural arguments. The first type is the reason: `reason(Cond, Concl)`

This formula says that `Cond` is a reason for `Concl`. The following formulas are examples of reasons:

```
reason(murderer(oj), punish(oj)),
reason(falsely_accused(oj), ~punish(oj)).
```

The fact that OJ is a murderer is a reason for punishing him (or a reason against not punishing him). The fact that OJ is falsely accused is a reason to not punish him (or a reason against not punishing him).

The second type of structural argument is the relation 'outweighs', saying that concerning some formula a set of reasons pro outweighs a set of reasons con.[2] The outweighing relation does not play an important role in the present paper.

In a procedure, also statements can become arguments. They are arguments not because of their structure, but because of the procedure in which they have been put forward. I call these type of arguments *procedural* arguments. A statement is a procedural argument, if it is used in a

---

[1] The reasons are slightly adapted version of the ones as presented in [Lodder and Herczog 1995]

[2] The formula `outweighs({Condpro`$_1$`, ..., Condpro`$_m$`}, {Condcon`$_1$`, ..., Condcon`$_n$`}, Concl) (m≥1,n≥0)`, reads as: The set `{Condpro`$_1$`, ..., Condpro`$_m$`}` originates from `reason(Condpro`$_i$`, Concl)`; the set `{Condcon`$_1$`, ..., Condcon`$_n$`}` originates from `reason(Condcon`$_1$`, ~Concl)`. The set of reasons pro must contain at least one term (`m≥1`), the set of reasons con may be empty (`n≥0`).

procedure to support or justify. Before procedural arguments are further explained in section 5, first structural arguments are analyzed.

3. Structural arguments

Suppose that Jeffrey has murdered someone. In that case an obvious structural argument is that Jeffrey is punishable, because he has murdered someone. This can be represented as DiaLaw's structural argument reason:

```
reason(murderer(jeffrey), punishable(jeffrey)).
```

Sometimes a structural argument is acceptable for anyone, but in cases it is not there must be an opportunity to either attack or question the argument. In the discussion of the possible attacks and defenses of arguments, as well in the discussion of the process of building arguments, different elements of a structural argument play a role. This paper distinguishes four elements of a structural argument (see for different, well-known analyses of arguments, e.g. Toulmin [1958] and Van Eemeren and Grootendorst [1987]):

1. the conclusion;
2. the condition;
3. the underlying rule in this specific case, or, the application of the rule;
4. the underlying rule in general, or, the validity of the rule.

The four elements can be easily recognized in a representation of the argument in First Order Predicate Logic[3]:

```
(4)  murderer(X) → punishable(X)
     ____________________________
(3)  murderer(jeffrey) → punishable(jeffrey)
(2)  murderer(jeffrey)
     ____________________________
(1)  therefore punishable(jeffrey)
```

The numbers refer to the different elements, so, the conclusion (1), the condition (2), the rule in this particular case, or the application of the rule (3), and the rule in general or the validity of the rule (4).

In sections 3.1-3.3 is discussed what role the four elements play in justifying an argument, attacking an argument, and building an argument, respectively.

3.1 Justifying structural arguments

A structural argument can be justified with all but the first element: the conclusion. Since the argument itself aims at justifying the conclusion, it is not strange that the conclusion cannot justify the argument.

[3] A similar scheme [Walton and Krabbe 1995, p. 180], more explicitly mentioning rules, is :

```
(4)     (Rule: n) Mx → Px
(2)     Mo
(3)     Rule n applies to the present case
        ____________________________
(1)     Therefore Po
```

We consider again the argument that Jeffrey is punishable, because he has murdered someone. There remain three elements that may justify the argument.

First, the condition of the argument, that is, the fact that Jeffrey is a murderer can be put forward. A discussion can start about whether Jeffrey really is a murderer.

Second, it can be claimed that the rule applies, or that the rule is true for this specific case.

Finally, the general rule that murderers are punishable can be adduced. A discussion can start about whether the rule is valid or the rule in general is true.

If the three ways to justify the argument have been used successfully, a firm basis for the justification of the argument is given. An example of how the elements may influence the justification of the argument in a dialog, is the way they are used in DiaLaw. Commitment to the condition of the argument and to the validity of the rule can never be used to force the opponent to accept the argument, i.e. the reason, but they can be used as means to convince the opponent of the fact that the rule applies. Namely, if the condition of the rule is satisfied, and the rule is valid, this is normally sufficient support to apply the rule. Commitment to the fact that the rule applies provides sufficient support for the reason, that is, the opponent is in that case forced to accept the reason.

3.2 Attacking structural arguments

All elements can play a role in the attack of structural arguments, so there are four different ways to attack. As an example, again the following argument is used:

```
reason(murderer(jeffrey), punishable(jeffrey)).
```

First, the conclusion of the argument can be denied:

```
~punishable(jeffrey).
```

An attacking argument supporting the denial of the conclusion is commonly referred to as a rebutter. An example of a rebutting argument is that Jeffrey is not punishable, because he is a diplomat. In DiaLaw this argument would be represented as:

```
reason(diplomat(jeffrey), ~punishable(jeffrey)).
```

Second, the condition of the argument can be denied:

```
~murderer(jeffrey).
```

This denial could be supported by the claim that it was not Jeffrey who was the murderer, but his twin brother Bert.

Third, the specific rule underlying the argument can be denied:

```
~applies(rule(murderer(jeffrey), punishable(jeffrey))).
```

The argument supporting this denial is usually referred to as an undercutter. For instance, the murderer is not punishable, because he acted in self-defense. In DiaLaw this argument would be represented as:

```
reason(self_defense(jeffrey),
      ~applies(rule(murderer(jeffrey), punishable(jeffrey))))
```

Finally, the general rule underlying the argument can be denied:

```
~valid(rule(murderer(Person), punishable(Person))).
```

An argument supporting this denial could be that the rule is not valid, since it was not announced properly. In DiaLaw this argument would be represented as:

```
reason(~announced_properly(rule(   murderer(Person),
```

```
                                    punishable(Person))),
~valid(rule(murderer(Person), punishable(Person)))
```

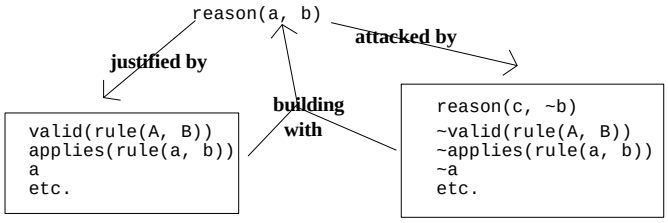
3.3 Building structural arguments

In case of attack and justification, an already existing argument is either attacked or justified. Instead of starting with the argument, the process can also be the other way around. So starting with the elements of an argument, in the end a structural argument is constructed.

It is not realistic to assume that always all necessary information for the construction of an argument is present at the beginning of a justification or attack. Only if it is clear what justifies a particular statement, a structural argument justifying the statement can be put forward. However, often a particular statement may seem justified, without it being clear why. For instance, a lawyer intuitively solves a case, and then tries to construct justifying reasons. This process of constructing justifying reasons can be modeled as a dialog in which the proponent puts forward statements, and the opponent indicates what statements he wants to be further justified. Adduced statements can slowly evolve into an indefeasible supporting argument.

Building the argument that Jeffrey is punishable comes down to the following. All elements of arguments can be used. The discussion starts with the conclusion (the intuitive solution to the case). Statements, that may turn out to correspond to conditions of the argument can be claimed and attacked. The validity of rules, that may turn out to lie under the argument can be claimed and attacked. Finally, the application of rules, that may turn out to lie under the argument can be claimed and attacked. In the end, that can be reached after one or more of the elements are adduced (and possibly attacked), the structural argument can be constructed.

The process that ends in the construction of a structural argument is opposite to the processes of justification and attack. In the latter processes the elements of an argument are used *after* the structural argument is adduced; in the former process the elements of an argument are used *before* the structural argument is adduced. The following picture shows the relation between the three processes.



4. Structural arguments without underlying rules

In DiaLaw the rules underlying reasons are implicit. In the Pleadings Game the backing of the rule, comparable to the validity of the rule in DiaLaw, always is part of the argument. The same is true for Reason-Based Logic. Each reason has a valid rule underlying it. In my opinion not all arguments (reasons) necessarily have underlying rules.

In order to make my point clear, I discuss three ways reasons and rules can be related. First, a reason can be based on the application of an already existing rule. Second, a reason can be based on the application of a new created rule. Finally, a reason can be based on no rule application at all.

4.1 Reasons based on an existing rule

An obvious example of an existing rule is one that is laid down in the statutes. But not only statutory rules, also rules taken from case law belong to the category of already existing rules. Take for example the legal rule 'if someone kills another person intentionally, then he is liable for second degree murder'. In case Oedipus killed his father intentionally, this fact is a justifying reason for the statement that Oedipus is liable for second degree murder. Although the rule is existing, terms in legal rules sometimes need to be interpreted, like in this case *intentionally*.

Suppose Oedipus *intentionally* shot three feet above his mother's head, and that accidentally this shot fatally wounded his father. Since in case law intentionally is also explained as 'knowingly and willingly accepting the ... chance that ...', it may be concluded here that Oedipus is liable for second degree murder. Since the interpretation is based on case law, the conclusion is justified on the basis of an existing rule. However, the first time intentionally was interpreted as above by the Dutch Supreme Court[4], the reason was not based on an existing rule.

4.2 Reasons based on a newly created rule

In the Dutch Road Traffic Act it is regulated that in case of an accident between a motorist and other non-motoring road users, e.g., pedestrians or bikers, the motorist has to pay for the damages, except if he acted in force majeure[5]. In the beginning of the nineties the Dutch Supreme Court decided that in case a child not older than fifteen is involved in the accident, the motorist always has to pay the damages, except if the child acted recklessly or intentionally. This narrowing of force-majeure can be considered as a rule, although it was not explicitly presented by the court as such.

In a later case the Dutch Supreme Court had to decide about an accident in which an elderly man was involved. The Dutch Supreme Court explicitly formulated a new rule: it is only fair that a motorist has to pay at least 50% of the damages in case he cannot demonstrate that he acted in force-majeure, and the mistake of the pedestrian or biker did not constitute intent or recklessness.

4.3 Reasons based on no rule at all

Finally, reasons can be put forward, based on no rule at all.

In the above-mentioned case in which a new rule was created, another decision dealt with the question whether the rule about young children should also be applied to elderly people. The Dutch Supreme Court decided that the rule was not applicable, based on the following reasons:

1. the arguments used to define the rule for young people are less striking for elderly people;

[4] HR 9 november 1954, NJ 1955,55.

[5] The concerning rule is laid down in section 185 Wegenverkeerswet 1994.

2. law knows no fixed age criterion for elderly people (for young people it does);
3. elderly people are less recognizable.

This collection of reasons constituted the decision of the Dutch Supreme Court. No rule was applied here, the decision was justified by these reasons alone. An objection may be that there is a rule after all, that can be created after the decision is made. First, this objection does not change the fact that the reasons were in the first place not based on a rule. Second, if reference is made to this case, probably the collection of reason will be mentioned, and not some –rather artificial- rule.

Reasons can be based on the application of (a new created) rule, or reasons can be based on no rule application at all. An interesting consequence of the fact that some reasons are not based on rule application, is that exceptions to rules also may be based on no rule at all. Does this mean that rules are useless? Not at all, rules are a good starting point, and often a justification can be based on the application of a rule alone. However, just as statements may be justified by unanticipated reasons, there are cases in which rules can be excluded by unexpected exceptions.

5. Statements as procedural arguments

In section 3 was described how structural arguments can be justified, attacked, and built in a procedure. In the previous section it was claimed that not all structural arguments have underlying rules. In this section it is claimed that not all arguments are structural arguments. Therefore the notion of a procedural argument is worked out.

Structural arguments can be characterized as logical arguments, procedural arguments as psychological arguments. The latter arguments are not logically compelling, but do convince in a procedure. The idea of procedural arguments is inspired by the work of Stevenson[6]. He claims that the relation between reasons, adduced for or against a normative statement, and the normative statement itself are not logical relations but rather only psychological.

In procedures, relations between statements can remain implicit. If the relation between two statements is not made explicit (like it is in a structural argument), and one statement is accepted after the introduction of an other statement, the latter statement is a procedural argument. In general a procedural argument is a statement that contributed to the acceptance of another statement, without structural arguments being used. It is a statement that itself or in combination with other non-structural arguments, leads to the acceptance of the other statement.

A trivial procedural argument is a statement that is accepted right after it has been put forward. Such a statement is accepted, because of its intrinsic cogency. For instance, in DiaLaw a statement can be accepted immediately after it is claimed, so before any supporting structural or procedural argument is claimed.

Non-trivial procedural arguments are statements that helped to accept other statements. I will give an example that illustrates the difference between a structural and procedural argument.

```
punishable(jeffrey)     punishable(jeffrey)
?                       ?
murderer(jeffrey)       reason(murderer(jeffrey), punishable(jeffrey))
```

In the left side mini-dialog a procedural argument is adduced, in the right side dialog a structural argument. The relation between the two statements in the left dialog remains implicit

[6] Stevenson [1969] is not directly consulted, but known through Alexy [1989, p. 42f.]

in structure, but is explicit in the procedure. The statement that Jeffrey killed someone else is a procedural argument that supports the statement that Jeffrey is punishable.

The importance of allowing procedural arguments, is in the first place because sometimes it is hard to adduce structural arguments, while procedural arguments can be given. This means that in order to support a statement other statements (procedural arguments) can be claimed, without the need to explicitly indicate relations (structural arguments). For instance, in one of the above examples (see section 4.3) the Dutch Supreme Court used three statements why a particular rule about young people should not be applied to elderly people. This decision can be modeled as a structural argument in which the three statements support the statement that the rule should not be applied. More in spirit with the decision is to model the acceptance of the statement that the rule should not be applied, after the three statements are claimed, and thus without the claim of some artificial structural argument. In section 4.3 it was claimed that not all reasons are the result of rule application. The representation of the decision with procedural arguments goes even beyond the representation proposed in section 4.3. The structural arguments (reasons) are replaced by procedural arguments (statements).

Another point in favor of using procedural arguments is that it is essential to use procedural arguments in case structural arguments are built during the process. The lawyer who justifies a statement needs to be able to use statements about the validity of a rule, the application of a rule, etc. Besides that in the end a structural argument can be built, that will be often of the outweighing type, it can also be the case that a statement is accepted without any structural argument being used.

Summarizing, it is not necessary to use structural arguments in order to justify a statement in a procedure, or in terms of a dialog, to convince the opponent. Often procedural arguments are a better means. In many cases procedural arguments are even necessary, because structural arguments cannot, or not immediately, be put forward.

6. DiaLaw, allowing structural and procedural arguments

A procedural model of argumentation must allow both structural and procedural arguments. Structural arguments to explicitly indicate relations between statements, and procedural arguments merely to convince without indicating an explicit relation. The importance of modeling procedural arguments was explained in the previous section. Structural arguments are necessary if a relation between statements has to be indicated explicitly. It is not claimed that it is inferior to use structural arguments, just that it is not always possible or necessary.

The model DiaLaw is addressed only briefly. For precise details of the game I refer to [Lodder and Herczog 1995][7].

DiaLaw is a two person dialog game, in which both players make moves alternately. The goal of the game is to justify statements in a dialog. The sentences put forward by one player, become justified because of acceptation by the other player. The dialogue game is a rhetorical procedure ([Perelman and Olbrechts-Tyteca 1971], [Witteveen 1988]). Characteristic of such a procedure is that there is no fixed outcome, the procedure is non-deterministic. By presenting arguments, either structural or procedural ones, each party tries to draw the outcome in his direction, but the final result cannot be determined in advance.

The key idea of DiaLaw is that justification of a statement can solely be based on agreement of participants in a dialog. This agreement can be reached by:

[7] A newer, adapted and extended version of DiaLaw, can be obtained by the author.

1. immediate acceptance (trivial procedural arguments);
2. acceptance after only procedural arguments were adduced;
3. acceptance after also structural arguments have been used.

The question may arise whether *anything* can be justified, solely by acceptance of opponents in a dialog. Acceptation in a dialog as criterion for justification seems rather shallow. However, because an external criterion of justification, i.e. a criterion outside the procedure, is impossible to define, it is essential to define the criterion inside the procedure, i.e. a procedural criterion. [see e.g. Hage *et al.* 1994, Lodder 1996]. Regarding all interests, procedural and structural arguments lead to justified statements.

7. Related research on the discussed topics

Of all topics discussed in this paper briefly related research is evaluated.

7.1 Justifying structural arguments

Justification of arguments is part of the Pleadings Game [Gordon 1995]. Each structural argument has a condition part, and a part in which it is expressed that the rule applied in the argument is valid (called backing, after Toulmin). To my knowledge, besides DiaLaw and the Pleadings Game, there are no other procedural models of argumentation that model the justification of arguments.

7.2 Attacking structural arguments

Except for attacking the underlying rule in general, procedural models allow all other types of attack (e.g. Prakken and Sartor [1996], Verheij [1996], Nitta and Shibasaki [1997]).

An interesting addition to extending the ways to attack an argument is the notion of the rationale [Loui and Norman 1995; see also Prakken and Sartor 1997]. A rationale explains why a particular conclusion of a case is correct or why a particular rule is assumed to be valid. Rationales are comparable to the backing of warrants in Toulmin's terminology. One of the five rationale types is the compression rationale. An argument of the form 'a therefore b', can be the result of two chained rules, e.g. 'a therefore c' and 'c therefore b'. If the opponent knows that c is not the case, he cannot directly attack the original argument 'a therefore b'. The use of a c-rationale will help in this situation. Informally the use of the rationale can be described as "you introduced your argument '...', but it actually comes down to '...' Once the rationale is introduced, an argument supporting the negation of c can be put forward as a counterargument.

7.3 Building structural arguments

No model allows to build an argument procedurally as it was described in this paper. Closest to the present description are Verheij [1996] and Vreeswijk [1993]. They build, in a procedure, so-called argument trees.

7.4 Structural arguments without underlying rules

Procedural models have rules underlying arguments [e.g. Gordon 1995, Prakken and Sartor 1996]. To my knowledge, DiaLaw is the only model in which arguments *can* have underlying rules, but do not have them *necessarily*.

7.5 Procedural arguments

Most models only model structural arguments. Gordon's Pleadings Game [1995] models trivial procedural arguments: statements that are part of a structural argument, e.g. the condition, can be conceded. However, it is not possible to support statements by other than structural arguments.

The idea of procedural arguments can be recognized in the work of Hamblin [1971] and MacKenzie [1979]. Their games know no structural arguments[8]. In the game the players alternately adduce a statement. After a statement is introduced both players become committed to this statement. If the other player disagrees, he has to let it know explicitly. In case a player does disagree, the player who introduced the statement will try to convince his opponent by adducing other statements, i.e. procedural arguments.

8. Conclusion

To summarize the main results of this paper, a distinction has been made between structural and procedural arguments, and it was illustrated why both argument types are important in a procedural model of legal argumentation.

There are many procedural models, all have their own merits. Although it depends on the purpose of a model what should be contained in it, I think that all aspects of arguments described in this paper are important and necessary in general. In future work a formal definition of the notion of procedural arguments will be given.

Although DiaLaw represents both procedural and structural arguments, the model is primarily a theoretical model, still far from being a model that can be easily used by lawyers. An important aim for the future is to make models suitable for lawyers. To realize this, exchange between practice and theory [Verheij *et al*. 1997] is essential.

9. References

*Alexy, R. (1989). Theory of legal argumentation

*Eemeren, F.H. van, and R. Grootendorst (1987)

Gordon, Thomas F. (1993/1995). *The Pleadings Game - An artificial intelligence model of procedural justice*. PHD-thesis, Technische Hochschule Darmstadt, Germany. Also published by Kluwer, The Netherlands, 1995.

Hage, J.C., G.P.J. Span, and A.R. Lodder (1992), A Dialogical Model of Legal Reasoning, in C.A.F.M. Grütters *et al.* (eds.), *Legal Knowledge Based Systems: Information Technology and Law , JURIX '92*, Koninklijke Vermande, Lelystad, NL

[8] Some statements closely resemble structural arguments. For instance, by the move *resolution demand* an opponent is asked to solve an inconsistency in his commitment, e.g. 'Resolve whether (If P then Q and P) and not Q'.

Hage, Jaap (1997). *Reasoning with rules. An essay on legal reasoning and its underlying logic*. Kluwer, Dordrecht.

Hamblin, C.L. (1971). Mathematical models of dialogue. *Theoria 2*, 130-155.

Lodder, Arno R; Aimée Herczog. (1995) DiaLaw - A dialogical framework for modelling legal reasoning. *Proceedings of the fifth International Conference on Artificial Intelligence and Law*, ACM, New York, p. 146-155.

Loui, R.P. and J. Norman (1995). Rationales and argument moves. *Artificial Intelligence and Law*. **3**: 159-189.

Loui, Ronald. P., Jeff Norman, Jon Olson and Andrew Merill (1993). A design for reasoning with policies, precedents and rationales. *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, p. 202-211. ACM, New York.

Mackenzie, J.D., Question-Begging in non-cumulative systems, in: *Journal of Philosophical Logic*, 8 (1979) 117-133, 1979.

Nitta, Katsumi, and Masato Shibasaki (1997). Defeasible reasoning in Japanese criminal jurisprudence. *Artificial Intelligence and Law*. **5**: 139-159.

Perelman, Ch. and L. Olbrechts-Tyteca (1971). *The New Rhetoric A Treatise on Argumention*. University of Notre Dame Press, London.

Prakken, H. and G. Sartor (1996). A dialectical model of assessing in conflicting arguments in legal reasoning. *Artificial Intelligence and Law*. **4**: 331-368.

Prakken, H. and G. Sartor (1997). Reasoning with precedents in a dialogue game. *Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, pp. 1-9. ACM, New York.

*Stevenson (1969), *Ethics and language*

Toulmin, S.E. (1958). *The uses of argument.* University press, Cambridge.

Verheij, Bart (1996). *Rules, Reasons, Arguments. Formal studies of argumentation and defeat.* Dissertation Universiteit Maastricht.

Verheij, Bart, Jaap Hage and Arno R. Lodder (1997). Logical tools for legal argument: a practical assessment in the domain of tort. *Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, pp. 243-249. ACM, New York.

Vreeswijk, G.A.W. (1993). *Studies in Defeasible Argumentation*. Doctoral thesis, Free University, Amsterdam.

*Walton and Krabbe (1995)

Witteveen, W.J. (1988). *De retoriek in het recht*, W.E.J. Tjeenk Willink, Zwolle.