



Dialectical argumentation with argumentation schemes: An approach to legal logic

BART VERHEIJ

Department of Artificial Intelligence, University of Groningen, The Netherland
E-mail: b.verheij@ai.rug.nl

Abstract. This paper describes an approach to legal logic based on the formal analysis of argumentation schemes. Argumentation schemes – a notion borrowed from the field of argumentation theory – are a kind of generalized rules of inference, in the sense that they express that given certain premises a particular conclusion can be drawn. However, argumentation schemes need not concern strict, abstract, necessarily valid patterns of reasoning, but can be defeasible, concrete and contingently valid, i.e., valid in certain contexts or under certain circumstances. A method is presented to analyze argumentation schemes and it is shown how argumentation schemes can be embedded in a formal model of dialectical argumentation. The approach also provides insight into the role of critical questions.

1. Introduction

Substantial progress has been made in the logic of law. Abstract topics like defeasibility and the process of argumentation have been formally analyzed with greater theoretical satisfactoriness, and concrete topics like precedent-based and statute-based reasoning have been formally analyzed in more detail and with greater empirical adequacy than before. Among the topics addressed are exceptions to rules, rule applicability, inconsistency handling, reasoning with priorities, the weighing of reasons, case comparison, analogy, the role of principles, values and goals, argument attack and defeat, burden of proof and the dynamics of reasoning.

In this research, many formal patterns of legal reasoning have been explicated, which has resulted in a deeper understanding of these reasoning patterns. Nonetheless, no formalism has gained the status of the canonical logic of law. Moreover, although there are signs of convergence it is not easy to get a coherent unified view on the different opinions and approaches that have been proposed. Recently some attempts towards unification have been made to bridge the unwarranted gap between precedent-based and statute-based reasoning models (cf. the work of Hage 1993, 1997; Prakken and Sartor 1996, 1998), but this integrative task has not yet been completed to full

satisfaction.¹ Recent work by Bench-Capon and Sartor (2003) on the role of values in precedent-based reasoning adds to the diversity of approaches. A reason for the diversity can be that there still is no common set of primitive notions that has turned out sufficiently flexible and expressive to accommodate the different approaches.

The present paper proposes the use of argumentation schemes – a concept borrowed from the field of argumentation theory² – as the main tool of analysis. Argumentation schemes are sensible patterns of reasoning that occur in ordinary argumentation. More specifically, argumentation schemes express that given certain premises a particular conclusion can be drawn. From the point of view of logic, argumentation schemes correspond to the rules of inference of a logical system.³ Argumentation schemes are however often defeasible (i.e., there can be exceptional circumstances in which the scheme's conclusion does not follow from its premises) and contingent (i.e., there can be conditions for the scheme's use). In the paper, a model of dialectical argumentation with argumentation schemes is presented as an attempt to provide a unifying approach to the formal modeling of legal reasoning, i.e., an approach to legal logic.⁴

In Sections 2 and 3 the concept of argumentation schemes is introduced and developed. Section 4 proposes a method for analyzing argumentation schemes. In Section 5, the role of critical questions is discussed. Sections 6 and 7 contain the formal approach to argumentation schemes in terms of the dialectical logic DEFLOG. Section 8 contains an application to the concrete logic Reason-Based Logic.

2. Argumentation schemes

Argumentation follows patterns. Here are two examples of patterns occurring in argumentation:

- (1) P . If P then Q . Therefore Q .
- (2) All P s are Q s. Some R is not a Q . Therefore some R is not a P .

The first is known as *Modus ponens*, which plays a major role in logical proof systems. The second is one of the categorical syllogisms that were already studied by Aristotle. These examples are well-known from standard logic.

But there are many other patterns in argumentation. For instance, the following:

- (3) Person E says that P . Person E is an expert with respect to the fact that P . Therefore P .
- (4) Doing act A contributes to goal G . Person P has goal G . Therefore person P should do act A .

- (5) As a rule, if P then Q . It is the case that P . There is no exception to the rule that if P , then Q . Therefore it is the case that Q .

Scheme (3) expresses a variant of argumentation from expert opinion, (4) a variant of means-end reasoning. Scheme (5) is a variant of *Modus ponens*, called *Modus non excipiens* (Verheij 1999b), that takes the possibility of rule exceptions into account.

All five example schemes are sensible patterns of argumentation. In real-life argumentation, the schemes (3)–(5) are very relevant, probably more than the schemes (1) and (2). Still (1) and (2) (or variants of them) are more often encountered in books on logic and argumentation than (3)–(5).

For present purposes, there is no need to dive deeply into the reasons for this somewhat paradoxical situation. It is however not hard to think of some reasons that may be adduced for this situation:

- The first two schemes fit in neat formal systems: *Modus ponens* (1) in natural deduction and other logical proof systems, scheme (2) in the complete classification of categorical syllogisms. It is not to be expected that the other schemes will appear in formalisms with similarly neat formal properties. (cf., e.g., Copi 1982)
- The first two schemes are thought of as necessarily valid schemes, and the latter three as pragmatical validities, or perhaps not even as validities, but as contingent argumentation schemes, i.e., schemes that can only be used in certain contexts or under certain circumstances. For instance, in a context of strict rules, i.e., rules that cannot have exceptions, the use of scheme (5) makes little sense.
- The first two schemes are strict in the sense that they allow no exceptions, while the latter three are defeasible. For instance, with respect to (3), there is the exception that the expert is wrong, and with respect to (4), there can be many ways to achieve a goal, some better than others.
- The first two schemes express abstract general schemes, where the latter three express concrete context-dependent schemes.

So, logical rules of inference seem to be neatly formalizable, necessarily valid, strict and independent of context, while pragmatic argumentation schemes are pragmatically valid or even contingent, defeasible and context-dependent.

It may therefore seem that a logical approach in which necessity, strictness and abstraction from context seem to be so crucial, is not appropriate for dealing with pragmatic argumentation schemes, such as (3)–(5). For instance, Toulmin (1958) has argued against the logic approach pointing at similar properties of real-life argumentation. The informal logic community has raised related issues (see van Eemeren et al. (1996) for a general overview).

Notwithstanding the distinctive nature of pragmatic argumentation schemes, it is certainly not the case that the methods of logic are of no value at all. This is for instance shown by the fruitful work on legal reasoning in the field of artificial intelligence and law (e.g., Prakken and Sartor 1996, 1998; Verheij 1996; Hage 1997; Bench-Capon and Sartor 2003), but also by Pollock's OSCAR project on building an artificial reasoner (e.g., Pollock 1995).

Here an attempt is made to show how pragmatic argumentation schemes can be systematically analyzed using formal methods. The key observation backing the approach is that there is an obvious structural resemblance between logical rules of inference like (1) and (2) and pragmatic argumentation schemes like (3)–(5): both consist of one or more premises and a conclusion, all possibly with variables that must be instantiated uniformly over the sentences in the scheme. As a result of this resemblance, logical rules of inference and pragmatic argumentation schemes will here all be treated as schemes of the premises-conclusion form (but see note 3).

Though the approach is logic-oriented, it deviates in several ways from conceptions of traditional logic. The approach can be described as a *concrete dialectical logic* approach.⁵ The approach is concrete in the sense that instances of the schemes presented can pertain to concrete argumentative situations (like the instances of (3)–(5)), and do not need to be generally applicable and independent of context. The approach is dialectical in the sense that the schemes are subject to counterarguments: there can be situations in which the scheme does not lead to its conclusion even though its premises obtain. The approach is logical in the sense that it uses formal methods for the analysis and representation of argumentation in a style related to formal logic.

Argumentation schemes are sensible patterns of reasoning that occur in ordinary argumentation. In the following we will speak of argumentation schemes casually as if it is intuitively clear what is an argumentation scheme and what is not. In spite of this, it is worth noting that there are of course no clear, predefined boundaries between patterns in argumentation that count as acceptable argumentation schemes and those that do not. Here a pragmatic approach is taken: the acceptability is considered to be determined by the context. For instance, the following is a pattern of argumentation that is common in a legal context:

- (6) Person P has committed crime C . Crime C is punishable by n years of imprisonment. Therefore person P can be punished with up to n years of imprisonment.

In the legal context, this scheme is certainly an acceptable scheme, outside a legal context it is irrelevant. An even more concrete example is the following:

(7) Person *A* is an unmarried man. Therefore person *A* is a bachelor.

Even this scheme can be regarded as an acceptable argumentation scheme, in the admittedly small context of social relations: the underlying rule that unmarried men are bachelors is then regarded as fixed in that context (cf. the logic of love and hate, Verheij 1996, p. 22, 1999b). Many will not think of the latter schemes as argumentation schemes, but as some other kind of rule.

According to the view underlying the present work, there is a continuum ranging from abstract reasoning patterns (like logical rules of inference, such as *Modus ponens*) via contextual reasoning patterns (like pragmatic argumentation schemes, such as argumentation from expert opinion) to domain rules (like the examples (6) and (7)). All can be given the premises-conclusion form. In this view, it is hardly significant where one places the boundary between argumentation schemes (form) and the other rules (content). In Figure 1, the border between form and content has been drawn in two ways. Rules and patterns below a chosen line count as argumentation schemes/rules of inference, those above the line as rules expressing content. Using the border extending to the right, only the abstract reasoning patterns count as formal rules and are part of the logic. Contextual reasoning patterns and domain rules are content and must be expressed as premises. The result is an abstract conception of logic. By the border extending to the left, also the contextual reasoning patterns express form and are part of the logic, hence a concrete, contextual conception of logic. The border can even be placed higher in the continuum than the examples in the figure to include domain rules (like the examples (6) and (7)) (see Verheij (1999b) for an account of concrete and abstract logic).

The present paper is however not about the question which rule-like schemes are to be counted as argumentation schemes and which not. It is assumed that there are rule-like schemes that express sensible patterns of reasoning as they occur in ordinary argumentation. The topic of the paper is how such argumentation schemes can be systematically analyzed.

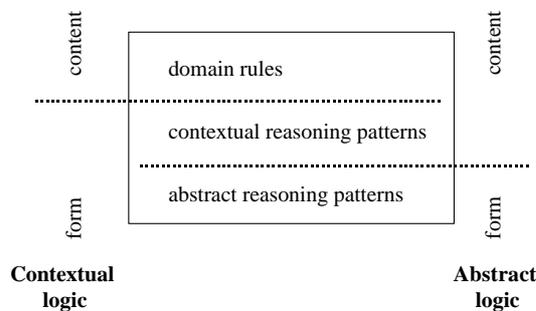


Figure 1. The border between form and content.

Walton's work on argumentation schemes was already mentioned (see note 2). In the next section it is discussed how Walton uses argumentation schemes in some of his recent work. This will provide motivation for the formally oriented methodology discussed in Section 4. First, the focus is on the method itself and no formal details are provided. In the Sections 6 and 7 a formally oriented discussion is given. In Section 5 it is shown how the approach of the paper clarifies the different roles of the critical questions that accompany argumentation schemes.

3. Walton on argumentation schemes⁶

A central aspect of Walton's recent work on argumentation concerns argumentation schemes (cf. e.g., Walton 1996). Argumentation schemes represent kinds of argument as they occur in conversation. Arguments based on argumentation schemes need not be conclusive, but can be defeasible. Walton lists argumentation schemes as a kind of semi-formal argument templates. For instance, in his book on ad hominem arguments the scheme 'Generic Ad Hominem Argument' looks as follows (Walton 1998, p. 249):

GENERIC AH

a is a bad person.

Therefore, *a*'s argument α should not be accepted.

While GENERIC AH looks like a semi-formal rule of inference, other argumentation schemes are like small derivations or pieces of dialogue (p. 256–257):

GUILT BY ASSOCIATION AH

a is a member of or is associated with group *G*, which should be morally condemned.

Therefore, *a* is a bad person.

Therefore, *a*'s argument α should not be accepted.

TWO WRONGS AH

Proponent: Respondent, you have committed some morally blameworthy action (and the specific action is then cited).

Respondent: You are just as bad, for you also committed a morally blameworthy action (then cited, generally a different type of action from the one cited by the proponent but comparable in respect of being blameworthy). Therefore, you are a bad person, and your argument against me should not be accepted as having any worth.

Note that GENERIC AH occurs in GUILT BY ASSOCIATION AH and TWO WRONGS AH (literally in the former, and with a minor adaptation in the latter).

Argumentation schemes come with critical questions, that can be asked to question the dialectical relevance of an argument based on the scheme. GENERIC AH has the following three (p. 249):

cQ1

Is the premise true (or well supported) that *a* is a bad person?

cQ2

Is the allegation that *a* is a bad person relevant to judging *a*'s argument α ?

cQ3

Is the conclusion of the argument that α should be (absolutely) rejected even if other evidence to support α has been presented, or is the conclusion merely (the relative claim) that α should be assigned a reduced weight of credibility, relative to the total body of evidence available?

In the book *Ad Hominem Arguments*, Walton uses argumentation schemes to classify different types of *ad hominem* arguments. He lists twenty-one argumentation schemes that are related to *ad hominem* style argumentation. As main types, he distinguishes the *direct* (or *ethotetic*) variant, in which an arguer's character is attacked, the *circumstantial* variant, in which an arguer advocates a claim that contradicts his earlier claims or behavior, and the *bias* variant, in which an arguer is discredited because of his biases (the main schemes for the three variants occur on the pages 249, 251 and 255, respectively).

For someone with a formal background, the way in which Walton uses argumentation schemes seems rather loose. He uses variables, like *a* for an arguer and *A* for a claim, but it turns out that different occurrences of a variable need not be identical. For instance, the scheme 'Argument from Commitment' is as follows (p. 248):

AC

a is committed to proposition *A* (generally, or in virtue of what she said in the past).

Therefore, in this case, *a* should support *A*.

The third critical question associated with AC is the following:

cQ3

Is the proposition *A*, as cited in the premise, identical to the proposition

A as cited in the conclusion? If not, what exactly is the nature of the relationship between the two propositions?

Apparently, *A* in the premise can differ from *A* in the conclusion. Also, the schemes and the critical questions as they are used by Walton cannot be regarded as purely formal specifications of kinds of arguments, in the sense that actual arguments simply are obtained by filling in variables. Good use of the schemes and questions requires further interpretation by a competent language user (cf. for instance the scheme TWO WRONGS AH cited above).

This may sound as a criticism, but it is not meant that way. Although Walton's looseness may for the formally inclined be somewhat unsettling, it can be warranted by Walton's goal: provide tools for the analysis and evaluation of real-life arguments. Purely formal schemes and questions might not be sufficiently flexible for that goal.

At the same time, it is tempting to investigate how far one can get with the formal method. The task then becomes to formalize concrete kinds of argumentation as they occur in real-life contexts, such as the law. The result would be the design of concrete, contextual logics, such as a logic of law (cf. Section 2 and Verheij 1999b). Of course, formalization leads to *idealizations* of the kinds of argumentation (which would lead Walton away from his focus on real-life argument), but there is the gain of unambiguous precision and the possibility of computer implementation.

In this paper, a method for the investigation of argumentation schemes along these lines is proposed.

4. A method for the investigation of argumentation schemes

The methodology for the investigation of argumentation schemes proposed in the present paper consists of four steps:

1. determine the relevant types of sentences;
2. determine the argumentation schemes;
3. determine the exceptions blocking the use of the argumentation schemes;
4. determine the conditions for the use of the argumentation schemes.

Though the steps have a natural order, the order need not be strictly adhered to while investigating argumentation schemes. Findings in one scheme will regularly lead to a return to a previous step.

As a start the method is discussed semi-formally. The embedding of argumentation schemes into a formal, dialectical logic is discussed in Sections 6 and 7. A rudimentary version of the methodology was already applied by Verheij (1996, chapter 2) in the context of legal reasoning.

4.1. STEP 1: DETERMINE THE RELEVANT TYPES OF SENTENCES

In Step 1, the relevant types of sentences are determined. The relevant sentence types are the building blocks of the argumentation schemes, viz. their premises and conclusions. Sentence types are sentences that contain variables. Each instantiation of the variables of the sentence type gives rise to a sentence. Sentence types can be identified extensionally, i.e., by considering the set of sentences that are instances of the sentence type. From the point of view of formal logic, Step 1 is the step in which the *language* is defined. Let's go back to the examples of the beginning of Section 2:

- (1) *P*. If *P* then *Q*. Therefore *Q*.
- (2) All *Ps* are *Qs*. Some *R* is not a *Q*. Therefore some *R* is not a *P*.
- (3) Person *E* says that *P*. Person *E* is an expert with respect to the fact that *P*. Therefore *P*.
- (4) Doing act *A* contributes to goal *G*. Person *P* has goal *G*. Therefore person *P* should do act *A*.
- (5) As a rule, if *P* then *Q*. It is the case that *P*. There is no exception to the rule that if *P*, then *Q*. Therefore it is the case that *Q*.

Scheme (1) uses one sentence type as a building block of the scheme:

If *P* then *Q*.

Here the variables *P* and *Q* are placeholders for sentences.

Scheme (2) uses two sentence types. This time *P* and *Q* are placeholders for properties:

All *Ps* are *Qs*.
Some *P* is not a *Q*.

Scheme (3) uses two:

Person *E* says that *P*.
Person *E* is an expert with respect to the fact that *P*.

E is a placeholder for (the name of) a person, and *P* for a sentence.

Scheme (4) uses three, where *A* stands for some act, *P* for a person and *G* for a goal:

Doing act *A* contributes to goal *G*.
Person *P* has goal *G*.
Person *P* should do act *A*.

Finally, scheme (5) uses three sentence types. P and Q are placeholders for sentences:

As a rule, if P then Q .
 It is the case that P .
 There is no exception to the rule that if P , then Q .

These examples may suggest that it is always possible to give a finite list of the relevant sentence types. Abstract logics (such as standard propositional logic) show us that this is not always the case, for instance when sentence types can be nested. An example is a sentence type expressing negation:

It is not the case that P .

Here P stands for a sentence. Of course negations can be negated, etc., leading to an infinite list of sentence types that are theoretically all meaningful, but in practice will only occur up to a level of complexity that can be humanly grasped:

It is not the case that P .
 It is not the case that it is not the case that P .
 It is not the case that it is not the case that it is not the case that P .
 ...

For determining the sentence types, it is not required that they can be finitely listed. Instead, it suffices that the sentence types are somehow effectively specified, for instance by a simple listing (as in our examples) or by a recursive definition, as is common in formal logic.

4.2. STEP 2: DETERMINE THE ARGUMENTATION SCHEMES

The second step is the determination of the schemes themselves. Argumentation schemes express that given certain premises a particular conclusion can be drawn. From the point of view of logic, this step corresponds to the definition of the *rules of inference* of a logical system. The general form of argumentation schemes that is used in the present paper is the following:

Premise₁. Premise₂ ... Premise_n. Therefore Conclusion.

The four schemes of the beginning of Section 2 all have this form.

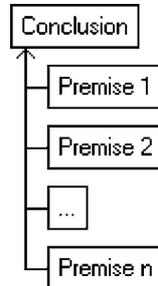


Figure 2. The premises-conclusion structure of an argumentation scheme.

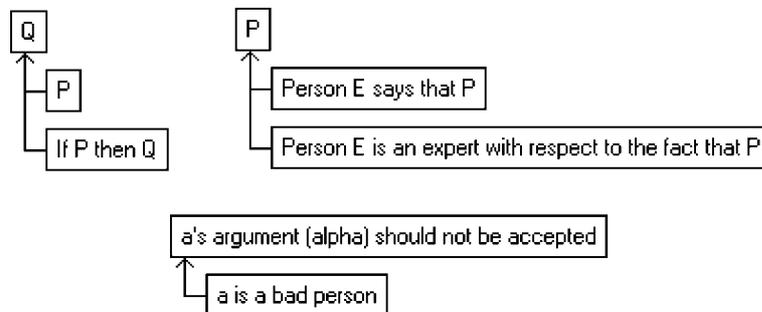


Figure 3. The graphical representation of some argumentation schemes.

The structure of a scheme is graphically represented in Figure 2 (in the style of the ArguMed system, Verheij 1999a, 2003). It shows the basic premises-conclusion form of argumentation schemes as used in this paper. Figure 3 shows the examples (1), (3) and *GENERIC AH* discussed earlier.

Elementary arguments are simply instances of an argumentation scheme. Complex arguments are chains of instances of an argumentation scheme. In a complex argument, a conclusion of one scheme can occur as a premise in another (subordination of arguments). It can also occur that schemes have the same conclusion (coordination of arguments).

The examples of Section 3 given by Walton are not all of this form. His scheme *GUILT BY ASSOCIATION AH* consists of an argumentative chain of two steps, while his *TWO WRONGS AH* is a small dialogue. The first example does not indicate a genuine restriction of the present methodology. It is very well possible to consider compositions of argumentation schemes whenever appropriate. From the point of view of the present methodology, the one-step building blocks of a composite structure would count as argumentation schemes themselves. For *GUILT BY ASSOCIATION AH*, this means that its parts

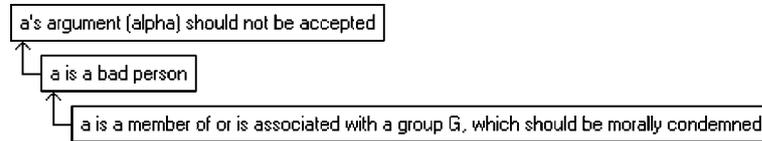


Figure 4. The graphical representation of GUILT BY ASSOCIATION AH.

a is a member of or is associated with group G , which should be morally condemned.

Therefore, a is a bad person.

and

a is a bad person.

Therefore, a 's argument α should not be accepted.

are considered as two separate schemes. In actual arguments, they can of course be chained. (cf. Figure 4.)

Arguably, the second deviating example, the mini-dialogue TWO WRONGS AH does provide a restriction of the present methodology, since the dialogue context of argumentation is here not explicitly addressed. Note however that the methodology addresses counterarguments, only not in a dialogue setting (see below).

In logic, there are rules of inference that do not fit in the premise-conclusion form of argumentation schemes assumed here. An important class of rules of inference allows the withdrawal of premises. For instance, given a derivation of Q using P as a premise, a new derivation can be formed with conclusion $P \rightarrow Q$ in which P is no longer a premise. This rule of inference is often called \rightarrow -Introduction. Since the focus is here on pragmatic kinds of argumentation, in which the withdrawal of premises is not or less relevant (e.g., Walton (1996) does not list schemes involving the withdrawal of premises), this complication is here not addressed. Formally, there is no problem (cf., e.g., Pollock's approach (1995) that uses sequents, i.e., premise-conclusion pairs, instead of sentences as basic building blocks).

4.3. STEP 3: DETERMINE THE EXCEPTIONS BLOCKING THE USE OF THE ARGUMENTATION SCHEMES

Many schemes do not always lead to arguments justifying their conclusions on the basis of their premises. Schemes are subject to exceptions. An example is provided by scheme (6) of Section 2:

Person P has committed crime C . Crime C is punishable by n years of imprisonment.

Therefore person P can be punished with up to n years of imprisonment.

One exception blocking the use of this scheme can be that there is a ground of justification for committing crime C . By a ground of justification, someone who has committed a crime, is not considered to be punishable for the crime. An example of such a ground of justification is *force majeure*.

Note that the exceptions blocking the use of a scheme can involve additional sentence types, requiring a return to Step 1 and an extension of the language. In the example, the scheme and its counterargument make use of the following sentence types:

Person P has committed crime C .

Crime C is punishable by n years of imprisonment.

Person P can be punished with up to n years of imprisonment.

There is a ground of justification for person P for committing crime C .

The situation is graphically depicted in Figure 5.

Another example of an argumentation scheme with a counterargument is the following, expressing a variant of rule application:

P s are Q s. A is a P . Therefore A is a Q .

Note that we have used the sentence type ' P s are Q s' and not 'All P s are Q s' since here a rule that allows exceptions is intended. Whenever an exception occurs, it blocks the use of the scheme. For instance, when A is an exceptional P , we could get the following exception blocking the scheme's use:

A is an exception to the rule that P s are Q s.

See Figure 6.

The result of counterarguments to a scheme is that – when the counterargument is successful – the scheme's conclusion does not follow from the scheme's premises.

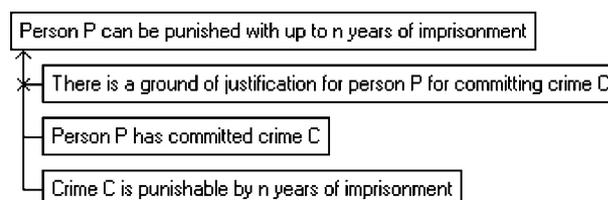


Figure 5. An argumentation scheme with an exception.

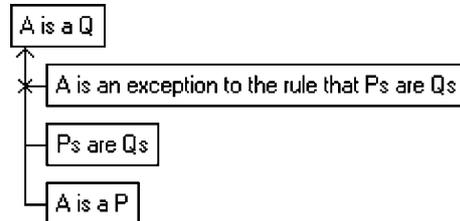


Figure 6. A second example.

4.4. STEP 4: DETERMINE THE CONDITIONS FOR THE USE OF THE ARGUMENTATION SCHEMES

Since argumentation schemes are not universally valid, but can only be acceptable in certain contexts or under certain circumstances, the use of a scheme can depend on the fulfillment of conditions. This property of argumentation schemes is addressed in the fourth, final step of the methodology. It consists of the determination of the conditions for the use of argumentation schemes.

Looking again at scheme (6) of Section 2, one condition can be that person *P* has been brought before a qualified criminal judge. When such conditions are considered to be sufficiently relevant to be made explicit, they require dedicated sentence types (and thus possibly a return to Step 1):

Person *P* has been brought before a qualified criminal judge.

The situation is graphically depicted in Figure 7.

Another example is an alternative version of the rule application scheme ‘*Ps are Qs. A is a P. Therefore A is a Q*’ discussed in Section 4.3. In the alternative, the sentence ‘*Ps are Qs*’ is not a premise of the scheme, but a condition for the use of the scheme ‘*A is a P. Therefore A is a Q*’. cf. Toulmin’s warrants (1958) (see also Section 6.1 below). Figure 8 shows the graphical representation.

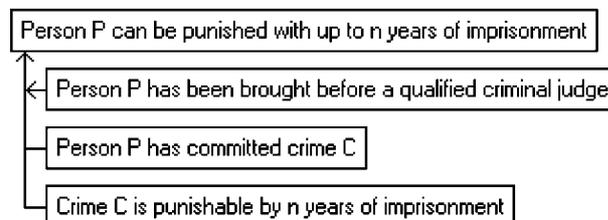


Figure 7. A condition for the use of an argumentation scheme.

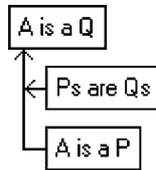


Figure 8. A second example.

There are two natural places to add the exception ‘ A is an exception to the rule that Ps are Qs ’: as a reason blocking the connection between ‘ A is a P ’ and ‘ A is a Q ’ or as a reason blocking the connection between ‘ Ps are Qs ’ and (the conditional underlying) the scheme ‘ A is a P . Therefore A is a Q ’. The two variants are shown in Figure 9.

The variant on the left in Figure 9 treats the rule and the exception in a uniform way as reasons for and against the connection between ‘ Ps are Qs ’, whereas in the variant on the right the exception blocks the rule’s function. The latter is a bit more satisfactory in the sense that it matches the idea that the exception is an exception to the rule. In the formal analysis of argumentation schemes to be discussed below (Sections 6 and 7), the variant on the right is used.

This ends the informal discussion of a formally oriented methodology for the investigation of argumentation schemes. In sum, an argumentation scheme consists of a conclusion, a set of premises, a set of conditions for the use of the scheme and a set of exceptions blocking the use of the scheme (Figure 10). The Sections 6 and 7 provide the formal setting.

5. Critical questions

Before we turn to the formal part of the paper, we briefly address the role of critical questions accompanying argumentation schemes as they appear in the argumentation theory literature (cf. Section 3). Critical questions fit well in

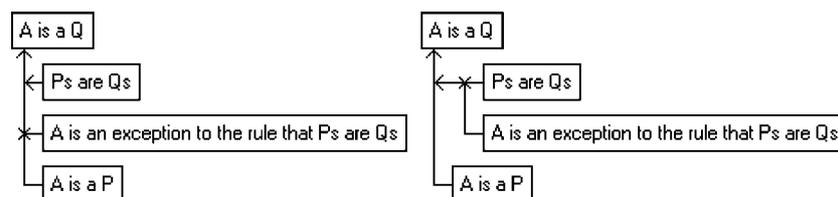


Figure 9. Two variants of the place of an exception.

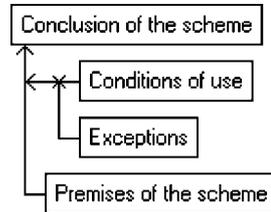


Figure 10. An argumentation scheme with conditions and exceptions.

the presented methodology, but in different ways. This has to do with the fact that critical questions play several roles.

The first role is that of criticizing a scheme's premises. For instance, in Section 2, we encountered the following critical question for the scheme GENERIC AH mentioned by Walton:

CQ1
IS THE PREMISE TRUE (OR WELL SUPPORTED) THAT *A* IS A BAD PERSON?

To any argumentation scheme of the form

*Premise*₁. *Premise*₂. *Premise*_{*n*}. Therefore *Conclusion*.

there are *n* critical questions of this kind:

CQ1
IS *PREMISE*₁ TRUE (OR WELL SUPPORTED)?
CQ2
IS *PREMISE*₂ TRUE (OR WELL SUPPORTED)?
...
CQ*N*
IS *PREMISE*_{*N*} TRUE (OR WELL SUPPORTED)?

In the present methodology, they do not need to be made explicit as exceptions blocking the use of a scheme: a precondition of the use of *any* scheme is that its premises are true, well supported, justified, ... As a result, given a scheme, the critical questions of this kind are already given, albeit implicitly.

The second role of critical questions is to point to exceptional situations in which the scheme should not be used (cf. also Girle et al. 2003). Reconsidering scheme (3) of the introduction

Person *E* says that *P*. Person *E* is an expert with respect to the fact that *P*. Therefore *P*.

the following critical question makes sense:

Did expert *E* not make a mistake?

Obviously, this critical question also points to an exception blocking the scheme's use (cf. Step 3), viz:

Expert *E* made a mistake.

The third role of the critical questions encountered in the literature corresponds to what here are called conditions for a scheme's use. Walton's second critical question for *GENERIC AH* (see Section 2) can be regarded to be of this kind:

cq2

Is the allegation that *a* is a bad person relevant to judging *a*'s argument α ?

If the allegation that *a* is a bad person is relevant to judging *a*'s argument α , the scheme *GENERIC AH* can be used. In this way, the question gives a condition for the scheme's use.

A fourth role of critical questions is to point to other possible arguments relevant for a scheme's conclusion. For instance, for Scheme (3) on following an expert's opinion, a critical question could be whether there are other experts that say or deny *P*, or whether there is other information with respect to *P*. Just like the first role of questioning premises, critical questions of this kind are not specific for a scheme. For *any* scheme it is relevant to find out whether there are other arguments for or against the scheme's conclusion. As a result, this kind of critical question can be asked for *any* scheme. The answer to such questions depends on the other argumentation schemes that pertain in the context.

6. Dialectical argumentation

Recently I have developed a theory of dialectical argumentation and a corresponding logic of dialectical interpretation, called *DEFLOG* (Verheij 2003a). *DEFLOG* is sentence-based instead of rule-based, derivation-based or sequent-based, such as related formalisms (e.g., Reiter 1980; Pollock 1987, 1995; Prakken and Sartor 1996). *DEFLOG* can be used as the basis for a formal

analysis of argumentation schemes along the lines of the previous sections. DEFLOG is related to my work on automated argument assistance (Verheij 1999a, 2003b). In this section, an overview of DEFLOG will be given. Section 7 addresses the issue of embedding argumentation schemes in DEFLOG.

In DEFLOG two logical primitives are introduced: dialectical negation \times and primitive implication \rightsquigarrow . Dialectical negation is used to express that a statement is defeated. Primitive implication is used to express that a statement implies another statement. When φ and ψ are sentences, then $\times\varphi$ and $(\varphi \rightsquigarrow \psi)$ express that φ is defeated and that φ implies ψ , respectively. When p , q , r and s are elementary sentences, then $p \rightsquigarrow (q \rightsquigarrow r)$, $p \rightsquigarrow \times(q \rightsquigarrow \times r)$ and $(p \rightsquigarrow q) \rightsquigarrow (p \rightsquigarrow \times(r \rightsquigarrow s))$ are some examples of sentences. (For convenience, outer brackets are omitted.)

Dialectical negation is not meant to replace standard, classical negation. It is a different notion expressing defeat. Dialectical negation makes a prima facie assumption defeated. In DEFLOG, sets of sentences are interpreted as prima facie assumptions, some of which are evaluated as actually justified, others as actually defeated. Prima facie assumptions are normally evaluated as actually justified. However, when the dialectical negation of a prima facie assumption is also prima facie justified, the assumption is – normally – evaluated as actually defeated and its dialectical negation as actually justified.

Primitive implication is used to express elementary conditional relations and has sharply delimited logical properties: it validates the logical rule Modus ponens (from φ and $\varphi \rightsquigarrow \psi$ conclude ψ) and nothing more. For instance, it does not allow Modus tollens. Note that when the conditional $\varphi \rightsquigarrow \psi$ and its antecedent φ are both actually justified, its consequent ψ is also actually justified. In this sense, the conditional of DEFLOG is conclusive. However a conditional can of course be prima facie assumed, just like non-conditional statements. In other words, it is possible that the conditional $\varphi \rightsquigarrow \psi$ is prima facie justified, but not actually justified.

Before formally defining the central notions of DEFLOG (Section 6.2), some examples are discussed as an illustration (Section 6.1).

6.1. EXAMPLES

Notwithstanding its simple structure, central notions of dialectical argumentation can be analyzed in terms of it.

Prima facie justified assumptions. Consider two prima facie assumptions:

```
innocent
proven_guilty  $\rightsquigarrow$   $\times$ innocent
```

The first sentence expresses the assumption of innocence, the second that when guilt is proven (by proof in the legal sense, not in the logical sense), the assumption of innocence is defeated. Given these two assumptions, the assumption of innocence is not only prima facie justified, but also actually. There is no information that can lead to the defeat of one of the prima facie justified assumptions: the antecedent of the conditional is not satisfied. When however a third assumption *proven_guilty* is added, it follows that *xinnocent*. When the assumptions would be taken as definitely true, an inconsistency arises: both *innocent* and *xinnocent* follow. Since the assumptions are interpreted as being prima facie justified, the situation is different. The prima facie assumption of innocence is countered by its dialectical negation. As a result, the prima facie assumption *innocent* is not actually justified, but defeated. Note that dialectical negation is inherently ‘directed’, in the following sense. Since *xinnocent* follows, *innocent* is defeated. However, it is not the case that since *innocent* is prima facie justified, *xinnocent* is defeated. This is in contrast with standard negation where the truth of a negated sentence, implies the sentence’s falsity, while also a sentence’s truth implies the falsity of its negation.⁷

Undercutting exceptions. An undercutting exception (Pollock 1987, 1995) blocks the connection between a reason and its conclusion. Consider the following prima facie assumptions:

```
testimony
testimony ~> guilty
```

These sentences express that there is a witness testimony that implies that the suspect is guilty. By applying Modus ponens, it follows that the suspect is guilty. However, let’s make two additional prima facie assumptions:

```
unreliable
unreliable ~> x(testimony ~> guilty)
```

The first sentence expresses that the testimony is unreliable. The second expresses that if the witness testimony is unreliable, it is defeated that the testimony implies the suspect’s guilt. When the four sentences are together assumed to be prima facie justified, the prima facie assumption that the testimony implies guilt, is defeated, and it does not follow that the suspect is guilty.

Warrants. Warrants (Toulmin 1958) are inference licenses, i.e., rule-like statements expressing that a reason implies its conclusion. Consider the following prima facie assumptions:

```

testimonies_are_often_truthful
testimonies_are_often_truthful ~> (testimony
~> guilty)

```

The first sentence expresses that witness testimonies are often truthful. The second sentence turns the first into a warrant for the statement that a witness testimony about a suspect's guilt implies the suspect's guilt. The example shows that in DEFLOG conditionals can be nested.

Pros and cons. Consider the following prima facie assumptions:

```

testimony_1
testimony_1 ~> guilty
testimony_2
testimony_2 ~> not-guilty

```

These sentences express that there are two contradicting witness testimonies, one indicating that the suspect is guilty, the other that he is not. The second witness testimony implies the standard negation of the suspect's guilt (*not-guilty*) and not its dialectical negation (*xguilty*). If the negation in *not-guilty* is interpreted as standard classical negation (which is not incorporated in DEFLOG as such), a contradiction follows. A natural way to deal with a situation of conflicting information is by determining how the conflict is solved. For instance, if the first testimony is more trustworthy, it could be an undercutting exception:

```

more_trustworthy
more_trustworthy ~>
(testimony_1 ~> x(testimony_2 ~> not-guilty))

```

Interpreting the six sentences would result in the defeat of the conditional that the second witness testimony implies the suspect's innocence, and no contradiction would follow.

DEFLOG does not contain dedicated primitives that can express priority information and the weighing of reasons. It is possible to add language elements for such notions and define their use in terms of DEFLOG's primitives.

6.2. THE FORMAL DEFINITION OF DIALECTICAL INTERPRETATION IN DEFLOG

The central definition of DEFLOG is its notion of the *dialectical interpretation* of a theory. Formally, DEFLOG's dialectical interpretations of theories are a variant of Reiter's (1980) extensions of default theories, Gelfond and

Lifschitz's (1988) stable models of logic programming, Dung's (1995) stable extensions of argumentation frameworks, and Bondarenko et al.'s (1997) stable extensions of assumption-based frameworks (see Verheij 2003a).

A theory is any set of sentences. A theory represents a set of *prima facie* justified assumptions. When a theory is dialectically interpreted, all sentences in the theory are evaluated, either as (actually) justified or as (actually) defeated (this is in contrast with the interpretation of theories in standard logic, where all sentences in an interpreted theory are assigned the same positive value, namely true, e.g., by giving a model of the theory).

An assignment of the values justified or defeated to the sentences in a theory gives rise to a dialectical interpretation of the theory, when two properties obtain. First, the justified part of the theory must be conflict-free. Second, the justified part of the theory must attack all sentences in the defeated part. Formally the definitions are as follows.

- (i) Let T be a set of sentences and φ a sentence. Then T *supports* φ when φ is in T or follows from T by the repeated application of $\sim\rightarrow$ -Modus ponens (from $\varphi \sim\rightarrow \psi$ and φ , conclude ψ). T *attacks* φ when T supports $\neg\varphi$.
- (ii) Let T be a set of sentences. Then T is *conflict-free* when there is no sentence φ that is both supported and attacked by T .
- (iii) Let Δ be a set of sentences, and let J and D be subsets of Δ that have no elements in common and that have Δ as their union. Then (J, D) *dialectically interprets* the theory Δ when J is conflict-free and attacks all sentences in D . The sentences in J are the *justified statements* of the theory Δ , the sentences in D the *defeated statements*.
- (iv) Let Δ be a set of sentences and let (J, D) dialectically interpret the theory Δ . Then $(\text{Supp}(J), \text{Att}(J))$ is a *dialectical interpretation* or *extension* of the theory Δ . Here $\text{Supp}(J)$ denotes the set of sentences supported by J , and $\text{Att}(J)$ the set of sentences attacked by J . The sentences in $\text{Supp}(J)$ are the *justified statements* of the dialectical interpretation, the sentences in $\text{Att}(J)$ the *defeated statements*.

Note that when (J, D) dialectically interprets Δ and $(\text{Supp}(J), \text{Att}(J))$ is the corresponding dialectical interpretation, J is equal to $\text{Supp}(J) \cap \Delta$, and D to $\text{Att}(J) \cap \Delta$.

There is a lot to say about the dialectical interpretation of theories. An interesting notion – related to Dung's (1995) admissibility – is that of *dialectical justification*, defined as follows. A conflict-free set C is said to *dialectically justify* a statement φ with respect to the theory Δ when C attacks any conflict-free set C' that is incompatible with C (conflict-free sets are incompatible when their union is not conflict-free). In terms of this notion an

elegant criterion for the existence and multiplicity of dialectical interpretations can be given. The reader is referred to Verheij (2003a).

7. Embedding a concrete logic into DEFLOG

The methodological conception of argumentation schemes (Section 4) can be used to specify a ‘concrete dialectical logic’ (Section 2) in DEFLOG. A concrete dialectical logic consists of a *language* and a set of *argumentation schemes*. DEFLOG is a suitable choice,⁸ since DEFLOG is sentence-based instead of derivation-based (see Verheij 2003a).

Let’s discuss a typical example, the one discussed in Section 4.4 and depicted in Figure 9 on the right:

Conclusion: A is a Q .
 Premise: A is a P .
 Condition: Ps are Qs .
 Exception: A is an exception to the rule that Ps are Qs .

The scheme uses three sentence types:

A is a P .
 Ps are Qs .
 A is an exception to the rule that Ps are Qs .

‘ A is a P ’ and ‘ A is a Q ’ are the same type since both have the same sentences as instances.

A concrete dialectical logic based on this scheme will have to make explicit which sentences match these sentence types, e.g., by specifying how the variables A , P and Q can be filled. P and Q could for instance stand for either ‘thief’ or ‘criminal’, and A for ‘John’ or ‘Mary’. The language would then consist of the following finite list of sentences:

John is a thief.
 Mary is a thief.
 John is a criminal.
 Mary is a criminal.
 Thiefs are criminals (including the spelling error and strange capitalization!)
 Criminals are thiefs.
 John is an exception to the rule that thiefs are criminals.
 John is an exception to the rule that criminals are thiefs.
 Mary is an exception to the rule that thiefs are criminals.
 Mary is an exception to the rule that criminals are thiefs.

Spelling errors were not corrected in order to show the intended formal character of the language specification. The details of the formal approach to the specification of the language are not relevant for present purposes. Here we have chosen a format that is close to natural language for presentation purposes, but other approaches (such as predicate logic or Prolog) can be used just as well.

Once the language of the concrete dialectical logic is fixed, argumentation schemes can be specified. In general, a scheme consists of a conclusion, a set of premises, a set of conditions for the use of the scheme and a set of exceptions blocking the use of the scheme (cf. Figure 10):

Conclusion: *Conclusion*
 Premises: *Premise₁, Premise₂, ..., Premise_n*
 Conditions: *Condition₁, Condition₂, ..., Condition_k*
 Exceptions: *Exception₁, Exception₂, ..., Exception_l*

The conclusion, premises, conditions and exceptions of an argumentation scheme are sentence types and not sentences. As a result, we assume that some matching function between the language's sentences and sentence types is available (analogous to unification in logic and matching in Prolog).

In order to embed a concrete dialectical logic into DEFLOG, the first step is to consider the sentences of its language as elementary sentences of DEFLOG. The sentences of the concrete logic can then be combined using DEFLOG's connectives \times and $\sim\rightarrow$. In this way, new sentences become expressible, that were not part of the language of the concrete logic. For instance, the sentences 'thiefs are criminals $\sim\rightarrow$ (John is a thief $\sim\rightarrow$ John is a criminal)' and 'John is a thief $\sim\rightarrow$ \times Mary is a thief' are expressible in the DEFLOG embedding of the example language.

The new sentences of the auxiliary language can be used to embed the argumentation schemes of the concrete logic into DEFLOG. Let's first consider the example scheme. Its DEFLOG embedding consists of the following two sentence schemes:

Ps are Qs $\sim\rightarrow$ (A is a P $\sim\rightarrow$ A is a Q).
A is an exception to the rule that Ps are Qs $\sim\rightarrow$ \times (Ps are Qs $\sim\rightarrow$ (A is a P $\sim\rightarrow$ A is a Q)).

Given this concrete dialectical logic, what happens in an example case? Assume the following:

John is a thief.
 Mary is a thief.
 Thiefs are criminals.
 Mary is an exception to the rule that thiefs are criminals.

It is expected that it follows that John is a criminal, but not that Mary is.

There are three instances of the logic's sentence schemes to which Modus ponens can be applied:

- (1) Thiefs are criminals $\sim\rightarrow$ (John is a thief $\sim\rightarrow$ John is a criminal).
- (2) Thiefs are criminals $\sim\rightarrow$ (Mary is a thief $\sim\rightarrow$ Mary is a criminal).
- (3) Mary is an exception to the rule that thiefs are criminals $\sim\rightarrow$ x (thiefs are criminals $\sim\rightarrow$ (Mary is a thief $\sim\rightarrow$ Mary is a criminal)).

To the first of these Modus ponens can indeed be applied, leading to the conclusion

John is a thief $\sim\rightarrow$ John is a criminal

and then by a second application of Modus ponens to

John is a criminal

To the third instance (3) Modus ponens can also be applied, resulting in:

x (thiefs are criminals $\sim\rightarrow$ (Mary is a thief $\sim\rightarrow$ Mary is a criminal))

This is the dialectical negation of (2). Since there is no further information, (2) is actually defeated, and Modus ponens cannot be applied to it.

In sum, only one sentence of the concrete dialectical logic follows, viz. 'John is a criminal'. The only other consequences are sentences of the auxiliary language and contain DEFLOG's connectives $\sim\rightarrow$ and x .

In general, a scheme of the form

Conclusion: *Conclusion*
 Premises: *Premise₁, Premise₂, ..., Premise_n*
 Conditions: *Condition₁, Condition₂, ..., Condition_k*
 Exceptions: *Exception₁, Exception₂, ..., Exception_l*

is embedded using the following sentence schemes:

Condition₁ & Condition₂ & ... & Condition_k $\sim\rightarrow$ (Premise₁ & Premise₂ & ... & Premise_n $\sim\rightarrow$ Conclusion)

Exception₁ $\sim\rightarrow$ x (Condition₁ & Condition₂ & ... & Condition_k $\sim\rightarrow$ (Premise₁ & Premise₂ & ... & Premise_n $\sim\rightarrow$ Conclusion))

Exception₂ $\sim\rightarrow$ x (Condition₁ & Condition₂ & ... & Condition_k $\sim\rightarrow$ (Premise₁ & Premise₂ & ... & Premise_n $\sim\rightarrow$ Conclusion))

...

Exception_l $\sim\rightarrow$ x (Condition₁ & Condition₂ & ... & Condition_k $\sim\rightarrow$ (Premise₁ & Premise₂ & ... & Premise_n $\sim\rightarrow$ Conclusion))

For readability, φ & ψ $\sim\rightarrow$ χ is used as an abbreviation of φ $\sim\rightarrow$ (ψ $\sim\rightarrow$ χ). The first scheme expresses that given the scheme's conditions its premises imply the scheme's conclusion. The other schemes express that exceptions

block the scheme's use, even when the scheme's conditions are fulfilled. The conditions are considered to be cumulative: all must be satisfied before the premises of the scheme imply its conclusion. Each exception can by itself block the scheme's use. Technically this is achieved in a straightforward generalization of the example.

8. Reason-Based Logic as a concrete dialectical logic

Reason-Based Logic⁹ is an example of a concrete dialectical logic. It has been developed as a theory of legal reasoning with rules. The development of Reason-Based Logic shaped my views on concrete, context-dependent logics (cf. Verheij 1999b). The signs of changes of opinion, shifts of focus and personal biases in the version history of Reason-Based Logic are abundant. Each version can be regarded as an example of a concrete logic with a specific purpose. Below Reason-Based Logic is described as a concrete dialectical logic by specifying a number of argumentation schemes.

Reason-Based Logic is a formal model of rules and reasons, inspired by the role of rules and reasons in the field of law. In Reason-Based Logic, rules are individuals that can have properties. Key properties of rules distinguished in Reason-Based Logic are their validity, applicability or exclusion. These properties are part of the core of Reason-Based Logic – they belong to its 'logical constants'.

In order to allow rules to have properties, they are not represented as sentences, but as terms. In Reason-Based Logic, the validity of a rule with antecedent φ and consequent ψ is for instance expressed by the sentence $\text{Valid}(\text{rule}(\varphi, \psi))$. Reason-Based Logic assumes a translation from sentences to terms.

The main relations between the properties of rules in Reason-Based Logic are (in Verheij's (1996) version) encoded in its semantic constraints. For instance, rules can only be excluded if they are valid. The semantic constraints do not determine which rules are valid, applicable or excluded. The representation of the facts about rules is to a large extent left to the domain theory.

Essential elements of Reason-Based Logic can be rephrased in terms of the argumentation schemes of a concrete dialectical logic. The following are inspired by the semantic constraints of Verheij's (1996) version of Reason-Based Logic.

Scheme 1

Conclusion: $\text{Reason}(\varphi, \psi)$
 Premises: φ
 Conditions: $\text{Valid}(\text{rule}(\varphi, \psi))$
 Exceptions: $\text{Excluded}(\text{rule}(\varphi, \psi))$

Scheme 2

Conclusion: χ
 Premises: Reason(φ , χ)
 Exceptions: Reason(ψ , not- χ)

Scheme 3

Conclusion: χ
 Premises: Reason(φ_1 , χ), Reason(φ_2 , χ), ..., Reason(φ_n , χ)
 Reason(ψ_1 , not- χ), Reason(ψ_2 , not- χ), ..., Reason(ψ_m , not- χ)
 Outweighs($\{\varphi_1, \dots, \varphi_n\}$, $\{\psi_1, \dots, \psi_m\}$, χ)
 Exceptions: Reason(ψ_{m+1} , not- χ), where ψ_{m+1} differs from ψ_1, \dots, ψ_m .

Greek letters φ , ψ etc. are used as metavariables for corresponding sentences and terms in these schemes and not- χ is a metavariable that stands for the negation of χ .

By Scheme 1, if the antecedent of a valid rule is satisfied, the antecedent becomes a reason for the rule's consequent, unless the rule is excluded. According to Scheme 2, if there is a reason for some conclusion, the conclusion follows unless there is also a reason against the conclusion. According to Scheme 3, a conclusion can be drawn if there are reasons for it that outweigh certain reasons against it, unless there is an additional reason against it that is not yet weighed. The schemes depend on each other. For instance, Scheme 2 is normally only applied after an application of Scheme 1 and Scheme 3 presupposes a number of applications of Scheme 1.

Details about Reason-Based Logic can be found in the work of Hage (1993, 1997) and Verheij (1996).

9. Conclusion

In the present paper, a four-step methodology for the investigation of argumentation schemes is proposed:

1. determine the relevant types of sentences;
2. determine the argumentation schemes;
3. determine the exceptions blocking the use of the argumentation schemes;
4. determine the conditions for the use of the argumentation schemes.

Step 3 is related to the defeasibility of argumentation schemes: there can be exceptional situations in which the scheme should not be used. Step 4 has to do with the contingency of schemes: it can be the case that the use of a scheme depends on certain conditions. The methodology is inspired by previous formal work on dialectical argumentation and concrete argumentation schemes (e.g., Verheij 1999b, 2001a). The paper demonstrates a

concrete, dialectical approach to logic by presenting how argumentation schemes can be embedded in a formal dialectical logic. The approach applies to legal logic, as shown by the discussion of Reason-Based Logic.

A deep issue concerning argumentation schemes raised by the proposal is their specificity. To what extent is the specification of argumentation schemes a good idea? Argumentation schemes are variable, flexible and robust: people use the schemes all the time, and do not seem to encounter difficulties in adapting a scheme to neatly fit new circumstances. The specification of argumentation schemes – and especially the formal specification – easily disregards this variation, flexibility and robustness. The example of a concrete logic discussed in Section 8 (Reason-Based Logic) is illustrative in this respect: it has been described in many versions, each with its own specific characteristics and purpose. How to deal with the issue of specificity of argumentation schemes other than in the pragmatic way of specifying what is needed in a specific context is beyond our current state of understanding and reminds of deep questions concerning language use and cognition.

Notes

¹ Roth's dissertation (2003) is also a step in the right direction since it shows how case comparison can take the structure of dialectical arguments into account (cf. Verheij 1999b, 2003a).

² Walton (1996) provides many examples and a useful overview. He builds on work of Perelman and Olbrechts-Tyteca (1958), Hastings (1963) and Kienpointner (1992).

³ Logical rules of inference do not always have the premises-conclusion form. In particular, there are rules of inference that involve the retraction of premises. Cf. the treatment of natural deduction by Gamut (1991, p. 128f.). Here we will restrict ourselves to rules of inference and argumentation schemes of the premises-conclusion form.

⁴ The paper is a follow up of Verheij (2001a, 2003c). In the first, a connection between dialectical argumentation and argumentation schemes is established; in the second a methodology for the investigation of argumentation schemes is proposed. A first expression of the idea (Section 3.3 of Gärle et al. 2003) dates from the inspiring *Symposium on Argument and Computation* in 2000 (see www.csd.abdn.ac.uk/~tnorman/sac/). There is other work on argumentation schemes by attendants of that symposium. Reed and Walton (2001) discuss argumentation schemes in the context of a software tool for analyzing arguments (see www.computing.dundee.ac.uk/staff/creed/araucaria/). Prakken et al. (2003) have used argumentation schemes to analyze reasoning about evidence.

⁵ The concrete dialectical logic approach is developed in my earlier work (Verheij 1996, 1999b, 2000, 2001a). Hage's work has been a major influence (see, e.g., Hage 1993, 1997, 2001).

⁶ This section is adapted from Verheij (2001b).

⁷ Bondarenko et al. (1997) also analyze the presumption of innocence. However, they do not analyze the presumption of innocence *per se*, but instead the conditional expression that someone is innocent unless proven guilty. To this effect they use a weak negation expressing (logical, not legal) non-provability, and a generic non-provability assumption that can be defeated. When \sim denotes non-provability, they use a conditional of the form $\sim\text{guilty} \sim \rightarrow$ innocent to formalize the example. Since $\sim\text{guilty}$ is defeasibly assumed (just like every other statement of the form $\sim\phi$), the innocence follows defeasibly. Note that in this way the presumption of innocence is a conditional with a

defeasibly fulfilled antecedent and not a separate, *prima facie* justified assumption, as in the analysis here.

⁸ There are other reasonable choices, such as Reiter's default logic (1980) and Pollock's theory of defeasible reasoning (1987, 1995).

⁹ Reason-Based Logic was initiated by Hage and elaborated in cooperation with me. See, e.g., Hage (1993, 1997) and Verheij (1996).

References

- Bench-Capon, T. J. M. and Sartor, G. (2003). A Model of Legal Reasoning with Cases Incorporating Theories and Values. *Artificial Intelligence* 150 (1–2): 97–143.
- Copi, I. M. (1982). *Introduction to Logic*, 6th Ed. Macmillan: New York.
- Gamut, L. T. F. (1991). *Logic, Language, and Meaning*. Introduction to Logic. Vol. I. The University of Chicago Press: Chicago (Illinois).
- Girle, R., Hitchcock, D. L., McBurney, P., and Verheij, B. (2003). Decision Support for Practical Reasoning: A Theoretical and Computational Perspective. Reed, C. and Norman, T. J. (eds.) *Argumentation Machines. New Frontiers in Argument and Computation*, 55–84. Kluwer Academic Publishers: Dordrecht.
- Hage, J. C. (1993). Monological Reason Based Logic. A Low Level Integration of Rule-based Reasoning and Case-based Reasoning. In *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, 30–39. ACM: New York.
- Hage, J. C. (1997). Reasoning with Rules. An Essay on Legal Reasoning and Its Underlying Logic. Kluwer Academic Publishers: Dordrecht.
- Hage, J. C. (2001). Legal Logic. Its Existence, Nature and Use. Soeteman A. (ed.) *Pluralism and Law*, 347–373. Kluwer Academic Publishers: Dordrecht.
- Hastings, A. (1963). A Reformulation of the Modes of Reasoning in Argumentation. Doctoral dissertation. Northwestern University.
- Kienpointner, M. (1992). *Alltagslogik: Struktur und Funktion von Argumentationsmustern*. Stuttgart-Bad Cannstatt: Frommann-Holzboog.
- McBurney, P. and Parsons, S. (2000). Tenacious Tortoises: A Formalism for Argument over Rules of Inference. *Computational Dialectics (ECAI 2000 Workshop)*, Berlin, Germany.
- Perelman, C. and Olbrechts-Tyteca, L. (1971/1958). *The New Rhetoric: A Treatise on Argumentation*. University of Notre Dame Press: Notre Dame/London, French original first published in 1958.
- Pollock, J. L. (1987). Defeasible reasoning. *Cognitive Science* 11: 481–518.
- Pollock, J. L. (1995). *Cognitive Carpentry: A Blueprint for How to Build a Person*. The MIT Press: Cambridge, MA.
- Prakken, H. (1997). *Logical Tools for Modelling Legal Argument. A Study of Defeasible Reasoning in Law*. Kluwer Academic Publishers: Dordrecht.
- Prakken, H. and Sartor, G. (1996). A Dialectical Model of Assessing Conflicting Arguments in Legal Reasoning. *Artificial Intelligence and Law* 4: 331–368.
- Prakken, H. and Sartor, G. (1998). Modelling Reasoning with Precedents in a Formal Dialogue Game. *Artificial Intelligence and Law* 6: 231–287.
- Prakken, H., Reed, C. and Walton, D. N. (2003). Argumentation Schemes and Generalisations in Reasoning about Evidence. In *proceedings of The Ninth International Conference on Artificial Intelligence and Law* 32–41. ACM: New York.
- Reed, C. and Walton, D. N. (2001). Applications of Argumentation Schemes. *Argumentation and its Applications*. In Hansen, H. V., Tindale, C. W., Blair, J. A. and Johnson, R. H. (eds.),

- Proceedings of the Fourth Biennial Conference of the Ontario Society for the Study of Argumentation (OSSA 2001)
- Reiter, R. (1980). A Logic for Default Reasoning. *Artificial Intelligence* 13: 81–132.
- Roth, A. C. (2003). Case-based Reasoning in the Law. A Formal Theory of Reasoning by Case Comparison. Dissertation. Universiteit Maastricht.
- Soeteman, A. (1989). *Logic in Law*. Kluwer Academic Publishers: Dordrecht.
- Toulmin, S. E. (1958). *The Uses of Argument*. Cambridge University Press: Cambridge.
- van Eemeren, F. H., Grootendorst, R., and Snoeck Henkemans, F. (1996). *Fundamentals of Argumentation Theory. A Handbook of Historical Backgrounds and Contemporary Developments*. Lawrence Erlbaum: Hillsdale, NJ.
- Verheij, B. (1996). *Rules, Reasons, Arguments. Formal studies of argumentation and defeat*. Dissertation Universiteit Maastricht.
- Verheij, B. (1999a). Automated Argument Assistance for Lawyers. In *Proceedings of the Seventh International Conference on Artificial Intelligence and Law*, 43–52. ACM: New York.
- Verheij, B. (1999b). Logic, Context and Valid Inference. Or: Can there be a Logic of Law? van den Herik, H. J., Moens, M.-F., Bing, J., van Buggenhout, B., Zeleznikow, J. and Grütters, C. A. F. M. (eds.), *Legal Knowledge Based Systems. JURIX 1999: The 12th Conference* 109–121. Gerard Noodt Instituut: Nijmegen.
- Verheij, B. (2000). Book Review: Formalism and Interpretation in the Logic of Law. (review of H. Prakken's 'Logical Tools for Modelling Legal Argument'). *Artificial Intelligence and Law* 8: 35–65.
- Verheij, B. (2001a). Legal Decision Making as Dialectical Theory Construction with Argumentation Schemes. In *proceedings of the 8th International Conference on Artificial Intelligence and Law*, 225–226. ACM: New York.
- Verheij, B. (2001b). Book Review of D. Walton's 'The New Dialectic', 'Ad Hominem Arguments' and 'One-Sided Arguments'. *Artificial Intelligence and Law* 9: 305–313.
- Verheij, B. (2003a). DEFLOG: On the Logical Interpretation of Prima Facie Justified Assumptions. *Journal of Logic and Computation* 13(3): 319–346.
- Verheij, B. (2003b). Artificial Argument Assistants for Defeasible Argumentation. *Artificial Intelligence* 150(1–2): 291–324.
- Verheij, B. (2003c). Dialectical Argumentation with Argumentation Schemes: Towards a Methodology for the Investigation of Argumentation Schemes. In van Eemeren, F. H., Blair, J. A., Willard, C. A. and Snoeck Henkemans, F. *Proceedings of the Fifth Conference of the International Society for the Study of Argumentation (ISSA 2002)* 1033–1037. Sic Sat, Amsterdam.
- Walton, D. N. (1996). *Argument Schemes for Presumptive Reasoning*. Lawrence Erlbaum: Hillsdale, NJ.
- Walton, D. N. (1998). *Ad Hominem Arguments*. The University of Alabama Press: Tuscaloosa.