The role of logic in computational models of legal argument – a critical survey

Henry Prakken[1] and Giovanni Sartor[2]

[1] Institute of Information and Computing Sciences
Utrecht University, The Netherlands
http://www.cs.uu.nl/staff/henry.html
[2] Faculty of Law, University of Bologna, Italy
sartor@cirfid.unibo.it

Abstract. This article surveys the use of logic in computational models of legal reasoning, against the background of a four-layered view on legal argument. This view comprises a logical layer (constructing an argument); a dialectical layer (comparing and assessing conflicting arguments); a procedural layer (regulating the process of argumentation); and a strategic, or heuristic layer (arguing persuasively). Each further layer presupposes, and is built around the previous layers. At the first two layers the information base is fixed, while at the third and fourth layer it is constructed dynamically, during a dialogue or dispute.

1 Introduction

1.1 AI & Law research on legal argument

This article surveys a field that has been heavily influenced by Bob Kowalski, the logical analysis of legal reasoning and legal knowledge representation. Not only has he made important contributions to this field (witness the many times his name will be mentioned in this survey) but also has he influenced many to undertake such a logical analysis at all. Our research has been heavily influenced by his work, building on logic programming formalisms and on the well-known argumentation-theoretic account of nonmonotonic logic, of which Bob Kowalski was one of the originators [Kakas *et al.*, 1992, Bondarenko *et al.*, 1997]. We feel therefore very honoured to contribute to this volume in honour of him.

The precise topic of this survey is the role of logic in computational models of legal argument. Argumentation is one of the central topics of current research in Artificial Intelligence and Law. It has attracted the attention of both logically inclined and design-oriented researchers. Two common themes prevail. The first is that legal reasoning is defeasible, i.e., an argument that is acceptable in itself can be overturned by counterarguments. The second is that legal reasoning is usually performed in a context of debate and disagreement. Accordingly, such notions are studied as argument moves, attack, dialogue, and burden of proof.

Historically, perhaps the first AI & Law attempt to address legal reasoning in an adversarial setting was McCarty's (partly implemented) Taxman project,

which aimed to reconstruct the lines of reasoning in the majority and dissenting opinions of a few leading American tax law cases (see e.g. [McCarty and Sridharan, 1981, McCarty, 1995]). Perhaps the first AI & Law system that explicitly defined notions like dispute and dialectical role was Rissland & Ashley's (implemented) HYPO system [Rissland and Ashley, 1987], which modelled adversarial reasoning with legal precedents. It generated 3-ply disputes between plaintiff and defendant in a legal case, where each dispute is an alternating series of attacks by the defendant on the plaintiff's claim, and of defences or counterattacks by the plaintiff against these attacks. This research was continued in Rissland & Skalak's CABARET project [Rissland and Skalak, 1991], and Aleven & Ashley's CATO project [Aleven and Ashley, 1997], both also in the 'design' strand. The main focus of all these projects is defining persuasive argument moves, moves which would be made by 'good' human lawyers.

By contrast, much logic-based research on legal argument has focused on defeasible *inference*, inspired by AI research on nonmonotonic reasoning and defeasible argumentation [Gordon, 1991, Kowalski and Toni, 1996, Prakken and Sartor, 1996, Prakken, 1997, Nitta and Shibasaki, 1997, Hage, 1997, Verheij, 1996]. Here the focus was first on reasoning with rules and exceptions and with conflicting rules. After a while, some turned their attention to logical accounts of case-based reasoning [Loui *et al.*, 1993, Loui and Norman, 1995, Prakken and Sartor, 1998]. Another shift in focus occurred after it was realised that legal reasoning is bound not only by the rules of logic but also by those of fair and effective procedure. Accordingly, logical models of legal argument have been augmented with a dynamic component, capturing that the information with which a case is decided is not somehow 'there' to be applied, but is constructed dynamically, in the course of a legal procedure (e.g. [Hage *et al.*, 1994, Gordon, 1994, Bench-Capon, 1998, Lodder, 1999, Prakken, 2001c]). In contrast to the above-mentioned work on dispute in the 'design' strand, here the focus is more on procedure and less on persuasive argument moves, i.e., more on the rules of the 'debating game' and less on how to play this game well.

In this survey we will discuss not only logical approaches but also some work from the 'design strand'. This is since, in our opinion, these approaches should not be regarded as alternatives but should complement and inspire each other. A purely logic-based approach runs the risk of becoming too abstract and ignored by the field for which it is intended, while a purely design-based approach is in danger of becoming too self-centred and ad-hoc.

1.2 A four-layered view on legal argument

How can all these research projects be compared and contrasted? We propose that models of legal argument can be described in terms of four layers.[1] The first, *logical* layer defines what arguments are, i.e., how pieces of information

[1] The combination of the first three layers was first discussed by [Prakken, 1995]. The first and third layer were also discussed by [Brewka and Gordon, 1994]. The fourth layer was added by [Prakken, 1997] and also discussed in [Sartor, 1997].

can be combined to provide basic support for a claim. The second, *dialectical* layer focuses on conflicting arguments: it introduces such notions as 'counterargument', 'attack', 'rebuttal' and 'defeat', and it defines, given a set of arguments and evaluation criteria, which arguments prevail. The third, *procedural* layer regulates how an actual dispute can be conducted, i.e., how parties can introduce or challenge new information and state new arguments. In other words, this level defines the possible speech acts, and the discourse rules governing them. Thus the procedural layer differs from the first two in one crucial respect. While those layers assume a fixed set of premises, at the procedural layer the set of premises is constructed dynamically, during a debate. This also holds for the final layer, the *strategic* or *heuristic* one, which provides rational ways of conducting a dispute within the procedural bounds of the third layer.

All four layers are to be integrated into a comprehensive view of argumentation: the logical layer defines, by providing a notion of arguments, the objects to be evaluated at the dialectical layer; the dialectical layer offers to the procedural and heuristic layers a judgement of whether a new argument might be relevant in the dispute; the procedural layer constrains the ways in which new inputs, supplied by the heuristic layer can be submitted to the dialectical one; the heuristic layer provides the matter which is to be processed in the system. Each layer can obviously be studied (and implemented) in abstraction from the other ones. However, a main premise of this article is that research at the individual levels would benefit if the connection with the other layers is always kept in mind. For instance, logical techniques (whether monotonic or not) have a better chance of being accepted by the AI & Law community when they can easily be embedded in procedural or heuristic layers of legal argument.

Let us illustrate the four layers with an example of a legal dispute.

P_1: I claim that John is guilty of murder.
O_1: I deny your claim.
P_2: John's fingerprints were on the knife.
If someone stabs a person to death, his fingerprints must be on the knife,
So, John has stabbed Bill to death.
If a person stabs someone to death, he is guilty of murder,
So, John is guilty of murder.
O_2: I concede your premises, but I disagree that they imply your claim:
Witness X says that John had pulled the knife out of the dead body.
This explains why his fingerprints were on the knife.
P_3 X's testimony is inadmissible evidence, since she is anonymous.
Therefore, my claim still stands.

P_1 illustrates the procedural layer: the proponent of a claim starts a dispute by stating his claim. The procedure now says that the opponent can either accept or deny this claim. O does the latter with O_1. The procedure now assigns the burden of proof to P. P attempts to fulfil this burden with an argument for his claim (P_2). Note that this argument is not deductive since it includes an abductive inference step; whether it is constructible, is determined at the

logical layer. The same holds for O's counterargument O_2, but whether it is a counterargument and has sufficient attacking strength is determined at the dialectical layer, while O's right to state a counterargument is defined by the procedure. The same remarks hold for P's counterargument P_3. In addition, P_3 illustrates the heuristic layer: it uses the heuristic that evidence can be attacked by arguing that it is inadmissible.

This paper is organised as follows. First, in Section 2 we discuss the four layers in more detail. Then in Section 3, we use them in discussing the most influential computational models of legal argument. In Section 4, we do the same for the main logical analyses of legal argument, after which we conclude.

2 Four layers in legal argument

Let us now look in more detail at the four layers of legal argument. It is important to note that the first two layers comprise the subject matter of nonmonotonic logics. One type of such logics explicitly separates the two layers, viz. logical systems for defeasible argumentation (cf. [Prakken and Vreeswijk, 2001]). For this reason we will largely base our discussions on the structure of these systems. However, since [Dung, 1995] and [Bondarenko *et al.*, 1997] have shown that essentially all nonmonotonic logics can be recast as such argument-based systems, most of what we will say also applies to other nonmonotonic logics.

2.1 The logical layer

The logical layer is concerned with the language in which information can be expressed, and with the rules for constructing arguments in this language.

The logical language

Deontic terms One ongoing debate in AI & Law is whether normative terms such as 'obligatory', 'permitted' and 'forbidden' should be formalised in (modal) deontic logics or whether they can be expressed in first-order logic; cf. e.g. [Jones and Sergot, 1992]. From our perspective this issue is not very relevant, since logics for defeasible argumentation can cope with any underlying logic. Moreover, as for the defeasibility of deontic reasoning, we think that special *deontic* defeasible logics (see e.g. [Nute, 1997]) are not very suited. It is better to embed one's preferred deontic monotonic logic in one's preferred general defeasible logic, since legal defeasibility is not restricted to deontic terms, but extends to all other kinds of legal knowledge, including definitions and evidential knowledge. Obviously, a unified treatment of defeasibility is to be preferred; cf. [Prakken, 1996].

Conceptual structures Others have focused on the formalisation of recurring conceptual legal structures. Important work in this area is McCarty's[1989] *Language of Legal Discourse*, which addresses the representation of such categories as space, time, mass, action, causation, intention, knowledge, and belief. This

strand of work is, although very important for AI & Law, less relevant for our concerns, for the same reasons as in the deontic case: argument-based systems can deal with any underlying logic.

Conditional rules A topic that is more relevant for our concerns is the representation of conditional legal rules. The main issue here is whether legal rules satisfy contrapositive properties or not. Some AI & Law formalisms, e.g. Gordon's [1995] Pleadings Game, validate contraposition. However, [Prakken, 1997] has argued that contraposition makes counterarguments possible that would never be considered in actual reasoning practice. A possible explanation for why this is the case is Hage's [1996, 1997] view on legal rules as being constitutive. In this view (based on insights of analytical philosophy) a legal rule does not describe but constitutes states of affairs: for instance, a legal rule *makes* someone a thief or something a contract, it does not *describe* that this is the case. According to Hage, a legal rule must be *applied* to make things the case, and lawyers never apply rules contrapositively. This view is related to AI interpretations of defaults as *inference licences* or *inference policies* [Loui, 1998, Nute, 1992], while the invalidity of contraposition has also been defended in the context of causal reasoning; see e.g. [Geffner, 1992]. Finally, contraposition is also invalid in extended logic programming, where programs can have both weak and strong negations; cf. [Gelfond and Lifschitz, 1990].

Weak and strong negation The desire to formalise reasoning with rules and exceptions sometimes motivates the use of a nonprovability, consistency or weak negation operator, such as negation as failure in logic programming. Whether such a device should be used depends on one's particular convention for formalising rules and exceptions (see further Section 2.2 below).

Metalogic features Much legal knowledge is metaknowledge, for instance, knowledge about the general validity of rules or their applicability to certain kinds of cases, priority principles for resolving conflicts between conflicting rules, or principles for interpreting legal rules. Clearly, for representing such knowledge metalogic tools are needed. Logic-based AI & Law research of legal argument has made ample use of such tools, as this survey will illustrate.

Non-logical languages Finally, non-logical languages can be used. On the one hand, there are the well-known knowledge representation formalisms, such as frames and semantic networks. In AI, their logical interpretation has been thoroughly studied. On the other hand, in AI & Law various special-purpose schemes have been developed, such as HYPO's factor-based representation of cases (see Section 3.3), ZENO's issue-position-based language [Gordon and Karaçapilidis, 1997], Room 5's encapsulated text frames [Loui *et al.*, 1997], ArguMed's linked-boxes language [Verheij, 1999], or variants of Toulmin's [1958] well-known argument scheme [Bench-Capon, 1998]. Simple non-logical languages are especially convenient in systems for intelligent tutoring (such as CATO) or argument mediation (such as ROOM 5, ZENO and ArguMed), since users of such systems

cannot be expected to formalise their arguments in logic. In formally reconstructing such systems, one issue is whether their representation language should be taken as primitive or translated into some known logical language. Argument-based logics leave room for both options.

Argument construction As for argument construction, a minor issue is how to format arguments: as simple premises - conclusion pairs, as sequences of inferences (deductions) or as trees of inferences. The choice between these options seems a matter of convenience; for a discussion of the various options see e.g. [Prakken and Vreeswijk, 2001]. More crucial issues are whether incomplete arguments, i.e., arguments with hidden premises, should be allowed and whether nondeductive arguments should be allowed.

Incomplete arguments In ordinary language people very often omit information that could make their arguments valid, such as in "John has killed Pete, so John is guilty of Murder". Here the hidden premise "Who kills another person is guilty of murder" is omitted. In some argument mediation applications, e.g. [Lodder, 1999], such incomplete arguments have been allowed, for instance, to give the listener the opportunity to agree with the argument, so that obvious things can be dealt with efficiently. In our opinion this makes sense, but only if a listener who does not agree with the argument has a way to challenge its validity.

Non-deductive argument types Non-deductive reasoning forms, such as inductive, abductive and analogical reasoning are clearly essential to any form of practical reasoning, so they must have a place in the four-layered view on argumentation. In legal reasoning inductive and abductive arguments play an important role in evidential reasoning, while analogical arguments are especially important in the interpretation of legal concepts.

The main issue is whether these reasoning forms should be regarded as argument construction principles (the logical layer) or as heuristics for finding new information (the heuristic layer). In [Prakken, 1995], one of us argued for the latter option. For instance, Prakken argued that an analogy is inherently unable to justify its conclusion since in the end it must always be decided whether the similarities outweigh the differences or not. However, others, e.g. [Loui *et al.*, 1993, Loui, 1998], have included analogical arguments at the logical layer on the grounds that if they are untenable, this will show itself in a rational dispute. Clearly, the latter view presupposes that the dialectical layer is embedded in the procedural layer. For a legal-theoretical discussion of the issue see [Peczenik, 1996, pp. 310–313]. Outside AI & Law, a prominent argument-based system that admits non-deductive arguments is [Pollock, 1995]'s OSCAR system.

Our present opinion is that both approaches make sense. One important factor here is whether the dialectical layer is embedded in the procedural layer. Another important factor is whether a reasoning form is used to justify a conclusion or not. For instance, some uses of analogy concern learning [Winston, 1980], while other uses concern justification (as in much AI & Law work on case-based

reasoning). One thing is especially important: if non-deductive arguments are admitted at the logical layer, then the dialectical layer should provide for ways to attack the link between their premises and conclusion; cf. Pollock's [1995] undercutters of defeasible inference rules. For instance, if analogies are admitted, it should not only be possible to rebut them with counterexamples, i.e., with analogies for contradictory conclusions, but it should also be possible to undercut analogies by saying that the similarities are irrelevant, or that the differences are more important than the similarities.

2.2 The dialectical layer

The dialectical layer addresses three issues: when arguments are in conflict, how conflicting arguments can be compared, and which arguments survive the competition between all conflicting arguments.

Conflict In the literature, three types of conflicts between arguments are discussed. The first is when arguments have contradictory conclusions, as in 'A contract exists because there was an offer and an acceptance' and 'A contract does not exist because the offerer was insane when making the offer'. Clearly, this form of attack, often called *rebutting* an argument, is symmetric. The other two types of conflict are not symmetric. One is where one argument makes a non-provability assumption (e.g. with logic-programming's negation as failure) and another argument proves what was assumed unprovable by the first. For example, an argument 'A contract exists because there was an offer and an acceptance, and it is not provable that one of the parties was insane', is attacked by any argument with conclusion 'The offerer was insane'. In [Prakken and Vreeswijk, 2001] this is called *assumption attack*. The final type of conflict (identified by Pollock, e.g. 1995) is when one argument challenges a rule of inference of another argument. After Pollock, this is usually called *undercutting* an inference. Obviously, a rule of inference can only be undercut if it is not deductive. For example, an analogy can be undercut by saying that the similarity is insufficient to warrant the same conclusion. Note, finally, that all these senses of attack have a direct and an indirect version; indirect attack is directed against a subconclusion or a substep of an argument. For instance, indirect rebuttals contradict an intermediate conclusion of an argument.

Comparing arguments The notion of conflicting, or attacking arguments does not embody any form of evaluation; comparing conflicting pairs of arguments, or in other words, determining whether an attack is successful, is another element of argumentation. The terminology varies: some terms that have been used are 'defeat' [Prakken and Sartor, 1996], 'attack' [Dung, 1995, Bondarenko *et al.*, 1997] and 'interference' [Loui, 1998]. In this article we shall use *defeat* for the weak notion and *strict defeat* for the strong, asymmetric notion.

How are conflicting arguments compared in the legal domain? Two main points must be stressed here. The first is that general, domain-independent standards are of little use. Lawyers use many domain-specific standards, ranging from

general principles such as "the superior law overrides the inferior law" and "the later regulation overrides the earlier one" to case-specific and context-dependent criteria such as "preferring this rule promotes economic competition, which is good for society", or "following this argument would lead to an enormous increase in litigation, which should be avoided". The second main point is that these standards often conflict, so that the comparison of conflicting arguments is itself a subject of dispute. For instance, the standards of legal certainty and individual fairness often conflict in concrete situations. For logical models of legal argument this means that priority principles must be expressible in the logical language, and that their application must be modelled as defeasible reasoning.

Specificity Some special remarks are in order about the specificity principle. In AI this principle is often regarded as very important. However, in legal reasoning it is just one of the many standards that might be used, and it is often overridden by other standards. Moreover, there are reasons to doubt whether specificity of regulations can be syntactically defined at all. Consider the following imaginary example (due to Marek Sergot, personal communication).

1. All cows must have earmarks
2. Calfs need not have earmarks
3. All cows must have earmarks, whether calf or not
4. All calfs are cows

Lawyers would regard (2) as an exception to (1) because of (4) but they would certainly not regard (2) as an exception to (3), since the formulation of (3) already takes the possible exception into account. Yet logically (3) is equivalent to (1), since the addition "whether calf or not" is a tautology. In conclusion, specificity may be suitable as a notational convention for exceptions, but it cannot serve as a domain-independent conflict resolution principle.

Assessing the status of arguments The notion of defeat only tells us something about the relative strength of two individual conflicting arguments; it does not yet tell us with what arguments a dispute can be won. The ultimate status of an argument depends on the interaction between all available arguments. An important phenomenon here is *reinstatement*:[2] it may very well be that argument B defeats argument A, but that B is itself defeated by a third argument C; in that case C 'reinstates' A. Suppose, for instance, that the argument A that a contract exists because there there was an offer and acceptance, is defeated by the argument B that a contract does not exist because the offerer was insane when making the offer. And suppose that B is in turn (strictly) defeated by an argument C, attacking B's intermediate conclusion that the offerer was insane at the time of the offer. In that case C reinstates argument A.

The main distinction is that between *justified*, *defensible* and *overruled* arguments. The distinction between justified and defensible arguments corresponds

[2] But see [Horty, 2001] for a critical analysis of the notion of reinstatement.

to the well-known distinction between sceptical and credulous reasoning, while overruled arguments are those that are defeated by a justified argument. Several ways to define these notions have been studied, both in semantic and in proof-theoretic form, and both for justification and for defensibility. See [Prakken and Vreeswijk, 2001] for an overview and especially [Dung, 1995, Bondarenko *et al.*, 1997] for semantical studies. For present purposes the differences in semantics do not matter much; what is more important is that argument-based proof theories can be stated in the dialectical form of an argument game, as a dispute between a *proponent* and *opponent* of a claim. The proponent starts with an argument for this claim, after which each player must attack the other player's previous argument with a counterargument of sufficient strength. The initial argument *provably* has a certain status if the proponent has a *winning strategy*, i.e., if he can make the opponent run out of moves in whatever way she attacks. Clearly, this setup fits well with the adversarial nature of legal argument, which makes it easy to embed the dialectical layer in the procedural and heuristic ones.

To give an example, consider the two dialogue trees of in Figure 1. Assume that they contain all constructible arguments and that the defeat relations are as shown by the arrows (single arrows denote strict defeat while double arrows stand for mutual defeat). In the tree on the left the proponent has a winning strategy, since in all dialogues the opponent eventually runs out of moves; so argument A is provable. The tree on the right extends the first tree with three arguments. Here the proponent does not have a winning strategy, since one dialogue ends with a move by the opponent; so A is not provable in the extended theory.

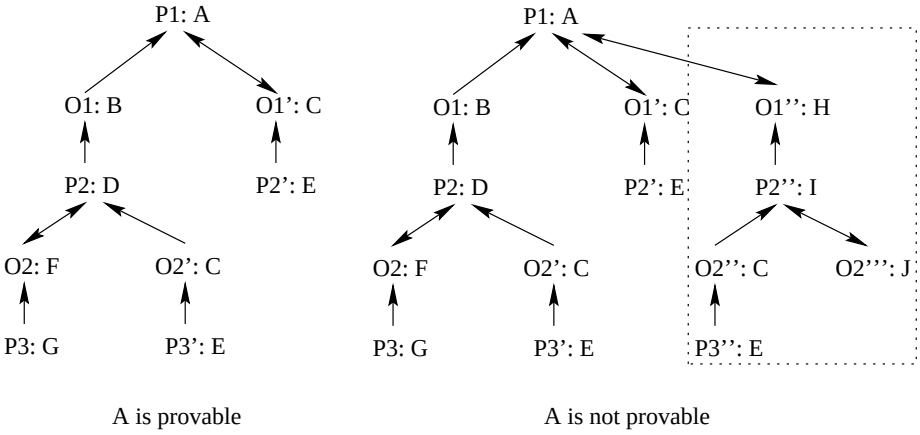

Fig. 1. Two trees of proof-theoretical dialogues.

Partial computation Above we said that the status of an argument depends on its interaction with all available arguments. However, we did not specify what 'available' means. Clearly, the arguments processed by the dialectical proof the-

ory are based on input from the procedural layer, viz. on what has been said and assumed in a dispute. However, should only the actually stated arguments be taken into account, or also additional arguments that can be computed from the theory constructed during the dispute? And if the latter option is chosen, should *all* constructible arguments be considered, or only those that can be computed within given resource bounds? In the literature, all three options have been explored. The methods with partial and no computation have been defended by pointing at the fact that computer algorithms cannot be guaranteed to find arguments in reasonable time, and sometimes not even in finite time (see especially Pollock 1995; Loui 1998). In our opinion, the choice essentially depends on the context and the intended use of the system.

Representing exceptions Finally, we discuss the representation of exceptions to legal rules, which concerns a very common phenomenon in the law. Some exceptions are stated by statutes themselves, while others are based, for instance on the purpose of rules or on legal principles. Three different techniques have been used for dealing with exceptions. Two of them are well-known from nonmonotonic logic, while the third one is, to our knowledge, a contribution of AI & Law research.

The first general technique is the *exception clause* or *explicit-exceptions approach*, which corresponds to the use of 'unless' clauses in natural language. Logically, such clauses are captured by a nonprovability operator, which can be formalised with various well-known techniques from nonmonotonic logic or logic programming. In argument-based models the idea is that arguments concluding for the exception, thus establishing what the rule requires not to be proved, defeat arguments based upon the rule. In some formalisations, the not-to-be-proved exception is directly included in the antecedent of the rule to which it refers. So, the rule 'A if B, unless C', is (semiformally) represented as follows (where $\sim$ stands for nonprovability).

r_1: $A \wedge \sim C \Rightarrow B$

A more abstract and modular representation is also possible within the exception clause approach. This is achieved when the rule is formulated as requiring that no exception is proved to the rule itself. The exception now becomes the antecedent of a separate conditional.

r_1: $A \wedge \sim Exc(r_1) \Rightarrow B$
r_2: $C \Rightarrow Exc(r_1)$

While in this approach rules themselves refer to their exceptions, a variant of this technique has been developed where instead the no-exception requirement is built into the logic of rule application [Routen and Bench-Capon, 1991, Hage, 1996, Prakken and Sartor, 1996]. Semiformally this looks as follows.

r_1: $A \Rightarrow B$
r_2: $C \Rightarrow Exc(r_1)$

We shall call this the *exclusion approach*. In argument-based versions it takes the form of allowing arguments for the inapplicability of a rule defeat the arguments using that rule. Exclusion resembles Pollock's [1995] notion of undercutting defeaters.

Finally, a third technique for representing exceptions is provided by the *choice* or *implicit-exceptions approach*. As in the exclusion approach, rules do not explicitly refer to exceptions. However, unlike with exclusion, the exception is not explicitly stated as an exception. Rather it is stated as a rule with conflicting conclusion, and is turned into an exception by preference information that gives the exceptional rule priority over the general rule.

r_1: $A \Rightarrow B$
r_2: $C \Rightarrow \neg B$
$r_1 < r_2$

In argument-based models this approach is implemented by making arguments based on stronger rules defeat arguments based on weaker rules.

In the general study of nonmonotonic reasoning usually either only the exception-clause- or only the choice approach is followed. However, AI & Law researchers have stressed that models of legal argument should support the combined use of all three techniques, since the law itself uses all three of them.

2.3 The procedural layer

There is a growing awareness that there are other grounds for the acceptability of arguments besides syntactic and semantic grounds. One class of such grounds lies in the way in which a conclusion was actually reached. This is partly inspired by a philosophical tradition that emphasises the procedural side of rationality and justice; see e.g. [Toulmin, 1958, Rawls, 1972, Rescher, 1977, Habermas, 1981].

Particularly relevant for present purposes is Toulmin's [1958, pp. 7–8] advice that logicians who want to learn about reasoning in practice, should turn away from mathematics and instead study jurisprudence, since outside mathematics the validity of arguments would not depend on their syntactic form but on the disputational process in which they have been defended. According to Toulmin an argument is valid if it can stand against criticism in a properly conducted dispute, and the task of logicians is to find criteria for when a dispute has been conducted properly; moreover, he thinks that the law, with its emphasis on procedures, is an excellent place to find such criteria.

Toulmin himself has not carried out his suggestion, but others have. For instance, Rescher [1977] has sketched a dialectical model of scientific reasoning which, so he claims, explains the bindingness of inductive arguments: they must be accepted if they cannot be successfully challenged in a properly conducted scientific dispute. A formal reconstruction of Rescher's model has been given by Brewka [1994]. In legal philosophy Alexy's [1978] discourse theory of legal argumentation addresses Toulmin's concerns, based on the view that a legal decision is just if it is the outcome of a fair procedure.

Another source of the concern for procedure is AI research on resource-bounded reasoning; e.g. [Simon, 1982, Pollock, 1995, Loui, 1998]. When the available resources do not guarantee finding an optimal solution, rational reasoners have to rely on effective procedures. One kind of procedure that has been advocated as effective is dialectics [Rescher, 1977, Loui, 1998].

It is not necessary to accept the view that rationality is essentially procedural in order to see that it at least has a procedural side. Therefore, a study of procedure is of interest to anyone concerned with normative theories of reasoning.

How can formal models of legal procedure be developed? Fortunately, there already exists a formal framework that can be used. In argumentation theory, formal dialogue systems have been developed for so-called 'persuasion' or 'critical discussion'; see e.g. [Hamblin, 1971, MacKenzie, 1990, Walton and Krabbe, 1995]. According to Walton and Krabbe [1995], dialogue systems regulate four aspects of dialogues:

- *Location rules* (what moves are possible)
- *Structural rules* (when moves are legal)
- *Commitment rules* (The effects of moves on the players' commitments);
- *Termination rules* (when dialogues terminate and with what outcome).

In persuasion, the parties in a dispute try to solve a conflict of opinion by verbal means. The dialogue systems regulate the use of speech acts for such things as making, challenging, accepting, withdrawing, and arguing for a claim. The proponent of a claim aims at making the opponent concede his claim; the opponent instead aims at making the proponent withdraw his claim. A persuasion dialogue ends when one of the players has fulfilled their aim. Logic governs the dialogue in various ways. For instance, if a participant is asked to give grounds for a claim, then in most systems these grounds have to logically imply the claim. Or if a proponent's claim is logically implied by the opponent's concessions, the opponent is forced to accept the claim, or else withdraw some of her concessions.

Most computational models of legal procedure developed so far [Hage *et al.*, 1994, Gordon, 1995, Bench-Capon, 1998, Lodder, 1999, Prakken, 2001c] have incorporated such formal dialogue systems. However, they have extended them with one interesting feature, viz. the possibility of counterargument. In argumentation-theoretic models of persuasion the only way to challenge an argument is by asking an argument for its premises. In a legal dialogue, by contrast, a party can challenge an argument even if he accepts all premises, viz. by stating a counterargument. In other words, while in the argumentation-theoretic models the underlying logic is deductive, in the AI & Law systems it is defeasible: support for a claim may be defeasible (e.g. inductive or analogical) instead of watertight, and forced or implied concession of a claim is defined in terms of defeasible instead of deductive consequence. Or in terms of our four-layered view: while the argumentation theorists only have the logical and procedural layer, the AI & Law models have added the dialectical layer in between.

In fact, clarifying the interplay between the dialectical and the procedural layer is not a trivial matter, and is the subject of ongoing logical research. See e.g. [Brewka, 2001, Prakken, 2000, Prakken, 2001d].

2.4 The heuristic layer

This layer (which addresses much of what is traditionally called 'rhetoric') is the most diverse one. In fact, heuristics play a role at any aspect of the other three levels: they say which premises to use, which arguments to construct, how to present them, which arguments to attack, which claims to make, concede or deny, etc. Heuristics can be divided into (at least) three kinds: *inventional heuristics*, which say how a theory can be formed (such as the classical interpretation schemes for legal rules), *selection* heuristics, which recommend a choice between various options (such as 'choose an argument with as few premises as possible, to minimise its attacking points'), and *presentation heuristics*, which tell how to present an argument (e.g. 'don't draw the conclusion yourself but invite the listener to draw it').

A keyword at the heuristic level is *persuasion*. For instance, which arguments are the most likely to make the opponent accept one's claims? Persuasiveness of arguments is not a matter of logic, however broadly conceived. Persuasiveness is not a function from a given body of information: it involves an essential nondeterministic element, viz. what the other player(s) will do in response to a player's dialectic acts. To model persuasiveness, models are needed predicting what other players (perhaps the normal, typical other player) will do. Analogous models have been studied in research on argument in negotiation [Kraus *et al.*, 1998, Parsons *et al.*, 1998].

An interesting issue is how to draw the dividing line between argument formation rules and inventional heuristics. Below we will discuss several reasoning schemes that can be reasonably regarded as of either type. We think that the criterion is whether the schemes are meant to *justify* a claim or not.

2.5 Intertwining of the layers

The four layers can be intertwined in several ways. For instance, allocating the burden of proof is a procedural matter, usually done by the judge on the basis of procedural law. However, sometimes it becomes the subject of dispute, for instance, when the relevant procedural provisions are open-textured or ambiguous. In such a case, the judge will consider all relevant arguments for and against a certain allocation and decide which argument prevails. To this the dialectical layer applies. The result, a justified argument concerning a certain allocation, is then transferred to the procedural layer as a decision concerning the allocation.

Moreover, sometimes the question at which layer one finds himself depends on the use that is made of a reasoning scheme instead of on the reasoning scheme itself. We already mentioned analogy, which can be used in learning (heuristic layer) but also in justification (dialectical layer). Or consider, for another example, the so-called teleological interpretation scheme, i.e., the idea that law texts should usually be understood in terms of their underlying purposes. This principle may be used by a party (when it provides him with a rule which is in his interest to state) as an inventional heuristic, i.e., as a device suggesting suitable contents to be stated in his argument: interpret a law text as a rule which

achieves the legislator's purposes, whenever this rule promotes your interest. If this is the use of the interpretation scheme, then a party would not input it in the dispute, but would just state the results it suggests. The principle, however, could also be viewed by a party as a justificatory premise, which the party explicitly uses to support the conclusion that a certain rule is valid, or that it prevails over alternative interpretations.

Not all inventional heuristics could equally be translatable as justificatory meta-rules. Consider for example the heuristic: interpret a text as expressing the rule that best matches the political ideology (or the sexual of racial prejudices) of the judge of your case, if this rule promotes your interest. This suggestion, even though it may be a successful heuristic, usually could not be inputted in the argumentation as a justificatory meta-rule.

3 Computational models of legal argument

In the introduction we said that logic-based and design-based methods in AI & law should complement and influence each other. For this reason, we now discuss some of the most influential implemented architectures of legal argument. We do so in the light of our four-layered view.

3.1 McCarty's work

The TAXMAN II project of McCarty (e.g. McCarty and Sridharan, 1981; McCarty, 1995) aims to model how lawyers argue for or against the application of a legal concept to a problem situation. In McCarty and Sridharan [1981] only a theoretical model is presented but in McCarty [1995] an implementation is described of most components of the model. However, their interaction in finding arguments is still controlled by the user.

Among other things, the project involves the design of a method for representing legal concepts, capturing their open-textured and dynamic nature. This method is based on the view that legal concepts have three components: firstly, a (possibly empty) set of necessary conditions for the concept's applicability; secondly, a set of instances ("exemplars") of the concept; and finally, a set of rules for transforming a case into another one, particularly for relating "prototypical" exemplars to "deformations". According to McCarty, the way lawyers typically argue about application of a concept to a new case is by finding a plausible sequence of transformations which maps a prototype, possibly via other cases, onto the new case. In our opinion, these transformations might be regarded as invention heuristics for argument construction.

3.2 Gardner

An early success of logic-based methods in AI & Law was their logical reconstruction of Gardner's [1987] program for so-called "issue spotting". Given an

input case, the task of the program was to determine which legal questions involved were easy and which were hard, and to solve the easy ones. If all the questions were found easy, the program reported the case as clear, otherwise as hard. The system contained domain knowledge of three different types: legal rules, common-sense rules, and rules extracted from cases. The program considered a question as hard if either "the rules run out", or different rules or cases point at different solutions, without there being any reason to prefer one over the other. Before a case was reported as hard, conflicting alternatives were compared to check whether one is preferred over the other. For example, case law sets aside legal rules or common-sense interpretations of legal concepts.

Clearly, Gardner's program can be reconstructed as nonmonotonic reasoning with prioritised information, i.e., as addressing the dialectical layer. Reconstructions of this kind have been given by [Gordon, 1991], adapting [Poole, 1988]'s abductive model of default reasoning, and [Prakken, 1997], in terms of an argument-based logic.

3.3 HYPO

HYPO aims to model how lawyers make use of past decisions when arguing a case. The system generates 3-ply disputes between a plaintiff and a defendant of a legal claim concerning misuse of a trade secret. Each move conforms to certain rules for analogising and distinguishing precedents. These rules determine for each side which are the best cases to cite initially, or in response to the counterparty's move, and how the counterparty's cases can be distinguished. A case is represented as a set of factors pushing the case towards (pro) or against (con) a certain decision, plus a decision which resolves the conflict between the competing factors. A case is citable for a side if it has the decision wished by that side and shares with the Current Fact Situation (CFS) at least one factor which favours that decision. A citation can be countered by a counterexample, that is, a case that is at least as much on point, but has the opposite outcome. A citation may also be countered by distinguishing, that is, by indicating a factor in the CFS which is absent in the cited precedent and which supports the opposite outcome, or a factor in the precedent which is missing in the CFS, and which supports the outcome of the cited case. Finally, HYPO can create hypothetical cases by using magnitudes of factors. In evaluating the relative force of the moves, HYPO uses the set inclusion ordering on the factors that the precedents share with the CFS. However, unlike logic-based argumentation systems, HYPO does not compute an 'outcome' or 'winner' of a dispute; instead it outputs 3-ply disputes as they could take place between 'good' lawyers.

HYPO in terms of the four layers Interpreting HYPO in terms of the four layers, the main choice is whether to model HYPO's analogising and distinguishing moves as argument formation rules (logical layer) or as inventional heuristics (heuristic layer). In the first interpretation, the representation language is simply as described above (a decision, and sets of factors pro and con a decision),

analogising a precedent is a constructible argument, stating a counterexample is a rebutter, and distinguishing a precedent is an undercutter. Defeat is defined such that distinctions always defeat their targets, while counterarguments defeat their targets iff they are not less on point. In the second interpretation, proposed by [Prakken and Sartor, 1998], analogising and distinguishing a precedent are regarded as 'theory constructors', i.e., as ways of introducing new information into a dispute. We shall discuss this proposal below in Section 4.2.

Which interpretation of HYPO's argument moves is the best one is not an easy question. Essentially it asks for the nature of analogical reasoning, which is a deep philosophical question. In both interpretations HYPO has some heuristic aspects, since it defines the "best cases to cite" for each party, selecting the most-on-point cases from those allowed by the dialectical protocol. This can be regarded as a selection heuristic.

3.4 CATO

The CATO system of Aleven and Ashley [1997] applies an extended HYPO architecture for teaching case-based argumentation skills to law students, also in the trade secrets domain. CATO's main new component is a 'factor hierarchy', which expresses expert knowledge about the relations between the various factors: more concrete factors are classified according to whether they are a reason pro or con the more abstract factors they are linked to; links are given a strength (weak or strong), which can be used to solve certain conflicts. Essentially, this hierarchy fills the space between the factors and decision of a case. Thus it can be used to explain why a certain decision was taken, which in turn facilitates debates on the relevance of differences between cases.

For instance, the hierarchy positively links the factor *Security measures taken* to the more abstract concept *Efforts to maintain secrecy*. Now if a precedent contains the first factor but the CFS lacks it, then not only could a citation of the precedent be distinguished on the absence of *Security measures taken*, but also could this distinction be emphasised by saying that thus no efforts were made to maintain secrecy. However, if the CFS also contains a factor *Agreed not to disclose information*, then the factor hierarchy enables downplaying this distinction, since it also positively links this factor to *Efforts to maintain secrecy*: so the party that cited the precedent can say that in the current case, just as in the precedent, efforts were made to maintain secrecy.

The factor hierarchy is not meant to be an independent source of information from which arguments can be constructed. Rather it serves as a means to *reinterpret* precedents: initially cases are in CATO, as in HYPO, still represented as one-step decisions; the factor hierarchy can only be used to argue that the decision was in fact reached by one or more intermediate steps.

CATO in terms of the four layers At the logical layer CATO adds to HYPO the generation of multi-steps arguments, exploiting the factor hierarchy. As for CATO's ability to reinterpret precedents, we do not regard this as an inventional

heuristic, since the main device used in this feature, the factor hierarchy, is given in advance; instead we think that this is just the logic-layer ability to build multi-steps arguments from given information. However, CATO's way of formatting the emphasising and downplaying moves in its output can be regarded as built-in presentation heuristics.

3.5 CABARET

The CABARET system of Rissland and Skalak [1991] attempts at combining rule-based and case-based reasoning. Its case-based component is the HYPO system. The focus is on statutory interpretation, in particular on using precedents to confirm or contest the application of a rule. In [Skalak and Rissland, 1992], CABARET's model is described as a hierarchy of argument techniques including strategies, moves and primitives. A strategy is a broad characterisation of how one should argue, given one's particular viewpoint and dialectical situation. A move is a way to carry out the strategy, while a primitive is a way to implement a move. For example, when one wants to apply a rule, and not all of the rule's conditions are satisfied, then a possible strategy is to broaden the rule. This strategy can be implemented with a move that argues with an analogised precedent that the missing condition is not really necessary. This move can in turn be implemented with HYPO's ways to analogise cases. Similarly, CABARET also permits arguments that a rule which *prima facie* appears to cover the case, should not be applied to it. Here the strategy is *discrediting a rule* and the move may consist in analogising a case in which the rule's conditions were met but the rule was not applied. Again the move can be implemented with HYPO's ways to analogise cases.

CABARET in terms of the four layers At the logical layer CABARET adds to HYPO the possibility to construct simple rule-based arguments, while at the dialectical layer, CABARET adds corresponding ways to attack arguments. CABARET's main feature, its model of argument strategies, clearly addresses the heuristic layer. The strategies can be seen as selection heuristics: they choose between the available attacking points, and pick up from the rule- and case-base the most relevant materials.

3.6 DART

Freeman & Farley [1996] have semi-formally described and implemented a dialectical model of argumentation. For legal applications it is especially relevant since it addresses the issue of burden of proof. Rules are divided into three epistemic categories: 'sufficient', 'evidential' and 'default', in decreasing order of priority. The rules for constructing arguments involve standard logic principles, such as modus ponens and modus tollens, but also nonstandard ones, such as for abductive reasoning ($p \Rightarrow q$ and q imply p) and a contrario reasoning ($p \Rightarrow q$ and $\neg p$ imply $\neg q$). Taken by themselves these inferences clearly are the well-known

fallacies of 'affirming the consequent' and 'denying the antecedent' but this is dealt with by defining undercutters for such arguments. For instance, the above abductive argument can be undercut by providing an alternative explanation for q, in the form of a rule $r \Rightarrow q$.

The defeat relations between arguments depend both on the type of premise and on the type of inference rule. The status of arguments is defined in terms of an argument game based on a static knowledge base. DART's argument game has several variants, depending on which level of proof holds for the main claim. This is because Freeman and Farley maintain that different legal problem solving contexts require different levels of proof. For instance, for the question whether a case can be brought before court, only a 'scintilla of evidence' is required (in present terms a defensible argument), while for a decision in a case 'dialectical validity' is needed (in our terms a justified argument).

DART in terms of the four layers DART essentially addresses the logical and dialectical layers, while it assumes input from the procedural layer. At the logical layer, it allows both deductive and nondeductive arguments. Freeman and Farley are well aware that this requires the definition of undercutters for the nondeductive argument types. DART's argument games are similar to dialectical proof theories for argument-based logics. However, they are not given a formal semantics. Finally, DART assumes procedural input in the form of an assignment of a level of proof to the main claim.

3.7 The Pleadings Game

Next we discuss Gordon's [1994, 1995] Pleadings Game, which is an attempt to model the procedural view on justice discussed above in Section 2.3. The legal-procedural example domain is 'civil pleading', which is the phase in Anglo-American civil procedure where the parties exchange arguments and counterarguments to identify the issues that must be decided by the court. The system is not only implemented but also formally defined. Thus this work is an excellent illustration of how logic can be used as a tool in computational models of legal argument. For this reason, and also since it clearly illustrates the relation between the first three layers, we shall discuss it in some detail.

The implemented system mediates between parties in a legal procedure: it keeps track of the stated arguments and their dialectical relations, and it checks whether the procedure is obeyed. Gordon models civil pleading as a Hamblin-MacKenzie-style dialogue game, defining speech acts for stating, conceding and denying (= challenging) a claim, and stating an argument for a claim. In addition, Gordon allows for counterarguments, thus choosing for a nonmonotonic logic as the underlying logical system. In fact, Gordon uses the argument-based proof theory of Geffner's [1992] conditional entailment.

As for the structural rules of the game, a game starts when the plaintiff states his main claim. Then the game is governed by a general rule saying that at each turn a player must respond in some permissible way to every move of

the opponent that is still relevant. A move is relevant iff it concerns an issue. An issue is, very roughly, defined as a claim that dialectically matters for the main claim and has not yet been replied-to.

The other structural rules define under which conditions a move is permissible. For instance, a claim of a player may be denied by the other player iff it is an issue and is not defeasibly implied by the denier's own previous claims. And a denied claim may be defended with an argument as long as (roughly) the claim is an issue, and the argument's premises are consistent with the mover's previous claims, and (in case the other party had previously claimed them) they were conceded by the mover. If no such 'permission rule' applies, the other player is to move, except when this situation occurs at the beginning of a turn, in which case the game terminates.

The result of a terminated game is twofold: a list of issues identified during the game (i.e., the claims on which the players disagree), and a winner, if there is one. Winning is defined relative to the set of premises agreed upon during a game. If issues remain, there is no winner and the case must be decided by the court. If no issues remain, then the plaintiff wins iff its main claim is defeasibly implied by the jointly constructed theory, while the defendant wins otherwise.

An Example We now illustrate the Pleadings Game with an example. Besides illustrating this particular system, the example also illustrates the interplay between the logical, dialectical and procedural layers of legal argument. For the sake of illustration we simplify the Game on several points, and use a different (and semiformal) notation. The example, loosely based on Dutch law, concerns a dispute on offer and acceptance of contracts. The players are called plaintiff (π) and defendant (δ). Plaintiff, who had made an offer to defendant, starts the game by claiming that a contract exists. Defendant denies this claim, after which plaintiff supports it with the argument that defendant accepted his offer and that an accepted offer creates a contract.

π_1: **Claim**[(1) `Contract`]
δ_1: **Deny**(1)
π_2: **Argue**[(2) `Offer`, (3) `Acceptance`,
(4) `Offer` $\wedge$ `Acceptance` $\Rightarrow$ `Contract`, so `Contract`]

Now defendant attacks plaintiff's supporting argument [2,3,4] by defeating its subargument that she accepted the offer. The counterargument says that defendant sent her accepting message after the offer had expired, for which reason there was no acceptance in a legal sense.

δ_2: **Concede**(2,4), **Deny**(3)
Argue[(5) "`Accept` `late`, (6) "`Accept` `late` $\Rightarrow$ $\neg$ `Acceptance`,
so $\neg$ `acceptance`]

Plaintiff responds by strictly defeating δ_2 with a more specific counterargument (conditional entailment compares arguments on specificity), saying that even

though defendant's accepting message was late, it still counts as an acceptance, since plaintiff had immediately sent a return message saying that he recognises defendant's message as an acceptance.

π_3: **Concede**(5), **Deny**(6),
Argue[(5) "`Accept` `late`, (7) "`Accept` `recognised`,
(8) "`Accept` `late` $\wedge$ "`Accept` `recognised` $\Rightarrow$ `Acceptance`,
so `Acceptance`]

Defendant now attempts to leave the issues for trial by conceding π_3's argument (the only effect of this is giving up the right to state a counterargument) and its premise (8), and by denying one of the other premises, viz. (7) (she had already implicitly claimed premise (5) herself, in δ_2). Plaintiff goes along with defendant's aim by simply denying defendant's denial of (7) and not stating a supporting argument for his claim, after which the game terminates since no relevant moves are left to answer for either party.

δ_3: **Concede**(8,[5,7,8]), **Deny**(7)
π_4: **Deny**(**Deny**(7))

This game has resulted in the following dialectical graph.

π_1: [2,3,4] for `Contract`
δ_1: [5,6] for $\neg$ `Acceptance`
π_2: [5,7,8] for `Acceptance`

The claims in this graph that have not been conceded are

(1) `Contract`
(3) `Acceptance`
(6) "`Accept` `late` $\Rightarrow$ $\neg$ `Acceptance`
(7) "`Accept` `recognised`

So these are the issues. Moreover, the set of premises constructed during the game, i.e. the set of conceded claims, is $\{2, 4, 5\}$. It is up to the judge whether to extend it with the issues (6) and (7). In each case conditional-entailment's proof theory must be used to verify whether the other two issues, in particular plaintiff's main claim (1), are (defeasibly) implied by the resulting premises. In fact, it is easy to see that they are entailed only if (6) and (7) are added.

The Pleadings Game in terms of the four layers Clearly, the Pleadings Game explicitly models the first three layers of our model. (In fact, the game was a source of inspiration of [Prakken, 1995]'s first formulation of these layers.) Its contribution to modelling the procedural layer should be apparent from the example. Gordon has also addressed the formalisation of the dialectical layer, adapting within conditional entailment well-known AI techniques concerning

naming of rules in (ab)normality predicates. Firstly, he has shown how information about properties of rules (such as validity and backing) can be expressed and, secondly, he has defined a way to express priority rules as object level rules, thus formalising disputes about rule priorities. However, a limitation of his method is that it has to accept conditional-entailment's built-in specificity principle as the highest source of priorities.

4 Logical models of legal argument

Having discussed several implemented models of legal argument, we now turn to logical models. Again we will discuss them in light of our four-layers model.

4.1 Applications of logic (meta-)programming

The British Nationality Act First we discuss the idea of formalising law as logic programs, viz. as a set of formulas of a logical language for which automated theorem provers exist. The underlying ideas of this approach are set out in [Sergot, 1988] and [Kowalski and Sergot, 1990], and is most closely associated with Sergot and Kowalski. The best known application is the formalisation of the British Nationality Act [Sergot *et al.*, 1986] (but see also [Bench-Capon *et al.*, 1987]). For present purposes the main relevance of the work of Sergot *et al.* is its treatment of exceptions by using negation by failure (further explored by Kowalski, 1989, 1995). To our knowledge, this was the first logical treatment of exceptions in a legal context.

In this approach, which implements the explicit-exceptions approach of Section 2.2, negation by failure is considered to be an appropriate translation for such locutions as 'unless the contrary is shown' or 'subject to section ...', which usually introduce exception clauses in legislation. Consider, for example, the norm to the effect that, under certain additional conditions, an abandoned child acquires British citizenship unless it can be shown that both parents have a different citizenship. Since Kakas *et al.* have shown that negation as failure can be given an argument-based interpretation, where negation-as failure assumptions are defeated by proving their contrary, we can say that [Sergot *et al.*, 1986] model reasoning with rules and exceptions at the logical and the dialectical layer.

Allen & Saxon's criticism An interesting criticism of Sergot *et al.*'s claim concerning exceptions was put forward by [Allen and Saxon, 1989]. They argued that the defeasible nature of legal reasoning is irreducibly procedural, so that it cannot be captured by current nonmonotonic logics, which define defeasible consequence only as a 'declarative' relation between premises and conclusion of an argument. In particular, they attacked the formalisation of 'unless shown otherwise' with negation as failure by arguing that 'shown' in this context does not mean 'logically proven from the available premises' but "shown by a process of argumentation and the presenting of evidence to an authorized decision-maker". So 'shown' would not refer to logical but to legal-procedural nonprovability.

In our opinion, Allen & Saxon are basically right, since such expressions address the allocation of the burden of proof, which in legal procedure is a matter of decision by the judge rather than of inference, and therefore primarily concerns the procedural layer rather than the dialectical one (as is Sergot *et al.*'s use of negation by failure). Note that these remarks apply not only to Sergot *et al.*'s work, but to any approach that stays within the dialectical layer. In Section 4.4 we will come back to this issue in more detail.

Applications of Logic Metaprogramming In two later projects the legal application of logic-programming was enriched with techniques from logic metaprogramming. Hamfelt [1995] uses such techniques for (among other things) representing legal collision rules and interpretation schemes. His method uses logic programming's DEMO predicate, which represents provability in the object language. Since much knowledge used in legal reasoning is metalevel knowledge, Hamfelt's approach might be a useful component of models of legal argument. However, it is not immediately clear how it can be embedded in a dialectical context, so that more research is needed.

The same holds for the work of Routen and Bench-Capon [1991], who have applied logic metaprogramming to, among other things, the representation of rules and exceptions. Their method provides a way to implement the exclusion approach of Section 2.2. They enrich the knowledge representation language with metalevel expressions $\texttt{Exception\_to}(rule_1, rule_2)$, and ensure that their metainterpreter applies a rule only if no exceptional rule can be applied. Although this is an elegant method, it also has some restrictions. Most importantly, it is not embedded in an argument-based model, so that it cannot easily be combined with other ways to compare conflicting arguments. Thus their method seems better suited for representing coherent legal texts than for modelling legal argument.

4.2 Applications of argument-based logics

Next we discuss legal applications of logics for defeasible argumentation. Several of these applications in fact use argument-based versions of logic programming.

Prakken & Sartor Prakken and Sartor [1996, 1997] have developed an argument-based logic similar to the one of [Simari and Loui, 1992], but that is expressive enough to deal with contradictory rules, rules with assumptions, inapplicability statements, and priority rules. Their system applies the well-known abstract approach to argumentation, logic programming and nonmonotonic reasoning developed by Dung [1995] and Bondarenko *et al.* [1997]. The logical language of the system is that of extended logic programming i.e., it has both negation as failure ($\sim$) and classical, or strong negation ($\neg$). Moreover, each formula is preceded by a term, its name. (In [Prakken, 1997] the system is generalised to the language of default logic.) Rules are *strict*, represented with $\rightarrow$, or else *defeasible*, represented with $\Rightarrow$. Strict rules are beyond debate; only defeasible rules can make an argument subject to defeat. Accordingly, facts are represented as strict rules

with empty antecedents (e.g. $\rightarrow$ *gave-up-house*). The input information of the system, i.e., the premises, is a set of strict and defeasible rules, which is called an *ordered theory* ('ordered' since an ordering on the defeasible rules is assumed).

Arguments can be formed by chaining rules, ignoring weakly negated antecedents; each head of a rule in the argument is a conclusion of the argument. Conflicts between arguments are decided according to a binary relation of *defeat* among arguments, which is partly induced by rule priorities. An important feature of the system is that the information about these priorities is itself presented as premises in the logical language. Thus rule priorities are as any other piece of legal information established by arguments, and may be debated as any other legal issue. The results of such debates are then transported to and used by the metatheory of the system.

There are three ways in which an argument Arg_2 can defeat an argument Arg_1. The first is *assumption* defeat (in the above publications called "undercutting" defeat), which occurs if a rule in Arg_1 contains $\sim L$ in its body, while Arg_2 has a conclusion L. For instance, the argument $[r_1\colon \rightarrow p,\ r_2\colon p \Rightarrow q]$ (strictly) defeats the argument $[r_3\colon \sim q \Rightarrow r]$ (note that $\sim L$ reads as 'there is no evidence that L'). This way of defeat can be used to formalise the explicit-exception approach of Section 2.2. The other two forms of defeat are only possible if Arg_1 does not assumption-defeat Arg_2. One way is by *excluding* an argument, which happens when Arg_2 concludes for some rule r in Arg_1 that r is not applicable (formalised as $\neg appl(r)$). For instance, the argument $[r_1\colon \rightarrow p,\ r_2\colon p \Rightarrow \neg appl(r_3)]$ (strictly) defeats the argument $[r_3\colon \Rightarrow r]$ by excluding it. This formalises the exclusion approach of Section 2.2. The final way in which Arg_2 can defeat Arg_1 is by *rebutting* it: this happens when Arg_1 and Arg_2 contain rules that are in a head-to-head conflict and Arg_2's rule is not worse than the conflicting rule in Arg_1. This way of defeat supports the implicit-exception approach.

Argument status is defined with a dialectical proof theory. The proof theory is correct and complete with respect to [Dung, 1995]'s grounded semantics, as extended by Prakken and Sartor to the case with reasoning about priorities. The opponent in a game has just one type of move available, stating an argument that defeats proponent's preceding argument (here defeat is determined as if no priorities were defined). The proponent has two types of moves: the first is an argument that combines an attack on opponent's preceding argument with a priority argument that makes the attack strictly defeating opponent's argument; the second is a priority argument that neutralises the defeating force of O's last move. In both cases, if proponent uses a priority argument that is not justified by the ordered theory, this will reflect itself in the possibility of successful attack of the argument by the opponent.

We now present the central definition of the dialogue game ('Arg-defeat' means defeat on the basis of the priorities stated by Arg). The first condition says that the proponent begins and then the players take turns, while the second condition prevents the proponent from repeating a move. The last two conditions were just explained and form the heart of the definition.

A *dialogue* is a finite nonempty sequence of moves $move_i = (Player_i, Arg_i)$ $(i > 0)$, such that

1. $Player_i = P$ iff i is odd; and $Player_i = O$ iff i is even;
2. If $Player_i = Player_j = P$ and $i \neq j$, then $Arg_i \neq Arg_j$;
3. If $Player_i = P$ then Arg_i is a minimal (w.r.t. set inclusion) argument such that
   (a) Arg_i strictly Arg_i-defeats Arg_{i-1}; or
   (b) Arg_{i-1} does not Arg_i-defeat A_{i-2};
4. If $Player_i = O$ then Arg_i $\emptyset$-defeats Arg_{i-1}.

The following simple dialogue illustrates this definition. It is about a tax dispute about whether a person temporarily working in another country has changed his fiscal domicile. All arguments are citations of precedents.[3]

P_1: [f_1: *kept-house*,
r_1: *kept-house* $\Rightarrow \neg$ *change*]

(Keeping one's old house is a reason against change of fiscal domicile.)

O_1: [f_2: $\neg$ *domestic-headquarters*,
r_2: $\neg$ *domestic-headquarters* $\Rightarrow \neg$ *domestic-company*,
r_3: $\neg$ *domestic-company* $\Rightarrow$ *change*]

(If the employer's headquarters are in the new country, it is a foreign company, in which case fiscal domicile has changed.)

P_2: [f_3: *domestic-property*,
r_4: *domestic-property* $\Rightarrow$ *domestic-company*,
f_4: r_4 *is decided by higher court than* r_2,
r_5: r_4 *is decided by higher court than* $r_2 \Rightarrow r_2 \prec r_4$]

(If the employer has property in the old country, it is a domestic company. The court which decided this is higher than the court deciding r_2.)

The proponent starts the dialogue with an argument P_1 for $\neg$ *change*, after which the opponent attacks this argument with an argument O_1 for the opposite conclusion. O_1 defeats P_1 as required, since in our logical system two rebutting arguments defeat each other if no priorities are stated. P_2 illustrates the first possible reply of the proponent to an opponent's move: it combines an object level argument for the conclusion *domestic-company* with a priority argument that gives r_4 precedence over r_2 and thus makes P_2 strictly defeat O_1. The second possibility, just stating a priority argument that neutralises the opponent's move, is illustrated by the following alternative move, which resolves the conflict between P_1 and O_1 in favour of P_1:

P_2': [f_5: r_1 *is more recent than* r_3,
p': r_1 *is more recent than* $r_3 \Rightarrow r_3 \prec r_1$]

[3] Facts f_i: $\rightarrow p_i$ are abbreviated as f_i: p_i.

Kowalski & Toni Like Prakken and Sartor, Kowalski and Toni [1996] also apply the abstract approach of [Dung, 1995, Bondarenko *et al.*, 1997] to the legal domain, instantiating it with extended logic programming. Among other things, they show how priority principles can be encoded in the object language without having to refer to priorities in the metatheory of the system. We illustrate their method using the language of [Prakken and Sartor, 1996]. Kowalski and Toni split each rule r: $P \Rightarrow Q$ into two rules

$$Applicable(r) \Rightarrow Q$$
$$P \wedge \sim Defeated(r) \Rightarrow Applicable(r)$$

The predicate *Defeated* is defined as follows:

$$r \prec r' \wedge Conflicting(r, r') \wedge Applicable(r') \rightarrow Defeated(r)$$

Whether $r \prec r'$ holds, must be (defeasibly) derived from other information. Kowalski and Toni also define the *Conflicting* predicate in the object language.

Three formal reconstructions of HYPO-style case-based reasoning The dialectical nature of the HYPO system has inspired several logically inclined researchers to reconstruct HYPO-style reasoning in terms of argument-based defeasible logics. We briefly discuss three of them, and refer to [Hage, 1997] for a reconstruction in Reason-based Logic (cf. Section 4.3 below).

Loui et al. (1993) Loui *et al.* [1993] proposed a reconstruction of HYPO in the context of the argument-based logic of [Simari and Loui, 1992]. They mixed the pro and con factors of a precedent in one rule

$$Pro\text{-}factors \wedge Con\text{-}factors \Rightarrow Decision$$

but then implicitly extended the case description with rules containing a superset of the con factors and/or a subset of the con factors in this rule. Loui *et al.* also studied the combination of reasoning with rules and cases. This work was continued in [Loui and Norman, 1995] (discussed below in Section 4.5).

Prakken and Sartor (1998) Prakken and Sartor [1998] have modelled HYPO-style reasoning in their [1996] system, also adding additional expressiveness. As Loui *et al.* [1993] they translate HYPO's cases into a defeasible-logical theory. However, unlike Loui *et al.*, Prakken and Sartor separate the pro and con factors into two conflicting rules, and capture the case decision with a priority rule. This method is an instance of a more general idea (taken from [Loui and Norman, 1995]) to represent precedents as a set of arguments pro and con the decision, and to capture the decision by a justified priority argument that in turn makes the argument for the decision justified. In its simplest form where, as in HYPO, there are just a decision and sets of factors pro and con the decision, this amounts to having a pair of rules

r_1: *Pro-factors* $\Rightarrow$ *Decision*
r_2: *Con-factors* $\Rightarrow \neg$*Decision*

and an unconditional priority rule

p: $\Rightarrow r_1 \succ r_2$

However, in general arguments can be multi-step (as suggested by [Branting, 1994]) and priorities can very well be the justified outcome of a competition between arguments.

Analogy is now captured by a 'rule broadening' heuristic, which deletes the antecedents missing in the new case. And distinguishing is captured by a heuristic which introduces a conflicting rule 'if these factors are absent, then the consequent of your broadened rule does not hold'. So if a case rule is r_1: $f_1 \wedge f_2 \Rightarrow d$, and the CFS consists of f_1 only, then r_1 is analogised by $b(r_1)$: $f_1 \Rightarrow d$, and $b(r_1)$ is distinguished by $d(b(r_1))$: $\neg f_2 \Rightarrow \neg d$. To capture the heuristic nature of these moves, Prakken and Sartor 'dynamify' their [1996] dialectical proof procedure, to let it cope with the introduction of new premises.

Finally, in [Prakken, 2001a] it is, inspired by [Bench-Capon and Sartor, 2001], shown how within this setup cases can be compared not on factual similarities but on the basis of underlying values.

Horty (1999) Horty [1999] has reconstructed HYPO-style reasoning in terms of his own work on two other topics: defeasible inheritance and defeasible deontic logic. Since inheritance systems are a forerunner of logics for defeasible argumentation, Horty's reconstruction can also be regarded as argument-based. It addresses the analogical citation of cases and the construction of multi-steps arguments. To support the citation of precedents on their intermediate steps, cases are separated into 'precedent constituents', which contain a set of factors and a possibly intermediate outcome. Arguments are sequences of factor sets, starting with the current fact situation and further constructed by iteratively applying precedent constituents that share at least one factor with the set constructed thus far. Conflicting uses of precedent constituents are compared with a variant of HYPO's more-on-point similarity criterion. The dialectical status of the constructible arguments is then assessed by adapting notions from Horty's inheritance systems, such as 'preemption'.

Other work on argument-based logics Legal applications of argument-based logic programming have also been studied by Nitta and his colleagues; see e.g. [Nitta and Shibasaki, 1997]. Besides rule application, their argument construction principles also include some simple forms of analogical reasoning. However, no undercutters for analogical arguments are defined. The system also has a rudimentary dialogue game component.

Formal work on dialectical proof theory with an eye to legal reasoning has been done by Jakobovits and Vermeir [1999]. Their focus is more on technical development than on legal applications.

4.3 Reason-based Logic

Hage [1996, 1997] and Verheij [1996] have developed a formalism for legal reasoning called 'reason-based logic' (RBL), centering around a deep philosophical account of the concept of a rule. It is meant to capture how legal (and other) principles, goals and rules give rise to reasons for and against a proposition and how these reasons can be used to draw conclusions. The underlying view on principles, rules and reasons is influenced by insights from analytical philosophy on the role of reasons in practical reasoning, especially [Raz, 1975]. Hage and Verheij stress that rule application is much more than simple modus ponens. It involves reasoning about the validity and applicability of a rule, and weighing reasons for and against the rule's consequent.

RBL's View on Legal Knowledge RBL reflects a distinction between two levels of legal knowledge. The primary level includes principles and goals, while the secondary level includes rules. Principles and goals express reasons for or against a conclusion. Without the secondary level these reasons would in each case have to be weighed to obtain a conclusion, but according to Hage and Verheij rules express the outcome of certain weighing process. Therefore, a rule does not only generate a reason for its consequent but also generates a so-called 'exclusionary' reason against applying the principles underlying the rule: the rule replaces the reasons on which it is based. This view is similar to Dworkin's [1977] well-known view that while principles are weighed against each other, rules apply in an all-or-nothing fashion. However, according to Hage [1996] and Verheij [Verheij *et al.*, 1998] this difference is just a matter of degree: if new reasons come up, which were not taken into account in formulating the rule, then these new reasons are not excluded by the rule; the reason based on the rule still has to be compared with the reasons based on the new principles. Consequently, in RBL rules and principles are syntactically indistinguishable; their difference is only reflected in their degree of interaction with other rules and principles (but Hage [1997] somewhat deviates from this account.)

A Sketch of the Formal System To capture reasoning about rules, RBL provides the means to express properties of rules in the object language. To this end Hage and Verheij use a sophisticated naming technique, viz. *reification*, well-known from metalogic and AI [Genesereth and Nilsson, 1988, p. 13], in which every predicate constant and logical symbol is named by a function expression. For instance, the conjunction $R(a) \wedge S(b)$ is denoted by the infix function expression $r(a) \wedge' s(b)$. Unlike the naming techniques used by [Gordon, 1995] and [Prakken and Sartor, 1996], RBL's technique reflects the logical structure of the named formula.

Rules are named with a function symbol *rule*, resulting in terms like

$\texttt{rule}(r, p(x), q(x))$

Here r is a 'rule identifier', $p(x)$ is the rule's condition, and $q(x)$ is its consequent. RBL's object language does not contain a conditional connective corresponding to the function symbol *rule*; rules can only be stated indirectly, by assertions that they are valid, as in

$\texttt{Valid}(\texttt{rule}(r, condition_r, conclusion_r))$

Hage and Verheij state RBL as extra inference rules added to standard first-order logic or, in some versions, as extra semantic constraints on models of a first-order theory. We first summarise the most important rules and then give some (simplified) formalisations.

1. If a rule is valid, its conditions are satisfied and there is no evidence that it is excluded, the rule is applicable.
2. If a rule is applicable, it gives rise to a reason for its application.
3. A rule applies if and only if the set of all derivable reasons for its application outweighs the set of all derivable reasons against its application.
4. If a rule applies, it gives rise to a reason for its consequent.
5. A formula is a conclusion of the premises if and only if the reasons for the formula outweigh the reasons against the formula.

Here is how a simplified formal version of inference rule (1) looks like. Note that *condition* and *consequent* are variables, which can be instantiated with the name of any formula.

If $\texttt{Valid}(\texttt{rule}(r, condition, consequent))$ is derivable
and $\texttt{Obtains}(condition)$ is derivable
and $\texttt{Excluded}(r))$ is not derivable,
then $\texttt{Applicable}(r, \texttt{rule}(condition, consequent))$ is derivable.

Condition (4) has the following form.

If $\texttt{Applies}(r, \texttt{rule}(condition, consequent))$ is derivable,
then $\texttt{Proreason}(consequent)$ is derivable.

Finally, here is how in condition (5) the connection between object- and metalevel is made.

If $\texttt{Outweighs}(\texttt{Proreasons}(formula), \texttt{Conreasons}(formula))$ is derivable,
then `Formula` is derivable.

Whether the pro-reasons outweigh the con-reasons must itself be derived from the premises. The only built-in constraint is that any nonempty set outweighs the empty set. Note that while $formula$ is a variable for an object term, occurring in a well-formed formula of RBL, `Formula` is a metavariable which stands for the formula named by the term $formula$. This is how object and metalevel are in RBL connected.

In RBL the derivability of certain formulas is defined in terms of the non-derivability of other formulas. For instance, in (1) it may not be derivable that the rule is excluded. To deal with this, RBL adapts techniques of default logic, by restating the inference rules as conditions on membership of an extension.

Using RBL In RBL exceptions can be represented both explicitly and implicitly. As for explicit exceptions, since RBL has the validity and applicability requirements for rules built into the logic, the exclusion method of Section 2.2 can be used. RBL also supports the choice approach: if two conflicting rules both apply and do not exclude each other, then their application gives rise to conflicting reasons, which have to be weighed.

Finally, Hage and Verheij formalise legal priority principles in a similar way as [Kowalski and Toni, 1996], representing them as inapplicability rules. The following example illustrates their method with the three well known legal principles *Lex Superior* (the higher regulation overrides the lower one), *Lex Posterior* (the later rule overrides the earlier one) and *Lex Specialis* (the specificity principle). It is formalised in the language of [Prakken and Sartor, 1996]; recall that with respect to applicability, this system follows, as RBL, the exclusion approach.

The three principles can be expressed as follows.

H: x `conflicts with` $y \wedge y$ `is inferior to` $x \wedge \sim \neg$`appl`$(x) \Rightarrow \neg$`appl`(y)
T: x `conflicts with` $y \wedge y$ `is earlier than` $x \wedge \sim \neg$`appl`$(x) \Rightarrow \neg$`appl`(y)
S: x `conflicts with` $y \wedge x$ `is more specific than` $y \wedge \sim \neg$`appl`$(x) \Rightarrow$
$\neg$`appl`(y)

Likewise for the ordering of these three principles:

HT: T `conflicts with` $H \wedge \sim \neg$`appl`$(H) \Rightarrow \neg$`appl`(T)
TS: S `conflicts with` $T \wedge \sim \neg$`appl`$(T) \Rightarrow \neg$`appl`(S)
HS: S `conflicts with` $H \wedge \sim \neg$`appl`$(H) \Rightarrow \neg$`appl`(S)

Thus the metatheory of the logic does not have to refer to priorities. However, the method contains another metareasoning feature, viz. the ability to express metalevel statements of the kind x `conflicts with` y.

Evaluation RBL clearly confines itself to the logical and dialectical layer of legal argument. At these layers, it is a philosophically well-motivated analysis of legal reasoning, while technically it is very expressive, supporting reasoning with rules and exceptions, with conflicting rules, and about rules and their priority relations. However, it remains to be investigated how RBL can, given its complicated technical nature and the lack of the notion of an argument, be embedded in procedural and heuristic accounts of legal argument.

4.4 Procedural accounts of legal reasoning

The Pleadings Game is not the only procedural AI & Law model. We now briefly discuss some formal models of this kind.

Hage, Leenes and Lodder At the same time when Gordon designed his system, Hage *et al.* [1994] developed a procedural account of Hart's distinction between clear and hard cases. They argued that whether a case is easy or hard depends on the stage of a procedure: a case that is easy at an earlier stage, can be made hard by introducing new information. This is an instance of their purely procedural view on the law, which incorporates substantive law by the judge's obligation to apply it. To formalise this account, a Hamblin-MacKenzie-style formal dialogue system with the possibility of counterargument was developed. This work was extended by [Lodder, 1999] in his DiaLaw system.

The general setup of these systems is the same as that of the Pleadings Game. For the technical differences the reader is referred to the above publications. One difference at the dialectical layer is that instead of an argument-based logic, Hage and Verheij's reason-based logic is used. Another difference in [Hage *et al.*, 1994] is that it includes a third party, the *referee*, who is entitled to decide whether certain claims should be accepted by the parties or not. The dialogue systems also support disputes about the procedural legality of a move. Finally, arguments do not have to be logically valid; the only use of reason-based logic is to determine whether a claim of one's opponent follows from one's commitments and therefore must be accepted.

Bench-Capon Bench-Capon [1998] has also developed a dialogue game for legal argument. As the above-discussed games, it has the possibility of counterargument (although it does not incorporate a formalised account of the dialectical layer). The game also has a referee, with roughly the same role as in [Hage *et al.*, 1994]. Bench-Capon's game is especially motivated by the desire to generate more natural dialogues than the "stilted" ones of Hamblin-MacKenzie-style systems. To this end, arguments are defined as variants of Toulmin's [1958] argument structures, containing a *claim*, *data* for this claim, a *warrant* connecting data and claim, a *backing* for the warrant, and possible *rebuttals* of the claim with an exception. The available speech acts refer to the use or attack of these items, which, according to Bench-Capon, induces natural dialogues.

Formalising allocations of the burden of proof Above we supported Allen and Saxon's [1989] criticism of Sergot *et al.*'s [1986] purely logical- and dialectical-layer account of reasoning with exceptions. Additional support is provided by Prakken [2001b], who argues that allocations of burden of proof cannot be modelled by 'traditional' nonmonotonic means.

Burden of proof is one of the central notions of legal procedure, and it is clearly connected with defeasibility [Loui, 1995, Sartor, 1995]. There are two aspects of having the burden of proving a claim: the task to come with an argument for that claim, and the task to uphold this argument against challenge in a dispute. The first aspect can be formalised in Hamblin-MacKenzie-style dialogue systems (discussed above in Section 2.3). The second aspect requires a system that assesses arguments on the basis of the dialectical interactions between all available arguments. At first sight, it would seem that dialectical

proof theories of nonmonotonic logics can be directly applied here. However, there is a problem, which we shall illustrate with an example from contract law.

In legal systems it is generally the case that the one who argues that a valid contract exists has the burden of proving those facts that ordinarily give rise to the contract, while the party who denies the existence of the contract has the burden of proving why, despite these facts, exceptional circumstances prevent the contract from being valid. Now suppose that plaintiff claims that a contract between him and defendant exists because plaintiff offered defendant to sell her his car, and defendant accepted. Then plaintiff has the burden of proving that there was such an offer and acceptance, while defendant has the burden of proving, for instance, that the car had a hidden defect. Suppose we formalise this in [Prakken and Sartor, 1996] as follows:

r_1: *offer* $\wedge$ *acceptance* $\wedge \sim$ *exception*$(r_1) \Rightarrow$ *contract*
r_2: *hidden defect* $\Rightarrow$ *exception*(r_1)

Suppose further that in the dispute arguments for and against *hidden defect* are exchanged, and that the judge regards them of equal strength.

What follows dialectically? If plaintiff starts with moving his argument for *contract*, then defendant can assumption-defeat this argument with her argument for *exception*(r_1). Plaintiff cannot attack this with his argument against *hidden defect* since it is of equal strength as defendant's argument, so it does not strictly defeat it. In conclusion, plaintiff's argument is not justified (but merely defensible), so the outcome of our logical reconstruction is that plaintiff has not fulfilled his burden of proof.

However, the problem with this reconstruction is that it ignores that neither has defendant fulfilled her burden of proof: she had to prove *hidden defect*, but her argument for this conclusion also is merely defensible. The problem with the (sceptical) dialectical proof theory is that plaintiff has the burden of proof with respect to all subissues of the dispute; there is no way to distribute the burden of proof over the parties, as is common in legal dispute. This problem is not confined to the particular system or knowledge representation method, but seems a fundamental problem of current 'traditional' nonmonotonic logics.

An additional problem for such logics is that in legal procedure the allocation of the burden of proof is ultimately a matter of decision by the judge, and therefore cannot be determined by logical form. Any full model of reasoning under burden of proof should leave room for such decisions.

In [Prakken, 2001b] the dialectical proof theory for grounded semantics is adapted to enable distributions of the burden of proof over the parties, which in [Prakken, 2001c] is embedded in a dialogue game model for legal procedure. The basic idea of [Prakken, 2001b] is that the required strength of a move depends on who has the burden of proof concerning the issue under attack (as decided by the judge in the dialogue game). The resulting system has no clear link with argument-based semantics in the style of [Dung, 1995, Bondarenko *et al.*, 1997]. For logicians this is perhaps disappointing, but for others this will count

as support for the view that the semantics of (legal) defeasible reasoning is essentially procedural.

ZENO's argumentation framework Another account of distributions of the burden of proof in dialectical systems is given by Gordon and Karaçapilidis [1997]. In fact, [Prakken, 2001b]'s proposal can partly be seen as a generalisation and logical formalisation of this account. Gordon and Karaçapilidis incorporate variants of Freeman and Farley's 'levels of proof' in their 'ZENO argumentation framework'. This is the dialectical-layer part of the ZENO argument mediation system: it maintains a 'dialectical graph' of the issues, the positions with respect to these issues, and the arguments pro and con these positions that have been advanced in a discussion, including positions and arguments about the strength of other arguments. Arguments are links between positions.

Part of the framework is a status assignment to positions: each position is assigned *in* or *out* depending on two factors: the required level of proof for the position, and the relative strengths of the arguments pro and con the position that themselves have antecedents that are *in*. For instance, a position with level 'scintilla of evidence' is *in* iff at least one argument pro is *in* (here they deviate from Freeman and Farley). And a position with level 'preponderance of evidence' is *in* iff the joint pro arguments that are *in* outweigh the joint con arguments that are *in*. The burden of proof can be distributed over the parties since levels of proof can be assigned to arbitrary positions instead of (as in [Freeman and Farley, 1996]) only to the initial claim of a dispute.

4.5 Formalisations of the heuristic layer

In logical models of legal argument the heuristic layer has so far received very little attention. Above we discussed Prakken and Sartor's [1998] logical reconstruction of HYPO-style analogising and distinguishing as premise introduction heuristics. Perhaps the most advanced formal work on the heuristic layer to date is Loui and Norman's [1995] study of the use of rationales of cases and rules in legal argument, which we will now discuss in some detail.

Loui and Norman (1995) Loui and Norman [1995] formally define a protocol for the exchange of arguments and counterarguments, and analyse within the protocol various uses of rationales of rule and cases. These uses are modelled as ways to modify a previously stated argument. Thus their various uses of rationales can be regarded as inventional heuristics.

More precisely, each move states and/or modifies one or more arguments. Newly stated arguments are added to the so-called *argument record*, which is shared by the players. Modifications modify an argument on the record moved by the other player, in order to have new ways to attack it. Each move must change the status of the main claim: if the proponent moves, the status must change to 'justified', while if the opponent moves, it must change to 'defensible'

or 'overruled'. Whether a move achieves this, is tested by applying the argumentation logic of [Simari and Loui, 1992] to the argument record resulting from the move (taking only the explicitly stated arguments into account).

We now summarise the types of rationales identified by Loui and Norman and how they can be used to generate new 'attacking points'. Then we discuss the use of one type in more detail.

Compression rationales. Some rules compress a line of reasoning in a single if-then rule. For instance, the rule 'vehicles are not allowed in the park' might compress 'vehicles used for private transportation are not allowed in the park' and 'vehicles are normally used for private transportation'. Unpacking the compressed rule enables an attack on the latter rule, for instance, with 'ambulances are not used for private transport'. Semiformally: unpack your opponent's use of $A \Rightarrow B$ as $A \Rightarrow C$, $C \Rightarrow B$ and state an argument for $\neg C$.

Specialisation rationales. Sometimes a rule can be argued to implement a principle. For instance, the rule 'mail order buyers can cancel their order within one week' could be argued to specialise the principle 'weak contract parties should be protected', since mail order buyers (usually consumers) are weak parties and allowing them to cancel their order within a week is a way to protect them. A rationale-based attack could restate the rule as 'insofar as mail order buyers are weak parties, they can cancel their order within one week'. This enables an attack on the weakness of the party, for instance, when the buyer is a company. The logical form is: if we have a rule $B \Rightarrow C$ and a principle $W \Rightarrow P$, and we have that $B \Rightarrow W$ and $C \Rightarrow P$, then replace the rule with $W \Rightarrow C$, and attack the modified argument with an argument for $\neg W$.

Fit rationales. Sometimes a rule is defended by arguing that it explains the decisions of a given set of precedents. This rule could be modified into a rule that equally well explains the cases but that does not apply in the new case, or is susceptible to a new attack. Other forms of attack are also possible, for instance, adding a precedent to the set and arguing that a conflicting rule better explains the resulting set.

Disputation rationales. Sometimes the *ratio decidendi* of a precedent is the result of a choice between conflicting arguments. Then the case rule can be replaced by these conflicting arguments, and by showing that in the new fact situation the outcome of the dispute would have been different.

Let us illustrate this in some detail. Assume a case rule $B \wedge C \Rightarrow A$, which compresses the adjudication between the following three arguments (for notational convenience we use specificity to express the comparison of the arguments).

Arg_1: B, $B \Rightarrow A$, so A
Arg_2: C, $C \Rightarrow D$, $D \Rightarrow \neg A$, so $\neg A$
Arg_3: B, C, $B \Rightarrow F$, $F \wedge C \Rightarrow \neg D$, so $\neg D$

Loui and Norman's protocol allows the following dispute:

- P: I have an argument for A:
  - Arg_0: B, C, $B \wedge C \Rightarrow A$, so A

  Argument record = $\{Arg_0\}$
- O: Your rule compresses the adjudication between three arguments, so:
  Argument record = $\{Arg_1, Arg_2, Arg_3\}$
- O: And I attack Arg_3 with:
  - Arg_4: B, G, $B \wedge G \Rightarrow \neg F$, so $\neg F$

  Argument record = $\{Arg_1, Arg_2, Arg_3, Arg_4\}$

Applying [Simari and Loui, 1992]'s system to the argument records before and after O's move, we see that A is justified on the basis of the former record, but overruled on the basis of the latter. So O has fulfilled her task of changing the status of P's main claim.

5 Conclusion

One aim of this review has been to show that there is more to legal argument than inference (whether deductive or defeasible). Another aim has been to argue that logic is more fruitfully applied to legal reasoning if the context in which it is to be used is taken into account. Our four-layered view on legal argument is an attempt to provide the necessary context. Two main features of this context are that it is dynamic and that it is dialectical: the theory with which to reason is not given but must be constructed, in dialectical interaction with one's adversaries, and within procedural bounds.

Summarising in more detail our overview of logical research on legal argument, we can say that the dialectical layer has been largely dealt with. Adapting general techniques from nonmonotonic logic, various sophisticated methods have been developed for formalising reasoning with rules and exceptions, with rule priorities (including combining several sources of priorities), about rule priorities, and about other properties of legal rules, such as their backing, validity or applicability, and their correct interpretation. Above all, AI & Law has shown how all these elements can be integrated. Of course, the implementation of these formal models involves computational issues. However, these issues fall outside the present paper: the field of AI & Law can here rely on relevant work in other fields, such as automated theorem proving.

At the procedural layer considerable progress has been made. However, a general framework is still lacking. Such a framework is needed since, although most current procedures are carefully designed, it is often hard to see their underlying structure. This makes it hard to study their properties and also to design new procedures. A possible framework is proposed in [Prakken, 2000], leaving room for various sets of speech acts, various underlying defeasible logics, and various options on trying alternative moves.

For logicians, the study of disputational procedures opens a new range of research questions. One issue is the formalisation of 'self-modifying' procedures,

i.e., the possibility to change a procedure in a dispute governed by that same procedure; cf. [Vreeswijk, 2000]. Another issue is the relation between the dialectical and procedural layer, especially when dialogue systems incorporate dialectical proof theories in their dialogue rules (as studied by [Prakken, 2001d]). Also, general principles should be studied for how to enable as many 'sensible' dialogues as possible while disallowing all 'non-sensible' dialogues.

Finally, the formalisation of the heuristic aspects of legal argument is still in its early stages. Some interesting research issues are:

- Formalisation of nondeductive types of arguments.
- Formalisation of inventional, presentation and selection heuristics.
- Formalisation of the notion of *persuasive* argumentation.
- Drawing the dividing line between argument construction rules and premise introduction heuristics.

Finally, there is the more general issue as to the limits of argument-based approaches. Perhaps more 'holistic' approaches are needed, where people exchange entire theories with each other, which are assessed on coherence; cf. e.g. [Bench-Capon and Sartor, 2001].

Acknowledgement

This paper is to appear in A. Kakas and F. Sadri (eds.), *Computational Logic: From Logic Programming into the Future (In honour of Bob Kowalski)*. Berlin: Springer Verlag, 2001.

References

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

[Alexy, 1978] R. Alexy. *Theorie der juristischen Argumentation. Die Theorie des rationalen Diskurses als eine Theorie der juristischen Begründung.* Suhrkamp Verlag, Frankfurt am Main, 1978.

[Allen and Saxon, 1989] L.E. Allen and C.S. Saxon. Relationship of expert systems to the operation of a legal system. In *Preproceedings of the III International Conference on "Logica, Informatica, Diritto" (Appendix)*, pages 1–15, Florence, 1989.

[Bench-Capon and Sartor, 2001] T.J.M. Bench-Capon and G. Sartor. Theory based explanation of case law domains. In *Proceedings of the Eighth International Conference on Artificial Intelligence and Law*, pages 12–21, New York, 2001. ACM Press.

[Bench-Capon *et al.*, 1987] T.J.M. Bench-Capon, G.O. Robinson, T.W. Routen, and M.J. Sergot. Logic programming for large scale applications in law: a formalisation of supplementary benefit legislation. In *Proceedings of the First International Conference on Artificial Intelligence and Law*, pages 190–198, New York, 1987. ACM Press.

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

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

[Branting, 1994] L.K. Branting. A computational model of ratio decidendi. *Artificial Intelligence and Law*, 2:1–31, 1994.

[Brewka and Gordon, 1994] G. Brewka and T.F. Gordon. How to buy a porsche, an approach to defeasible decision making. In *Working Notes of the AAAI-94 Workshop on Computational Dialectics*, pages 28–38, Seattle, Washington, 1994.

[Brewka, 1994] G. Brewka. A logical reconstruction of Rescher's theory of formal disputation based on default logic. In *Proceedings of the Eleventh European Conference on Artificial Intelligence*, pages 366–370, 1994.

[Brewka, 2001] G. Brewka. Dynamic argument systems: a formal model of argumentation processes based on situation calculus. *Journal of Logic and Computation*, 11:257–282, 2001.

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

[Dworkin, 1977] R.M. Dworkin. Is law a system of rules? In R.M. Dworkin, editor, *The Philosophy of Law*, pages 38–65. Oxford University Press, Oxford, 1977.

[Freeman and Farley, 1996] K. Freeman and A.M. Farley. A model of argumentation and its application to legal reasoning. *Artificial Intelligence and Law*, 4:163–197, 1996.

[Gardner, 1987] A. Gardner. *Artificial Intelligence Approach to Legal Reasoning.* MIT Press, Cambridge, MA, 1987.

[Geffner, 1992] H. Geffner. *Default reasoning: causal and conditional theories.* MIT Press, Cambridge, MA, 1992.

[Gelfond and Lifschitz, 1990] M. Gelfond and V. Lifschitz. Logic programs with classical negation. In *Proceedings of the Seventh Logic Programming Conference*, pages 579–597, Cambridge, MA, 1990. MIT Press.

[Genesereth and Nilsson, 1988] M.R. Genesereth and N.J. Nilsson. *Logical Foundations of Artificial Intelligence.* Morgan Kaufmann Publishers Inc, Palo Alto, CA, 1988.

[Gordon and Karaçapilidis, 1997] T.F. Gordon and N. Karaçapilidis. The Zeno argumentation framework. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, pages 10–18, New York, 1997. ACM Press.

[Gordon, 1991] T.F. Gordon. An abductive theory of legal issues. *International Journal of Man-Machine Studies*, 35:95–118, 1991.

[Gordon, 1994] T.F. Gordon. The Pleadings Game: an exercise in computational dialectics. *Artificial Intelligence and Law*, 2:239–292, 1994.

[Gordon, 1995] T.F. Gordon. *The Pleadings Game. An Artificial Intelligence Model of Procedural Justice.* Kluwer Academic Publishers, Dordrecht/Boston/London, 1995.

[Habermas, 1981] J. Habermas. *Theorie des Kommunikativen Handelns.* p, Frankfurt, 1981.

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

[Hage, 1996] J.C. Hage. A theory of legal reasoning and a logic to match. *Artificial Intelligence and Law*, 4:199–273, 1996.

[Hage, 1997] J.C. Hage. *Reasoning With Rules. An Essay on Legal Reasoning and Its Underlying Logic.* Law and Philosophy Library. Kluwer Academic Publishers, Dordrecht/Boston/London, 1997.

[Hamblin, 1971] C.L. Hamblin. Mathematical models of dialogue. *Theoria*, 37:130–155, 1971.

[Hamfelt, 1995] A. Hamfelt. Formalizing multiple interpretation of legal knowledge. *Artificial Intelligence and Law*, 3:221–265, 1995.

[Horty, 1999] J. Horty. Precedent, deontic logic, and inheritance. In *Proceedings of the Seventh International Conference on Artificial Intelligence and Law*, pages 63–72, New York, 1999. ACM Press.

[Horty, 2001] J. Horty. Argument construction and reinstatement in logics for defeasible reasoning. *Artificial Intelligence and Law*, 9, 2001. To appear.

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

[Jones and Sergot, 1992] A.J.I. Jones and M.J. Sergot. Deontic logic in the representation of law: towards a methodology. *Artificial Intelligence and Law*, 1:45–64, 1992.

[Kakas *et al.*, 1992] A.C. Kakas, R.A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2:719–770, 1992.

[Kowalski and Sergot, 1990] R.A. Kowalski and M.J. Sergot. The use of logical models in legal problem solving. *Ratio Juris*, 3:201–218, 1990.

[Kowalski and Toni, 1996] R.A. Kowalski and F. Toni. Abstract argumentation. *Artificial Intelligence and Law*, 4:275–296, 1996.

[Kowalski, 1989] R.A. Kowalski. The treatment of negation in logic programs for representing legislation. In *Proceedings of the Second International Conference on Artificial Intelligence and Law*, pages 11–15, New York, 1989. ACM Press.

[Kowalski, 1995] R.A. Kowalski. Legislation as logic programs. In Z. Bankowski, I. White, and U. Hahn, editors, *Informatics and the Foundations of Legal Reasoning*, Law and Philosophy Library, pages 325–356. Kluwer Academic Publishers, Dordrecht/Boston/London, 1995.

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

[Lodder, 1999] A.R. Lodder. *DiaLaw. On Legal Justification and Dialogical Models of Argumentation.* Law and Philosophy Library. Kluwer Academic Publishers, Dordrecht/Boston/London, 1999.

[Loui and Norman, 1995] R.P. Loui and J. Norman. Rationales and argument moves. *Artificial Intelligence and Law*, 3:159–189, 1995.

[Loui *et al.*, 1993] R.P. Loui, J. Norman, J. Olson, and A. Merrill. A design for reasoning with policies, precedents, and rationales. In *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, pages 202–211, New York, 1993. ACM Press.

[Loui *et al.*, 1997] R.P. Loui, J. Norman, J. Alpeter, D. Pinkard, D. Craven, J. Linsday, and M. Foltz. Progress on Room 5: A testbed for public interactive semi-formal legal argumentation. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, pages 207–214, New York, 1997. ACM Press.

[Loui, 1995] R.P. Loui. Hart's critics on defeasible concepts and ascriptivism. In *Proceedings of the Fifth International Conference on Artificial Intelligence and Law*, pages 21–30, New York, 1995. ACM Press.

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

[MacKenzie, 1990] J.D. MacKenzie. Four dialogue systems. *Studia Logica*, 51:567–583, 1990.

[McCarty and Sridharan, 1981] L.T. McCarty and N.S. Sridharan. The representation of an evolving system of legal concepts: II. Prototypes and deformations. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 246–253, 1981.

[McCarty, 1989] L.T. McCarty. A language for legal discourse I. basic features. In *Proceedings of the Second International Conference on Artificial Intelligence and Law*, pages 180–189, New York, 1989. ACM Press.

[McCarty, 1995] L.T. McCarty. An implementation of Eisner v. Macomber. In *Proceedings of the Fifth International Conference on Artificial Intelligence and Law*, pages 276–286, New York, 1995. ACM Press.

[Nitta and Shibasaki, 1997] K. Nitta and M. Shibasaki. Defeasible reasoning in Japanese criminal jurisprudence. *Artificial Intelligence and Law*, 5:139–159, 1997.

[Nute, 1992] D. Nute. Inferences, rules, and instrumentalism. *International Journal of Expert Systems*, 5:267–274, 1992.

[Nute, 1997] D. Nute, editor. *Defeasible Deontic Logic*, volume 263 of *Synthese Library*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1997.

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

[Peczenik, 1996] A. Peczenik. Jumps and logic in the law. *Artificial Intelligence and Law*, 4:297–329, 1996.

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

[Poole, 1988] D.L. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.

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

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

[Prakken and Sartor, 1998] H. Prakken and G. Sartor. Modelling reasoning with precedents in a formal dialogue game. *Artificial Intelligence and Law*, 6:231–287, 1998.

[Prakken and Vreeswijk, 2001] H. Prakken and G.A.W. Vreeswijk. Logics for defeasible argumentation. In D. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume 5. Kluwer Academic Publishers, Dordrecht/Boston/London, 2001. Second edition, to appear.

[Prakken, 1995] H. Prakken. From logic to dialectics in legal argument. In *Proceedings of the Fifth International Conference on Artificial Intelligence and Law*, pages 165–174, New York, 1995. ACM Press.

[Prakken, 1996] H. Prakken. Two approaches to the formalisation of defeasible deontic reasoning. *Studia Logica*, 57:73–90, 1996.

[Prakken, 1997] H. Prakken. *Logical Tools for Modelling Legal Argument. A Study of Defeasible Argumentation in Law*. Law and Philosophy Library. Kluwer Academic Publishers, Dordrecht/Boston/London, 1997.

[Prakken, 2000] H. Prakken. On dialogue systems with speech acts, arguments, and counterarguments. In *Proceedings of the 7th European Workshop on Logic for Artificial Intelligence (JELIA'2000)*, number 1919 in Springer Lecture Notes in AI, pages 224–238, Berlin, 2000. Springer Verlag.

[Prakken, 2001a] H. Prakken. An exercise in formalising teleological case-based reasoning. *Artificial Intelligence and Law*, 9, 2001. to appear.

[Prakken, 2001b] H. Prakken. Modelling defeasibility in law: logic or procedure? *Fundamenta Informaticae*, 2001. To appear.

[Prakken, 2001c] H. Prakken. Modelling reasoning about evidence in legal procedure. In *Proceedings of the Eighth International Conference on Artificial Intelligence and Law*, pages 119–128, New York, 2001. ACM Press.

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

[Rawls, 1972] J. Rawls. *A Theory of Justice*. Oxford University Press, Oxford, 1972.

[Raz, 1975] J. Raz. *Practical Reason and Norms*. Princeton University Press, Princeton, 1975.

[Rescher, 1977] N. Rescher. *Dialectics: a Controversy-oriented Approach to the Theory of Knowledge*. State University of New York Press, Albany, N.Y., 1977.

[Rissland and Ashley, 1987] E.L. Rissland and K.D. Ashley. A case-based system for trade secrets law. In *Proceedings of the First International Conference on Artificial Intelligence and Law*, pages 60–66, New York, 1987. ACM Press.

[Rissland and Skalak, 1991] E.L. Rissland and D.B. Skalak. CABARET: statutory interpretation in a hybrid architecture. *International Journal of Man-Machine Studies*, 34:839–887, 1991.

[Routen and Bench-Capon, 1991] T. Routen and T.J.M. Bench-Capon. Hierarchical formalizations. *International Journal of Man-Machine Studies*, 35:69–93, 1991.

[Sartor, 1995] G. Sartor. Defeasibility in legal reasoning. In Z. Bankowski, I. White, and U. Hahn, editors, *Informatics and the Foundations of Legal Reasoning*, Law and Philosophy Library, pages 119–157. Kluwer Academic Publishers, Dordrecht/Boston/London, 1995.

[Sartor, 1997] G. Sartor. Logic and argumentation in legal reasoning. *Current Legal Theory*, pages 25–63, 1997.

[Sergot *et al.*, 1986] M.J. Sergot, F. Sadri, R.A. Kowalski, F. Kriwaczek, P. Hammond, and H.T. Cory. The British Nationality Act as a logic program. *Communications of the ACM*, 29:370–386, 1986.

[Sergot, 1988] M.J. Sergot. Representing legislation as logic programs. In J.E. Hayes, D. Michie, and J. Richards, editors, *Machine Intelligence*, volume 11, pages 209–260. Oxford University Press, Oxford, 1988.

[Simari and Loui, 1992] G.R. Simari and R.P. Loui. A mathematical treatment of defeasible argumentation and its implementation. *Artificial Intelligence*, 53:125–157, 1992.

[Simon, 1982] H. Simon. *Models of Bounded Rationality*, volume 2 (collected papers). MIT Press, Cambridge, MA, 1982.

[Skalak and Rissland, 1992] D.B. Skalak and E.L. Rissland. Arguments and cases. an inevitable intertwining. *Artificial Intelligence and Law*, 1:3–44, 1992.

[Toulmin, 1958] S.E. Toulmin. *The Uses of Argument*. Cambridge University Press, Cambridge, 1958.

[Verheij *et al.*, 1998] B. Verheij, J.C. Hage, and H.J. van der Herik. An integrated view on rules and principles. *Artificial Intelligence and Law*, 6:3–26, 1998.

[Verheij, 1996] B. Verheij. *Rules, reasons, arguments: formal studies of argumentation and defeat*. Doctoral dissertation University of Maastricht, 1996.

[Verheij, 1999] B. Verheij. Automated argument assistance for lawyers. In *Proceedings of the Seventh International Conference on Artificial Intelligence and Law*, pages 43–52, New York, 1999. ACM Press.

[Vreeswijk, 2000] G.A.W. Vreeswijk. Representation of formal dispute with a standing order. *Artificial Intelligence and Law*, 8:205–231, 2000.

[Walton and Krabbe, 1995] D.N. Walton and E.C.W. Krabbe. *Commitment in Dialogue. Basic Concepts of Interpersonal Reasoning.* State University of New York Press, Albany, NY, 1995.

[Winston, 1980] P.H. Winston. Learning and reasoning by analogy. *Communications of the ACM*, 23:689–703, 1980.