Evaluating an Approach to Reasoning with Cases Using Abstract Dialectical Frameworks

Latifa Al-Abdulkarim, Katie Atkinson and Trevor Bench-Capon

Department of Computer Sciences
University of Liverpool, Liverpool, UK

latifak@csc.liv.ac.uk

August 13, 2015

Abstract

Abstract Dialetical Frameworks (ADFs) are a recent development in computational argumentation which are, it has been suggested, a fruitful way of implementing factor based reasoning with legal cases. In this paper we evaluate this proposal, by reconstructing CATO using ADFs. We evaluate the ease of implementation, the efficacy of the resulting program, ease of refinement of the program, transparency of the reasoning, relation to formal argumentation techniques, and transferability across domains.

1 Introduction

A recent development in computational argumentation has been Abstract Dialectical Frameworks (ADFs) [11]. ADFs can be seen as a generalisation of standard Argumentation Frameworks (AFs) [16] in which the nodes represent statements rather than abstract arguments, and each node is associated with an acceptance condition which determines when a node is acceptable in terms of whether its children are acceptable. Thus links in AFs express only one relationship, namely *defeat*, but ADFs can represent a variety of attack and support relations. In consequence, whereas nodes in an AF have only the single acceptance condition that all their children are defeated, nodes in ADFs can have different acceptance conditions specifically tailored for each node. In [1] it was argued that ADFs are very suitable for representing factor based reasoning with legal cases as found in CATO [3] and as formalised in [20] and [18].

The key idea of [1] is that the abstract factor hierarchy of CATO [3] (an extract given in Figure 1) corresponds directly to the node and link structure of an ADF, or rather (since the links are labelled "+" or "−") a Prioritised ADF (PADF) [10], which partitions links into supporting and attacking links, and so corresponds to the labels on the links in the factor hierarchy. To express CATO's factor hierarchy as an ADF, acceptance conditions need to be supplied for each of the nodes. Finally the logical model of IBP [12] can be used to tie the various parts of the factor hierarchy together to supply decisions for particular cases. In [1] it was suggested that the acceptance conditions could be expressed as Prolog procedures. These could then be used directly to form a Prolog program which could be executed to classify cases represented as sets of factors, as in [20] and [18].

In this paper we will evaluate this approach. We will use US Trade secrets as the domain, allowing us to use the analysis of CATO which will permit direct comparison with CATO, IBP, the various systems used as comparators in [12] and the AGATHA system of Chorley [13], [14]. We will firstly consider a quantitative analysis, in terms of performance and how easily the program can be refined to improve performance, and then consider the system in terms of the transparency of its outputs, the relation to case decisions texts, and the relation to formal frameworks for structured argumentation such as [19]. Finally we will consider whether the method can be readily applied to other domains, by briefly describing an application of the ADF approach to *Popov v Hayashi* and related cases as modelled in [6].

2 Background

In this section we will recapitulate the essentials of ADFs [10], CATO [3] and IBP [12].

2.1 Abstract Dialectical Frameworks

An ADF is defined in [10] as:

> **Definition 1**: An ADF is a tuple $ADF = < S, L, C >$ where S is the set of statements (positions, nodes), L is a subset of $S \times S$, a set of links and $C = \{C_{s \in S}\}$ is a set of total functions $C_s : 2^{par(s)} \rightarrow \{t, f\}$, one for each statement s. C_s is called the acceptance condition of s.

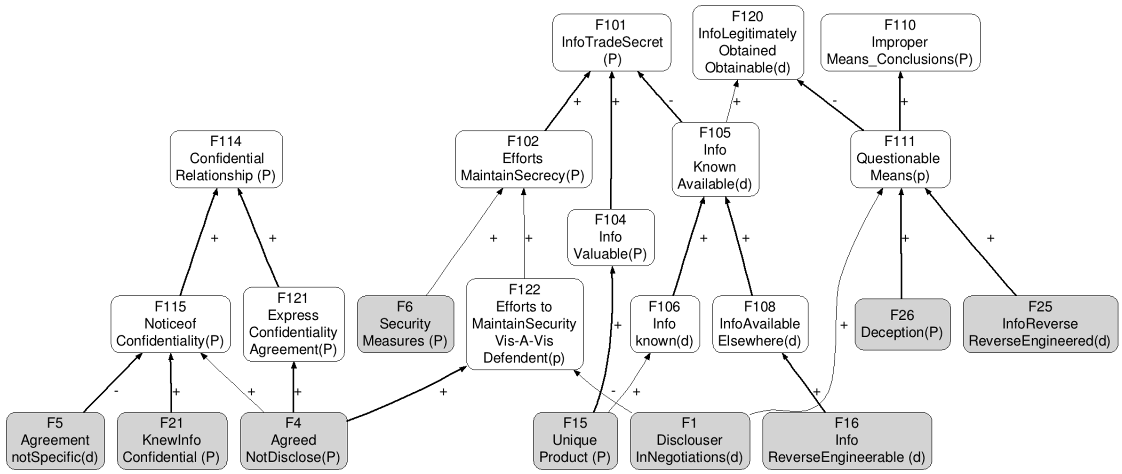

Figure 1: CATO Abstract Factor Hierarchy from [3]

In a Prioritised ADF, L is partitioned into $L+$ and $L-$, supporting and attacking links, respectively. Although the acceptance conditions are often expressed as propositional functions, this need not be the case: all that is required is the specification of conditions for the acceptance or rejection of a node in terms of the acceptance or rejection of its children.

2.2 CATO

CATO [3], which was developed from Rissland and Ashley's HYPO, most fully described in [4], takes as its domain US Trade Secret Law. CATO was primarily directed at law school students, and was intended to help them form better case-based arguments, in particular to improve their skills in distinguishing cases, and emphasising and downplaying distinctions. A core idea was to describe cases in terms of factors, legally significant abstractions of patterns of facts found in the cases, and to build these *base-level factors* into an hierarchy of increasing abstraction, moving upwards through intermediate concerns (abstract factors) to issues. An extract from the factor hierarchy is shown in Figure 1. Each factor favours either the plaintiff or the defendant. The program matches precedent cases with a current case to produce arguments in three plies: first a precedent with factors in common with the case under consideration is cited, suggesting a finding for one side. Then the other side cites precedents with factors in common with the current case but a decision for the other side as counter examples, and distinguishes the cited precedent by pointing to factors not shared by the precedent and current case. Finally the original side rebuts by downplaying distinctions, citing cases to prove that weaknesses are not fatal and distinguishing counter examples. CATO used twenty-six base level factors (there is no F9), as shown in Table 1.

ID	Factor
F1	DisclosureInNegotiations (d)
F2	BribeEmployee (p)
F3	EmployeeSoleDeveloper (d)
F4	AgreedNotToDisclose (p)
F5	AgreementNotSpecific (d)
F6	SecurityMeasures (p)
F7	BroughtTools (p)
F8	CompetitiveAdvantage (p)
F10	SecretsDisclosedOutsiders (d)
F11	VerticalKnowledge (d)
F12	OutsiderDisclosuresRestricted (p)
F13	NoncompetitionAgreement (p)
F14	RestrictedMaterialsUsed (p)
F15	UniqueProduct (p)
F16	InfoReverseEngineerable (d)
F17	InfoIndependentlyGenerated (d)
F18	IdenticalProducts (p)
F19	NoSecurityMeasures (d)
F20	InfoKnownToCompetitors (d)
F21	KnewInfoConfidential (p)
F22	InvasiveTechniques (p)
F23	WaiverOfConfidentiality (d)
F24	InfoObtainableElsewhere (d)
F25	InfoReverseEngineered (d)
F26	Deception (p)
F27	DisclosureInPublicForum (d)

Table 1: **Base Level Factors in CATO**

There is, however, no single root for the factor hierarchy as presented in [3]: rather we have a collection of hierarchies, each relating to a specific issue. To tie them together we turn to the Issue Based Prediction (IBP) system of Bruninghaus

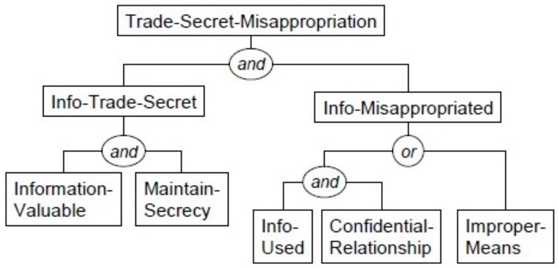

Figure 2: IBP Logical Model from [12]

and Ashley [12].

2.3 Issue Based Prediction

In IBP, which is firmly based on CATO, the aim is not simply to present arguments, but to predict the outcomes of cases. To enable this, the issues of CATO's hierarchy are tied together using a logical model derived from the *Uniform Trade Secret Act*, which has been adopted by forty seven States in the US, and the *Restatement of Torts*. The model is shown in Figure 2.

Now consider the factor hierarchy, part of which is shown in Figure 1. We can now regard this as an ADF by forming the set *S* from the issues, intermediate concerns and base level factors, $L+$ from the links labeled "+" and $L-$ from the links labeled "−". Using the complete factor hierarchy given in Figures 3.2 and 3.3 of [3] we will have an ADF which has as its leaf nodes the base level factors of CATO. This is described in tabular form in Table 2.

The roots of CATO's hierarchies correspond to the leaves of this logical model: we can therefore form them into a single ADF by using this structure. The relevant additions to the ADF needed to integrate the IBP model are shown in Table 3 (note that F124 is not discussed in [12]) .

IBP used 186 cases, 148 cases analysed for CATO and 38 analysed specifically for IBP. Unfortunately, these cases are not all publicly available and so we will use the 32 cases harvested from public sources by Alison Chorley and used to evaluate her AGATHA system [14], [13]. As part of the evaluation in [12], nine other systems were also considered to provide a comparison. Most of these were different forms of machine learning system, but programs representing CATO and HYPO were also included. IBP was the best performer: results reported in [12]

ID	S	$L+$	$L-$
F102	EffortstoMaintainSecrecy	F6, F122, F123	F19, F23, F27
F104	InfoValuable	F8, F15	F105
F105	InfoKnownOrAvailable	F106, F108	
F106	InfoKnown	F20, F27	F15, F123
F108	InfoAvailableElsewhere	F16, F24	
F110	ImproperMeans	F111	F120
F111	QuestionableMeans	F2, F14, F22, F26	F1, F17, F25
F112	InfoUsed	F7, F8, F18	F17
F114	ConfidentialRelationship	F115, F121	
F115	NoticeOfConfidentiality	F4, F13, F14, F21	F5, F23
F120	LegitimatelyObtainable	F105	F111
F121	ConfidentialityAgreement	F4	F23
F122	MaintainSecrecyDefendant	F4	F1
F123	MaintainSecrecyOutsiders	F12	F10
F124	DefendantOwnershipRights	F3	

Table 2: **CATO as ADF**

for the IBP, Naive Bayes (the best performer of the ML systems), CATO, HYPO and a version of IBP which uses only the model, with no CBR component, are shown in Table 4[1]:

Direct comparison with AGATHA is hampered by the fact that evaluation in AGATHA was directed towards evaluating the different heuristics and search algorithms used in that system, and so no version can be considered "definitive", and, of course, many fewer cases were used in the experiments. However, typically 27-30 of the 32 ($\approx 84-93\%$) cases were correctly decided by the theories produced by AGATHA [13].

3 Acceptance Conditions

We now supply acceptance conditions for each node, to supply the elements of C. We will rely only on the definitions of the factors in [3], as informed by the strong

[1]No explanation for using a different number of cases for CATO and IBP model is given in [12].

ID	S	$L+$	$L-$
F200	TradeSecretMisappropriation	F201,F203	F124
F201	Info-Miasappropriated	F110,F112,F114	
F203	Info-Trade-Secret	F102,F104	

Table 3: **IBP Logical Model as ADF**

	correct	error	abstain	accuracy
IBP	170	15	1	91.4
Naive Bayes	161	25	0	86.5
CATO	152	19	22	77.8
HYPO	127	9	50	68.3
IBP-model	99	15	38	72.6

Table 4: Results from [12]

and weak links shown in the figures. We do not use precedents at this stage; as Aleven remarks:

> for certain conflicts, it is self evident how they should be resolved. For example, the fact that plaintiff's product was unique in the market (F15) arguably supports a conclusion that plaintiff's information (which is used to make the product) is not known outside plaintiff's business (F106), but not if it is also known, for example, that plaintiff disclosed its information in a public forum (F27). Common sense dictates that in those circumstances, the information is known outside plaintiff's business. It is not necessary to look to past cases to support that point. CATO's use of link strength enables a knowledge engineer to encode inferences like this. (p47).

From Tables 2 and 3 we can see that we have eighteen nodes to provide with acceptance conditions.

One (F124) has only a single supporting child: thus the acceptance condition will be $Parent \longleftrightarrow Child$. We will write this (and the other acceptance conditions) as a set of tests for acceptance and rejection, to be applied in the order given, which allows us to express priority between them. The last test will always be a default. We choose this form of expression because we find it easier to read in many cases, because it corresponds directly to the defeasible rules with priorities used in formalisms such as ASPIC+ [19], and because it is directly usable as Prolog code. Thus we write *Parent* $\longleftrightarrow$ *Child* as

```
Accept Parent if Child.
Reject Parent.
```

Where NOT is required we use negation as failure. The tests are individually sufficient and collectively necessary, ensuring equivalence with the logical expression (see [15]).

Six nodes (F201, F203, F105, F108, F114 and F124) have only supporting links: these can be straightforwardly represented using AND and OR. We followed the IBP model for the two nodes taken from that model (F201 and F203), and used OR for the other four. The most complicated was InfoMisappropriated (F201):

```
Accept InfoMisappropriated if F114 AND F112.
Accept InfoMisappropriated if F110.
Reject InfoMisappropriated.
```

Five nodes have one supporting and one attacking link. These are best seen as forming an exception structure: accept (reject) the parent if and only if supporting (attacking) child unless attacking (supporting) child. Note that the exception may be the supporting *or* the attacking child: in the former case the default will be *reject*, and in the latter the default will be *accept*. Thus:

```
Accept Parent if Support AND (NOT Attack).
Reject Parent.

Reject Parent if Attack AND (NOT Support).
Accept Parent.
```

For F110, F120 and F121 the attacking child is the exception, while for F122 and F123 the supporting links are the exceptions. This leaves seven nodes. For F200 we regard the attacking link as an exception to the case where the conjunction of the supporting links holds:

```
Accept Trade Secret Misappropriation  if
   Info Trade  Secret AND
```

```
    Info Misappropriated AND
    (NOT Defendant Ownership Rights).
Reject Trade Secret Misappropriation.
```

For F104 and F112 we see the supporting links as offering disjoint ways of accepting the parent, and the attacking child as a way of establishing that the factor is not present. We default to *yes* because in many cases there are no factors for either side present relating to this point. We take it that this factor was often simply accepted on the facts and uncontested, and so there was no discussion on the point. Thus for F112:

```
Accept F112 if F18.
Accept F112 if F8.
Accept F112 if F7
Reject F112 if F17.
Accept F112.
```

The remaining four are more complicated because they involve more factors. Here our approach is to make an initial attempt to supply tests, but remain prepared to make any necessary adjustments. For F106 (*InfoKnown*) we use

```
Accept F106 if F20.
Accept F106 if F27 AND (NOT F15).
Accept F106 if F27 AND (NOT F123).
Reject F106.
```

The rationale is that if the information is known to competitors, it is known, but even if it is disclosed in a public forum, the uniqueness of the product can suggest that the disclosure had no impact (i.e it was not sufficiently widely known), and so the secret remained effectively unknown (and so F24 (Information obtainable elsewhere) is more appropriate). The third clause suggests that the public disclosure might be restricted (e.g. if the secret was disclosed in a court of law during a trade secrets hearing), so that the information may be known, but embargoed.

For F115, we regard each of the four supporting links (F4, F13, F14 and F21) as distinct ways of establishing notice of confidentiality. F23 (waiver of confidentiality) is an exception to each of them whereas F5 (agreement is not specific) is an exception only to F4 and F13, since the other two do not relate to agreements. Similarly for F111 (Questionable Means) we see the supporting links as four different ways in which questionable means can be established. The attacking links here seem like counter claims rather than exceptions, and suggest that for this node we may need to explore precedents to identify preferences. However, as a first attempt we will regard them as three distinct ways of rejecting the claim. We thus have seven clauses, one for each factor, which we initially order as F17, F25, F22, F14, F26, F1, F2, to reflect the strong and weak links shown in CATO.

This leaves F102, Efforts to Maintain Secrecy, which has three supporting and three attacking links. F19 is applicable only if no security measures at all are taken, which suggests that it has priority. Similarly a waiver of confidentiality (F23) or disclosure in a public forum (F27) could be seen as negating any efforts to maintain secrecy, although F123 provides a possible exception to the latter. The remaining two supporting links we also regard as independent. Thus here we have six clauses, offering reasons to reject or accept, ordered F19, F23 (F27 and NOT F123), F6, F122, with *reject* as the default.

4 Prolog Program

The Prolog [2] program was formed by ascending the ADF, rewriting the acceptance conditions as groups of Prolog clauses to determine the acceptability of each node in terms of its children. This requires restating the tests using the appropriate syntax, adding some reporting to indicate whether the node is satisfied (defaults are indicated by the use of "accepted that"), and some control to call the procedure to determine the next node, and to maintain a list of accepted factors. Thus the Prolog for F112 (for which the acceptance conditions were given earlier) is:

```
% determine acceptability of
% F112, Information used
getf112(Case,FactorsSoFar):-
      member(f18,FactorsSoFar),
      write([the,information,was,used]),
      nl, getf111(Case,[f112|FactorsSoFar]).
getf112(Case,FactorsSoFar):-
     member(f8,FactorsSoFar),
     write([the,information,was,used]),
     nl, getf111(Case,[f112|FactorsSoFar]).
getf112(Case,FactorsSoFar):-
```

[2] Prolog was used because of its closeness to the acceptance conditions, made the implementaion quick, easy and transparent.

```
    member(f7,FactorsSoFar),
    write([the,information,was,used]),
    nl, getf111(Case,[f112|FactorsSoFar]).
getf112(Case,FactorsSoFar):-
    member(f17,FactorsSoFar),
    write([the,information,was,not,used]),
    nl, getf111(Case,FactorsSoFar).
getf112(Case,FactorsSoFar):-
      write([accepted,that,
             the,information,was,used]),
      nl, getf111(Case,[f112|FactorsSoFar]).
```

Each of the four tests in the acceptance condition is applied in a separate clause, using the set of factors currently identified as present in the case, before proceeding to the next factor (F111), with F112 added to the applicable factors if it is accepted. To allow completion of the database [15], a final clause is added to catch any case not covered by any of the preceding clauses. These defaults may favour either side. In some cases, such as F112 and F104 we decided that the default should be *accept* because in most cases there were no factors in the case descriptions relating to these abstract factors, and yet they are a *sine qua non* for any claim. Our belief is that these aspects were uncontested and so the factors were not explicitly discussed in the trial, and so do not appear in the CATO analysis. Where we felt it was clear that the factor needed to be explicitly established (e.g. F106 (Information Known) and F111(Questionable Means)) the default was *reject*.

The above demonstrates that it is a straightforward and reasonably objective process to transform a factor based analysis such as is found in [3] to an executable program via an ADF. Although judgement was sometimes required to form the acceptance conditions, we would suggest that such judgements were not difficult to make. Moreover if there are difficult choices, the effect of the alternatives can be compared on a set of test cases. Overall the relatively small number of factors relevant to particular nodes, greatly simplifies the task.

5 Results

We can now run the program on the cases. We represent the cases as a list of base-level factors. For example, the Boeing case[3] is represented as

```
case(boeing,[f4,f6,f12,f14,f21,f1,f10]).
```

giving output:

```
1 ?- go(boeing).
accepted that defendant is not owner of secret
efforts made vis a vis outsiders
efforts made vis a vis defendant
there was a confidentiality agreement
defendant was on notice of confidentiality
there was a confidential relationship
accepted that the information was used
questionable means were used
accepted that the information
    was not available elsewhere
accepted that information is not known
accepted that the information
    was neither known nor available
accepted that the information was valuable
not accepted that the information
    was legitimately obtained
improper means were used
efforts were taken to maintain secrecy
information was a trade secret
a trade secret was misappropriated
find for plaintiff
boeing[f200, f201, f203, f102,f110,f104,f111,
       f112,f114,f115,f121,f122,f123,
       f4,f6,f12,f14,f21,f1,f10]
decision is correct
```

The initial program correctly classified 25 out of the 32 cases (78.1%) of the cases. While all ten of the cases won by the defendant were correctly classified,

[3] The Boeing Company v. Sierracin Corporation, 108 Wash.2d 38, 738 P.2d 665 (1987).

seven of the 22 cases won by the plaintiff were not. The figure for correct answers is remarkably close to the 77.8% reported for the version of CATO used in [12], which, of course, uses exactly the same analyisis of the domain and cases that we have adopted here. Thus as a first conclusion we can tentatively suggest that executing the analysis in [3] as an ADF produces very similar results to those obtainable using the original CATO program (albeit we are using a smaller set of cases). We can now investigate how the initial program might be improved.

The wrongly predicted cases were:

```
case(spaceAero,[f8,f15,f18,f1,f19]).
case(televation, [f6,f12,f15,f18,f21,f10,f16]).
case(goldberg,[f1,f10,f21, f27]).
case(kg,[f6,f14,f15,f18,f21,f16,f25]).
case(mason,[f6,f15,f21,f1,f16]).
case(mineralDeposits,[f18,f1,f16,f25]).
case(technicon,[f6,f12,f14,f21,f10,f16,f25]).
```

Examination of the cases showed that five of the seven had F16 (ReverseEngineerable) present and that these cases were the only cases found for the plaintiff with F16 present. The problem in these five cases is that the program finds for the defendant because the information is available elsewhere (F105). This is established by the presence of ReverseEngineerable and is unchallengeable. Examination of the ADF shows that F16 is immediately decisive: if that factor is present, there is no way the plaintiff can demonstrate that the information is a trade secret. *Goldberg*[4] also fails through F105 (information known or available), since disclosure in a public forum (F27) is sufficient to deny the information trade secret status. It would appear that we could significantly improve performance by refining this branch to allow the plaintiff some way to defend against, in particular, F16.

5.1 Refinement

At this point we should observe that CATO is likely to be more robust in the face of imperfect analysis than an approach based on a logical model. Because CATO generates arguments based on considering all the available factors taken together, it is less likely to have the outcome determined by a single factor than a logical model, for which, for example, the presence of F16 or F24 can be seen

[4]Goldberg v. Medtronic, 686 F.2d 1219 (7th Cir. 1982).

to immediately determine a decision for the defendant. Moreover, CATO was designed to assist law students, not to predict outcomes. We should expect similar problems to arise in IBP, which also uses a logical model, albeit one that is applied at a later stage of the process. In [12] it is stated

> We found that some Factors, called KO-Factors (or Knockout Factors), almost always dominate the outcome of a case. For instance, as an empirical matter, the plaintiff will not win a case with Factor F20, Info-Known.

Such factors are given special treatment in IBP, and so it does not seem unreasonable to use the initial results to suggest possible refinements to our original analysis. First we consider *Goldberg v Medtronic*. In that decision it is explicitly stated that

> The district court found that Medtronic could not avoid its obligation of confidence due to the availability of lawful means of obtaining the concept when those means were not employed. We affirm.

Thus the factor which was decisive[5] for our progam, F27, was in fact explicitly held to be insufficient in the actual decision. Whether this decision was correct or not is not for us to say, but it does explain why our program misidentified the case. Assuming the decision to be correct, we should either redefine F27 to include the defendant's actual *use* of this public domain knowledge, so that it is not present in *Goldberg*, or allow F21 (Knew information confidential) as an exception to F27 in determining the acceptance of F106, on the grounds that if the defendant believed the information to be confidential, he could not have been aware that the information was publicly available. Since we have no other case with F21 and F27 both present, we cannot choose between these two solutions on the precedents available to us.

We now turn to the problem created by F16, reverse engineerable. In [3] the note on the applicability of this factor states:

> The factor applies if: Plaintiff's information could be ascertained by reverse engineering, that is, by inspecting or analyzing plaintiff's product (regardless of whether defendant actually obtained the information in this way)

[5]Running a version of *Goldberg* without F27 finds for the plaintiff.

Thus it is clear that we cannot use the defence of Goldberg, that the defendant did not in fact reverse engineer the information. None the less, the ease with which the product could be reverse engineered does need to be considered. Thus in Mason we read:

> In this regard, we note that courts have protected information as a trade secret despite evidence that such information could be easily duplicated by others competent in the given field.

citing *KFC v Marion Kay* and *Sperry Rand v Rothlein*. The KFC decision cited in Mason states

> Marion-Kay maintains that the recipes and formulas for the making of KFC seasoning are not unique and that Marion-Kay is capable, both financially and technically, of producing KFC seasoning.

which suggests that the uniqueness of the product (F15) might be a factor capable of attacking the acceptability of F108 as well as F106 (as identified in CATO). Adding F15 as an exception to F16 would give the correct decision in *Televation*, *KG* and *Mason*. In Technicon we find that the phrase "readily ascertainable" is used:

> Curtis claimed that Technicon's trade secrets were "readily ascertainable" and that the company had not made reasonable attempts to ensure its trade secrets. The Court reasoned that Bridgmon's "wire-tap" process had required over two-thousand hours, and still had not yielded a fully functional product. The Court held that this amount of time indicated that a trade secret was not readily ascertainable.

In fact in two of the cases (*KG* and *Technicon*) restricted material was used by the defendants, strongly implying that the information was not, in fact, readily ascertainable. In *Mineral Deposits*, we find that

> After Zigan received the spiral, he removed the label which indicated that patent applications were pending and gave the spiral to defendant Zbikowski. Zbikowski then cut the spiral into pieces, made molds of the components, and proceeded to manufacture copies of the spiral. If a trade secret is divulged under an express or implied restriction of nondisclosure or nonuse, a party who engages in unauthorized use of the information will be liable in damages to the owner of the trade secret.

which strongly suggests to us that F14 was also in fact present in this case. Moreover in *Televation*, whether the secret counted as reverse engineerable was contested:

> The mere fact, however, that a competitor could, through reverse engineering, duplicate plaintiff's product does not preclude a finding that plaintiff's techniques or schematics were trade secrets, particularly where, as here, the evidence demonstrated that the reverse engineering process would be time-consuming.

and there is a strong suggestion that the court in fact believed that copies of the plaintiff's drawings had, in fact been used by the defendant, which would mean that F14 would apply. Finally Sperry Rand, another decision cited in *Mason*, states

> The defendants claim that there is no trade secret if it is disclosed by prior art or if it is readily discernible by others skilled in the field. It is no defense in an action of this kind that the process in question could have been developed independently, without resort to information gleaned from the confidential relationship. As stated in the landmark case of Tabor v. Hoffman, 118 N.Y. 30, 35, 23 N.E. 12, 13 (1889): "Even if resort to the patterns of the plaintiff was more of a convenience than a necessity, still if there was a secret, it belonged to him, and the defendant had no right to obtain it by unfair means, or to use it after it was thus obtained."

which suggests that the use of any kind of questionable means (rather than just F14) could be used to block a defence relying on reverse engineerability.

The decisions thus give a number of suggestions for exceptions to F16 as a support for F108: especially uniqueness of the product and use of restricted materials. Incorporating those exceptions would raise success of our program to 29 out of 32 (90.6%), and removing F27 from *Goldberg* (or allowing F21 as an exception) and adding F14 to *Mineral Deposits*, both of which seem eminently justifiable on the facts of the cases concerned, would give correct decisions in these cases also (96.8%).

This leaves only *Space Aero* as an unexplained failure. The output for this case is:

```
1 ?- go(spaceAero).
accepted that defendant is not owner of secret
```

```
accepted that efforts made vis a vis outsiders
no efforts made vis a vis defendant
accepted that there was
    no confidentiality agreement
accepted that defendant was
     not on notice of confidentiality
accepted that there was
     no confidential relationship
the information was used
questionable means were not used
accepted that the information
        was not available elsewhere
accepted that information is not known
accepted that the information was
  neither known nor available
the information was valuable
not accepted that the information
   was legitimately obtained
accepted that improper means were not used
no efforts were taken to maintain secrecy
information was not accepted as a trade secret
no trade secret was misappropriated
find for defendant
spaceAero[f104,f112,f123,
               f8,f15,f18,f1,f19]
decision is wrong
```

This case fails on two branches: the information is not a trade secret because no security measures were taken, and because it appears that no confidential relationship existed. A key feature of this case is that the defendants were former employees of the plaintiff, and had been provided with the disputed information when employed by the plaintiff because they needed it to carry out their duties. The decision itself states

> The testimony, taken as a whole, convinces us that Darling took precautions to guard the secrecy of its process which, under the circumstances, were reasonably sufficient.

This suggests to us that F19 (no security measures) was not, in fact, accepted as present, and removing this factor from the case is enough to establish that there was a trade secret. Turning to the issue of confidentiality we read

> While none of the former employees had signed a contract with Darling in which they formally agreed not to use the information acquired by them, and while they were free to leave their employment at will, Judge Pugh found that they owed the duty of fidelity to their employer while they were employed. We agree. ... The court below found as a fact that some of the former employees had in their possession, after leaving Darling's employment, certain sketches of oxygen breathing hoses which they had taken while they were employed by Darling, without Darling's knowledge. ... the former employees knew that they were acting wrongfully in violation of their confidential relationship and their duty of loyalty. We agree with the court below that the former employees violated the duty of fidelity and trust which they owed to Darling in respect of the trade secret and that their conduct was such as to entitle Darling to the protection of a court of equity.

Again we cannot comment on whether this decision was correctly made or not. However, it does seem that F21 (Knew Information Confidential) should be included. If this is added, we can establish a confidential relationship and find for the plaintiff.

5.2 Discussion

The previous section shows that by using the ADF we can readily explain the points at which the acceptability conditions do not concur with the decisions taken in the actual cases. We can then return to the original decisions and use them to determine possible refinements to the representation. In some cases, the problem seems to lie with the attribution of the factors. Should *Goldberg* really have F27? Should *Mineral Deposits* have F14? Should *Space Aero* include F14 or F21 and exclude F19? Such matters were contested in the actual case, and ascribing the presence or absence of particular factors requires interpretation of the case by the analyst. The interpretation cannot be disputed without descending to the level of facts as advocated by [5] and [1]. Addition of the fact layer has been the subject of work subsequent to that reported here.

As well as disputed factors, a decision like *Goldberg* suggests that we may wish to modify the description of factors intended to guide the analyst. In that decision it was suggested that to count for the defendant the information not only had to be publicly available but that the public source needed to be known to and *used by the defendant*, which would narrow the applicability of F27 as described in [3].

Adding or removing a factor to or from a particular case provides a local solution which will solve a problem with a particular case. Our results, however, indicated a general problem which was applicable to several cases: the dominant affect of F16, reverse engineerable. It seemed clear to us that the presence of F16 should not by itself be sufficient for a finding for the defendant. Again the decisions themselves suggested several possible ways of arguing against F16: in particular the use of restricted materials and the uniqueness of the product. Either or both of these exceptions could be incorporated in the ADF without adversely affecting any currently available cases, but we would need to have a reasonably large set of new cases in order to evaluate the different solutions and to guard against over fitting.

Finally it should be conceded that the decisions themselves may be erroneous. Assuming that there are least some poor decisions which we would not wish to serve as precedents, we should be willing to tolerate a certain number of divergences from our results.

To summarise:

- Simply translating the analysis of [3] into an ADF and executing the resulting program gave results almost identical to those found for CATO in the IBP experiments reported in [12]. Note that this is achieved without need for balancing of pro and con factors central to existing case based reasoning systems.

- The reasons for the "incorrect" decisions can be readily identified from the output and the ADF, as we saw from the discussion of the wrongly decided cases above.

- Examination of the texts of the decisions readily explained why the results diverged, and suggested ways in which the analysis could be improved, either at the case level by changing the factors attributed, or at the domain level by including additional supporting or attacking links.

From this we conclude that use of ADFs provides good performance, and has a number of positive features from a software engineering (and domain analysis) standpoint, which would enable the ADF to be refined and performance improved. We also believe that we do need to include a fact layer to permit increased transparency in the ascription of factors to cases.

6 Quality of Decisions

As the Prolog program proceeds it reports on the acceptability or otherwise of the various abstract factors and the resolution of issues. As shown above, this provides an excellent diagnostic for divergent decisions, but how does it measure up the the actual decisions found in cases?

Of course, without facts, we will not be able to follow the decision very closely. But consider a reordering of the elements of our decision for, say *Boeing*. We also omit some elements, and add a little linking text. Recall too that we wrote the program used thus far to "decide" the cases: in a version to supply explanation we would want to customise the text reports to indicate the particular clause being used for a node by giving the base level factors used. Below is what a decision might look like: we show the current program output in boldface, possible clause-specific customisations in italics and linking text in ordinary font.

> We **find for plaintiff**. The **information was a trade secret**: **efforts were taken to maintain secrecy**, *since disclosures to outsiders were restricted and the defendant entered into a non-disclosure agreement and other security measures were applied.* **The information was unique**. It is **accepted that the information was valuable** and it is **accepted that the information was neither known nor available**.
>
> **A trade secret was misappropriated**: **there was a confidential relationship** *since the defendant entered into a non-disclosure agreement* and it is **accepted that the information was used**.
>
> Moreover **improper means were used** *since the defendant used restricted materials*.

This seems to have the makings of a reasonable summary of the decision. There are two problems: it does not indicate what the defendant contended, since the clauses of the program which were not reached do not feature in the report, and, of course, the facts on which the finding are based are not present. None the less, we find the output a distinct improvement on previous work such as [13]. We believe that the output from the current program could be readily used to drive a program of the sort envisaged by Branting [9], and that this will become even more useful when we have added a fact layer to allow the explanation of the attribution of factors. What is missing, however, is the citation of precedents which are such an important feature of real decisions: we will consider this aspect in the next subsection.

6.1 Portions of Precedents

Using the ADF approach, we do not see confrontations between large sets of pro and con factors covering the whole case. Instead factors are opposed to one another in the context of accepting or rejecting particular nodes, and so will represent a specific point in the argument. Thus two cases may be identical with respect to the factors determining whether a relationship was confidential, while very different with respect to whether the information was a trade secret. Two points are significant here: first that some apparent distinctions are insignificant, since they relate to different issues: this was partly what the factor hierarchy was introduced in CATO to address. More importantly, however, a precedent might be citable to establish that a confidential relationship existed, even though the case as a whole was found for the defendant, because of some other issue (such as that the information was not, in fact, a trade secret). For this reason it is sometimes desirable to be able to reason with portions of precedent, as urged by Branting in [8]:

> This paper argues that the task of matching in case-based reasoning can often be improved by comparing new cases to portions of precedents. An example is presented that illustrates how combining portions of multiple precedents can permit new cases to be resolved that would be indeterminate if new cases could only be compared to entire precedents.

This is borne out by a reading of the decisions in the various cases: rarely do they begin with a precedent and then discuss similarities and differences, but rather they use precedents at particular points of the decisions to identify questions and issues to be addressed, and to justify answers and consequences. This is effectively what is done in the ADF approach: competing factors are considered in the context of accepting or rejecting a particular node.

Part of the output of the program is, for each case, the set of nodes satisfied. This information could be used to find the precedents needed to make particular points. For example we could retrieve all cases where the defendant had agreed not to disclose (F4) and yet efforts to maintain secrecy (F102) were not established. This query would return *CMI*, and so we can cite *CMI* as a precedent when arguing (in the context of a case containing F4 and F27) that the efforts taken to maintain secrecy were insufficient to establish the information as a trade secret. Matching at this level of granularity will direct us to the precedents most relevant to the specific point we need to argue. Moreover, such precedents can then be incorporated in our decisions to justify the acceptance or rejection of particular nodes, which is close to the way they are used in practice, and which corresponds to the downplaying and emphasis of distinctions which are such an important feature of CATO.

Another reason to consider portions of precedents is provided by [18]. In that paper *a fortiori* reasoning was explained in terms of identifying a rule using only a subset of factors available for the winning side preferable to the rule using all the factors available to the losing side. That paper gave, however, no indication of how this subset should be chosen. The output from the ADF in contrast, does show which factors were instrumental and active in winning the case.

6.2 Relation to Structured Argumentation

The overall output of our system bears a strong resemblance to the kind of structured argumentation found in ASPIC+ [19]. The union of the acceptance conditions can be seen as the underlying knowledge base. Determining the acceptance or rejection of the various nodes produces sub-arguments, and these can be linked to produce the argument for finding for the plaintiff (or defendant) which follows the argument-subargument structure of ASPIC+ arguments. There are also differences: because the ordering of clauses expresses priority between arguments only the winning arguments are generated. Thus the output does not include all arguments, but only the winning line of argument. Where, however, potential attackers are children rather than siblings, but are not acceptable, this is reported. Thus although there are correspondences, especially through the argument-subargument structure, the control regime employed by the program means that there are also important differences. These relate mainly to conflicts: the output puts only the winning side of the case, and does not provide a good record of the rejected arguments available to the losing side. None the less, the correspondences are such that this is worth exploring further, using a different control regime in the implementation to generate all arguments, as a potentially fruitful way of formalising the reasoning.

6.3 Application to a Second Domain

In the above we have considered the approach with respect to a single domain. If the approach is to be of general significance, however, it needs to be applicable to other domains. In this subsection we will describe a further exercise designed to show that the approach is more generally applicable. We will apply the method to another domain, one which has often been used as an illustration of factor-based reasoning as that analyses are available: the wild animals cases and *Popov v Hayashi*. The wild animals cases were introduced into AI and Law in [7] and extended to the baseball case of *Popov* in [21]. We will use the factor-based analysis of [6] as our starting point.

Briefly the wild animals cases concern plaintiffs chasing wild animals when their pursuit was interrupted by the defendant. Post was chasing a fox for sport. Keeble was hunting ducks, Young fish and Ghen a whale, all in pursuit of their livelihoods. *Popov v Hayashi* concerned disputed ownership of a baseball (valuable because it had been hit by Barry Bonds to break a home run record). Popov had almost completed his catch when he was assaulted by a mob of fellow spectators and Hayashi (who had not taken part in the assault) ended up with the baseball when it came free. The wild animals cases were cited when considering whether Popov's efforts had given him possession of the ball.

Thirteen, base-level, factors are identified in [6]. The first task is to form them (together with appropriate abstract factors) into a factor hierarchy, to use as the node and link structure of our ADF. This factor hierarchy is shown in Figure 3: some adaptations have been made: for example we include a factor *Res* (Residence Status) to indicate the attachment of the animals to the land, since it appears to make a difference whether they are there permanently, seasonally, habitually, occasionally, or whatever. The nodes and links are given in Table 5.

S	L+	L-
Decide	Ownership,RightToPursue,IllegalAct	NoBlame
Capture	HotPursuit, Vermin	NotCaught
Ownership	Convention, Capture, OwnsLand, Res	
PMotive	Pliving, PSport, PGain	DLiving
DMotive	DLiving, DSport, DGain	Malice
OwnsLand	LegalOwner	
RightToPursue	OwnsLand, Pmotive, HotPursuit	DMotive
AntiSocial	Nuisance, Impolite	DMotive
Trespass	LegalOwner, AntiSocial	
IllegalAct	Assault, Trespass	

Table 5: **Popov as ADF**

We now supply acceptance conditions for the nine non-leaf nodes.

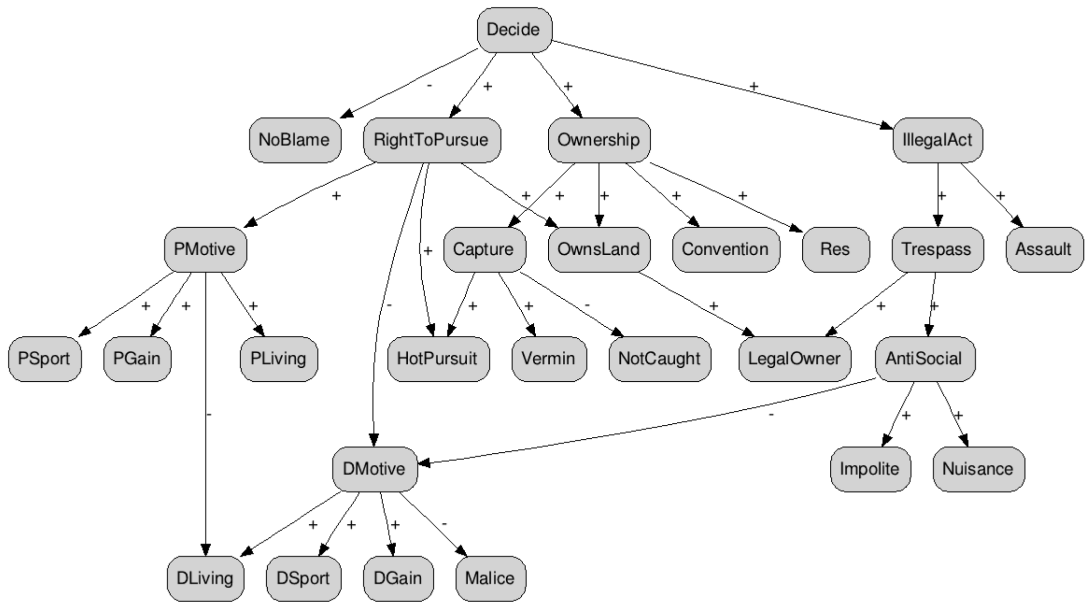

Figure 3: Factor Hierarchy/ADF for Popov

1. Decide for Plaintiff if NOT (NoBlame) AND (Ownership OR (RightToPursue AND IllegalAct))
2. Ownership if (OwnsLand AND Resident) OR Convention OR Capture
3. Capture if NOT (NotCaught) OR (Vermin and HotPursuit)
4. RightToPursue if OwnsLand OR (HotPursuit AND PMotive AND (NOT (better) DMotive))
5. PMotive if PLiving OR (PSport OR PGain) AND (NOT DLiving)
6. DMotive if NOT Malice AND (DLiving OR DSport OR DGain)
7. IllegalAct if Trespass OR Assault
8. Trespass if LegalOwner AND AntiSocial
9. AntiSocial if (Nuisance OR Impolite) AND (NOT DLiving)

The only real controversy here is with the determination of *Right to Pursue* when both the plaintiff and the defendant have good motives. Essentially we want to say that if the land is not owned by one of them, the right to pursue is given to the party with the better motive. The remainder seem fairly uncontroversial.

The acceptance conditions can easily be expressed as Prolog procedures and then embedded in code as was done for CATO. We can now execute the program. Running the case for *Young v Hitchens* produces the output (note that the program abbreviates factor names):

```
1 ?- go(young).
the plaintiff had not captured the quarry
the plaintiff did not own the quarry
plaitiff has good motive
defendant has good motive
plainiff did not own the land
plainiff had a right to pursue the quarry
defendant committed no antisocial acts
defendant committed no trespass
no illegal act was committed
do not find for the plaintiff
find for the defendant
young[rtToPursue,dMotive,pMotive,
              nc,hp,imp,pliv,dliv]
```

We produce correct results from all five cases discussed in [6], and on this basis we believe that the ADF representation can be used to encapsulate the knowledge of the domain as represented in [6], suggesting that the method can be applied straightforwardly to a second domain. In general we believe that the method can be applied to any domain for which factor based reasoning in the CATO (or HYPO or IBP) style is appropriate. This has encouraged us sufficiently to attempt to apply the method to a larger scale problem in the domain of the US automobile exception to the fourth amendment rule for which there is no accepted analysis into factors available, so we that need to start from the case decision texts: we will also incorporate a fact layer in this domain. This is the subject of the next stage of our project.

7 Concluding Remarks

In this paper we have evaluated an approach to reasoning with legal cases described in terms of factors using Abstract Dialectical Frameworks, as described and advocated in [1]. We find that:

- The success of the implementation depends to a large extent on the quality of the analysis. Making a direct translation of the analysis of CATO [3] yields a success rate almost identical to that found for CATO in [12]. This is a creditable 78.1% of the cases decided "correctly".

- The ADF does, however, provide very transparent output that identifies precisely where the outcomes suggested by the implementation diverge from the actual outcomes. Now reading the original decision texts suggests one of four solutions. These are, in ascending order of divergence from the original analysis:

  1. Removing a factor wrongly attributed to the case
  2. Adding a factor wrongly omitted from the case
  3. Modifying an acceptance condition: e.g. changing the priorities
  4. Modifying the ADF: e.g. adding a supporting or attacking node for the problem node.

  Often several of these modifications can potentially solve the problem, and the choice is made according to the context provided by the other divergent cases we are trying to accommodate.

- The ADF approach provides a good way of using a set of test cases to refine an initial analysis.

- While the output from the program provided good diagnostics and a reasonable explanation of the result, it will need post-processing to put it into a form resembling real decisions. Promising techniques for post-processing exist (e.g. [9]). Our output does, however, currently lack the citations and facts which are prominent in actual decisions.

- The method emphasises reasoning with portions of precedents, rather than whole cases. We believe that this does correspond to legal practice as manifest in real decisions. Moreover the output from the current program would be suitable to identify relevant precedents when considering a case, and to provide a source of citations for incorporation in decisions. This could also provide a potential database to support conceptual retrieval of cases, which has been an important issue in AI and Law since its very beginnings (see, e.g. [17]).

- Our approach has clear correspondences with structured argumentation [19]. These merit further exploration.
- The method can be applied to different domains. We believe that any domain for which factor based reasoning would be appropriate would be amenable to this method.

We find all of this encouraging. The next important step will be to extend the method to the fact level, so as to permit argument about the ascription of factors, and to be able to ground our explanations in the particular facts of a case. Also worthy of exploration is producing a variant program to generate all the possible arguments and resolve conflicts explicitly, so as to facilitate comparison with ASPIC+, and another variant to enable conceptual retrieval of cases. Once the method has been extended to include the facts of particular cases at the lowest level of the ADF, a program to present the output in a form resembling the texts of actual decisions can also be considered.

APPENDIX: Popov as 2-Regular ADF

In [2] the notion of a *2-regular ADF* was introduced. A 2-regular ADF is an ADF, but every non-leaf node has *exactly two* children. This has certain advantages: it offers a regular, uniform structure, and it maximises the number of nodes, giving a good level of granularity. In this Appendix we rewrite the ADF shown in Figure 3 and Table 5 as a 2-regular ADF. The 2 regular ADF is shown in Table 6. A fourth column has been added to indicate the nature of the acceptance conditions, as discussed in [2]. Children which are leaf nodes are marked (L), and those which are non leaf nodes are annotated with their node number.

A list of the leaf nodes (coincidently also 27) is:

1. OwnsLand
2. Whaling
3. Baseball
4. PhysicalPossession
5. CertainCapture
6. HotPursuit

7. Vermin
8. Residence
9. FrequentVisitor
10. StartedPursuit
11. Assault
12. DefOnLand
13. PMalice
14. Impolite
15. Nuisance
16. PLivelihood
17. POpportunistic
18. PSport
19. PAltruism
20. DMalice
21. DefNoFault
22. DLivelihood
23. DOpportunistic
24. DPleasure
25. DSport
26. DAltruism
27. PPleasure

Table 6: 2 Regular ADF for Popov and Wild Animals Cases

	Node	Elder	Younger	Type
1	Decide	Ownership (2)	RightToPusue (10)	Disjunction
2	Ownership	Capture (6)	ConsequentRight (3)	Disjunction
3	ConsequentRight	ByLand (4)	ByConvention (5)	Disjunction
4	ByLand	OwnsLand (L)	QuarryAssociated	Conjunction
5	ByConvention	Whaling (L)	Baseball (L)	Disjunction
6	Capture	Possession (7)	Pursuit (PP Min) (8)	Disjunction
7	Possession	PhysicalPossession (L)	CertainCapture (L)	Disjunction
8	Pursuit	HotPursuit (L)	Vermin (L)	Conjunction
9	QuarryAssociated	Residence (L)	FrequentVisitor (L)	Disjunction
10	RightToPusue	Pre-possessory interest (11)	DefBlameworthy (15)	Conjunction
11	Pre-possessory interest	FavouredPusuer (12)	ExclusiveRight (13)	Conjunction
12	FavouredPusuer	OwnsLand (L)	StartedPursuit (L)	Disjunction
13	ExclusiveRight	PMotiveOk (20)	FairCompetition (14)	Exception
14	FairCompetition	PMoney (21)	DMoney (26)	Conjunction
15	DefBlameworthy	IllegalAct (16)	DefBlameless (24)	Exception
16	IllegalAct	Criminal (17)	AntiSocial (19)	Disjunction
17	Criminal	Trespass (18)	Assault (L)	Disjunction
18	Trespass	OwnsLand (L)	DefOnLand (L)	Conjunction
19	AntiSocial	Impolite (L)	Nuisance (L)	Disjunction
20	PMotiveOk	PMoney (21)	PPleasure (22)	Disjunction
21	PMoney	PLivelihood (L0	POpportunistic (L)	Disjunction
22	PPleasure	PSport (L)	PAltruism (L)	Disjunction
23	DefGoodMotive	DMotiveOk (25)	DMalice (L)	Rebuttal
24	DefBlameless	NoFault (L)	DefGoodMotive (23)	Disjunction
25	DMotiveOk	DMoney (26)	DPleasure (27)	Disjunction
26	DMoney	DLivelihood (L)	DOpportunistic (L)	Disjunction
27	DPleasure	DSport (L)	DAltruism (L)	Disjunction

References

[1] L. Al-Abdulkarim, K. Atkinson, and T. Bench-Capon. Abstract dialectical frameworks for legal reasoning. In *Proceedings of Jurix 2014*, pages 61–70, 2014.

[2] L. Al-Abdulkarim, K. Atkinson, and T. Bench-Capon. Factors, issues and values: Revisiting reasoning with cases. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Law*, pages 3–12, 2015.

[3] V. Aleven. *Teaching case-based argumentation through a model and examples*. PhD thesis, University of Pittsburgh, 1997.

[4] K. Ashley. *Modelling Legal Argument: Reasoning with Cases and Hypotheticals*. MIT Press, Cambridge, MA, 1990.

[5] K. Atkinson, T. Bench-Capon, H. Prakken, and A. Wyner. Argumentation schemes for reasoning about factors with dimensions. *Proceedings of JURIX 2013*, pages 39–48, 2013.

[6] T. Bench-Capon. Representing popov v hayashi with dimensions and factors. *Artif. Intell. Law*, 20(1):15–35, 2012.

[7] D. Berman and C. Hafner. Representing teleological structure in case-based legal reasoning: The missing link. In *Proceedings of the Fourth International Conference on Artificial intelligence and Law*, pages 50–59, 1993.

[8] L. K. Branting. Reasoning with portions of precedents. In *Proceedings of the 3rd ICAIL*, pages 145–154. ACM, 1991.

[9] L. K. Branting. An issue-oriented approach to judicial document assembly. In *Proceedings of the 4th ICAIL*, pages 228–235. ACM, 1993.

[10] G. Brewka, H. Strass, S. Ellmauthaler, J. Wallner, and S. Woltran. Abstract dialectical frameworks revisited. In *Proceedings of 23rd IJCAI*, 2013, issued on CD.

[11] G. Brewka and S. Woltran. Abstract dialectical frameworks. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference*, 2010.

[12] S. Brüninghaus and K. Ashley. Predicting outcomes of case-based legal arguments. In *Proceedings of the 9th ICAIL*, pages 233–242, 2003.

[13] A. Chorley and T. Bench-Capon. Agatha: Using heuristic search to automate the construction of case law theories. *Artificial Intelligence and Law*, 13(1):9–51, 2005.

[14] A. Chorley and T. Bench-Capon. An empirical investigation of reasoning with legal cases through theory construction and application. *Artif. Intell. Law*, 13(3-4):323–371, 2005.

[15] K. L. Clark. Negation as failure. In *Logic and data bases*, pages 293–322. Springer, 1978.

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

[17] C. D. Hafner. Conceptual organization of case law knowledge bases. In *Proceedings of the 1st ICAIL*, pages 35–42, 1987.

[18] J. Horty and T. Bench-Capon. A factor-based definition of precedential constraint. *Artif. Intell. Law*, 20(2):181–214, 2012.

[19] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

[20] H. Prakken and G. Sartor. Modelling reasoning with precedents in a formal dialogue game. *Artif. Intell. Law*, 6(2-4):231–287, 1998.

[21] A. Z. Wyner, T. J. M. Bench-Capon, and K. Atkinson. Arguments, values and baseballs: Representation of popov v. hayashi. In *JURIX2007*, pages 151–160, 2007.