A formalisation of argumentation schemes for legal case-based reasoning in ASPIC+

Henry Prakken[1], Adam Wyner[2],* Trevor Bench-Capon[3], and Katie Atkinson[3]

[1]University of Utrecht, Department of Information and Computing Sciences
and University of Groningen, Faculty of Law
[2]University of Aberdeen, Department of Computing Science
[3]University of Liverpool, Department of Computer Science

May 2, 2013

Abstract

In this paper we offer a formal account of reasoning with legal cases in terms of argumentation schemes. These schemes, and undercutting attacks associated with them, are formalised as defeasible rules of inference within the ASPIC+ framework. We begin by modelling the style of reasoning with cases developed by Aleven and Ashley in the CATO project, which describes cases using factors, and then extend the account to accommodate the dimensions used in Rissland and Ashley's earlier HYPO project. Some additional scope for argumentation is then identified and formalised.

Keywords: argumentation, legal case based reasoning

1 Introduction

Legal case-based reasoning (LCBR) has long been a topic of interest in AI and Law, and a variety of approaches have evolved. One important line of work on LCBR began with HYPO [2], developed by Edwina Rissland and her student, Kevin Ashley at Amherst. HYPO represented reasoning with legal cases as the exchange of arguments and counter arguments based on *dimensions*, legally significant aspects of the cases. Subsequently the ideas of HYPO were further developed by Ashley at Pittsburgh where he worked with his student, Vincent Aleven, on CATO [1], which introduced the notions of *factors* and a *factor hierarchy*, and with another student, Steffi Brüninghaus, on IBP [10] which attempted to predict case outcomes instead of simply identifying the arguments for the two sides. Like HYPO these systems were applied to US Trade

*Corresponding author. Email: adam@wyner.info. Postal address: Department of Computing Science, University of Aberdeen, Meston Building, Meston Walk, Aberdeen, AB24 3UE, United Kingdom.

Secrets law. Meanwhile Rissland stayed at Amherst where she worked with her student, David Skalak, on CABARET [26], which was based on Home Office Deduction cases and embedded the case based reasoning within a structure of rules modelling the relevant legislation, and with Skalak and Timur Friedman on BankXX [24], which generated arguments about Home Office Deduction through heuristic search. The model of case based reasoning used in this paper is largely based on the model developed in CATO, although we shall also draw on these other systems where convenient. More theoretically-oriented research related to this general approach appears in [22], [4], and [7]. In all these approaches, a current undecided case is decided by comparing and contrasting features in the current case against precedent cases in a case-base that have similar features. The decision in the "best" precedent case is then taken as the decision into the current case following the legal reasoning principle of *stare decisis*[1].

In [28], a number of novel argumentation schemes designed to reflect reasoning with factors as in Aleven and Ashley's CATO [1] were described, where the focus is to determine how and in what way a precedent case does (or does not) argue in support of a determination in the current case. However, the presentation in [28] was semi-formal and not set in an analytic framework which supports reasoning about these schemes. A simplified version of the schemes of [28] was modelled in Carneades [15] and included (along with many other schemes) in the semi-formal presentation of Argumentation Schemes for legal reasoning in [17]. In that paper the six schemes of [28] were reduced to three, and this resulted in distinctions that are arguably important being lost. Improved versions of the schemes of [28] were presented in [29]. In this paper, we reanalyse and formalise these legal case-based argumentation schemes in terms of the formal argumentation framework of ASPIC+ [21]. Formalising these schemes clarifies them and makes them more precise, while formalising them in ASPIC+ makes the metatheory of the ASPIC+ framework available for our account. In particular, we shall use its metatheory to prove that our specification satisfies the rationality postulates of [12]. A formalisation in ASPIC+ also illustrates the potential of that framework for formalising reasoning with argumentation schemes. This paper thus represents a substantially rewritten, revised and extended version which improves on both the formal representation and the analysis of [28], [17] and [29].

The current paper advances the state of the art in several respects: legal case-based reasoning with factors is clarified, defeasible legal case-based reasoning is represented and formalised in argumentation schemes, the arguments are compatible with and evaluated in a formally defined argumentation framework, and the analysis presents a well-developed and justified instantiation of defeasible argumentation schemes in a formal framework. Furthermore, the analysis provides a uniform representation language into which various alternative proposals for LCBR can be cast, compared, integrated, and reasoned with. Finally, the paper as a whole provides a demonstration of how an aspect of domain expertise, in this case reasoning with legal precedents, can be fruitfully captured and represented as a set of argumentation schemes, and the specific domain conceptualisation required to support them. This technique is generally applicable to expertise which comprises the ability to reason in a particular way.

[1] Literally "let the decision stand". This is the legal principle by which previous legal decisions are binding of courts of an equal or inferior status.

The organisation of the paper is as follows. We first set out elements of the formal framework for argumentation that we are assuming. In section 3, we discuss case-based reasoning as in CATO. We introduce our running example and the elements of the language we need for the argumentation schemes before presenting CATO style argumentation schemes in the formal framework. In section 4, issues concerning reasoning about dimensions, the relationship between facts and factors, and factor incompatibility are identified and discussed. Section 5 offers some additional discussion.

2 The Formal Setting

We first briefly summarise the formal frameworks used in this paper. An *abstract argument framework*, as introduced by Dung, [14] is a pair $AF = \langle \mathcal{A}, \mathit{defeat} \rangle$, where $\mathcal{A}$ is a set of arguments and *defeat* a binary relation on $\mathcal{A}$. A subset $\mathcal{B}$ of $\mathcal{A}$ is said to be *conflict-free* if no argument in $\mathcal{B}$ defeats an argument in $\mathcal{B}$ and it is said to be *admissible* if it is both conflict-free and also defends itself against any attack, i.e., if an argument A_1 is in $\mathcal{B}$ and some argument A_2 in $\mathcal{A}$ but not in $\mathcal{B}$ defeats A_1, then some argument in $\mathcal{B}$ defeats A_2. A *preferred extension* is then a maximal (with respect to set inclusion) admissible set. Dung defines several other types of extensions but they are not used in our model.

Dung's arguments are entirely abstract, with no features other than the defeat relation. A general framework for giving structure to arguments is the ASPIC framework, most fully defined as ASPIC+ in [21, 20]. The ASPIC+ framework first defines the notion of an *argumentation system*, which consists of a logical language $\mathcal{L}$ with a binary contrariness relation $^{-}$ and two sets of inference rules $\mathcal{R}_s$ and $\mathcal{R}_d$ of *strict* and *defeasible inference rules* defined over $\mathcal{L}$, written as $\varphi_1, \ldots, \varphi_n \rightarrow \varphi$ and $\varphi_1, \ldots, \varphi_n \Rightarrow \varphi$. Informally, that an inference rule is strict means that if its antecedents are accepted, then its consequent must be accepted *no matter what*, while that an inference rule is defeasible means that if its antecedents are accepted, then its consequent must be accepted *if there are no good reasons not to accept it*. An argumentation system also contains a function n which for each defeasible rule in $\mathcal{R}_d$ returns a formula in $\mathcal{L}$. Informally, $n(r)$ is a wff in $\mathcal{L}$ which says that the defeasible rule $r \in \mathcal{R}$ is applicable.

In the present paper we use an argumentation system in which $\mathcal{L}$ is a first-order language with equality further specified in the coming sections, its contrariness relation corresponds to classical negation, the strict rules $\mathcal{R}_s$ are all valid first-order inferences over $\mathcal{L}$ and the defeasible rules $\mathcal{R}_d$ are as specified in the coming sections.

Arguments are in ASPIC+ constructed from a knowledge base $\mathcal{K}$, which contains two disjoint kinds of formulae: the *axioms* $\mathcal{K}_n$ and the *ordinary premises* $\mathcal{K}_p$. The formal definition of an argument is as follows (here, for a given argument, the function `Prem` returns all the formulae of $\mathcal{K}$ (called *premises*) used to build the argument, `Conc` returns its conclusion, `Sub` returns all its sub-arguments, and `TopRule` returns the last inference rule used in the argument):

Definition 2.1 [Argument] An *argument* A on the basis of a knowledge base $\mathcal{K}$ in an argumentation system $(\mathcal{L}, ^{-}, \mathcal{R}_s, \mathcal{R}_d, n)$ is:

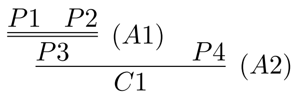

Figure 1: An argument

1. φ if $\varphi \in \mathcal{K}$ with: $\texttt{Prem}(A) = \{\varphi\}$; $\texttt{Conc}(A) = \varphi$; $\texttt{Sub}(A) = \{\varphi\}$; $\texttt{TopRule}(A)$ = undefined.
2. $A_1, \dots A_n \rightarrow/\Rightarrow \psi$ if $A_1, \dots, A_n$ are arguments such that there exists a strict or a defeasible rule $\texttt{Conc}(A_1), \dots, \texttt{Conc}(A_n) \rightarrow/\Rightarrow \psi$ in $\mathcal{R}_s/\mathcal{R}_d$.
   $\texttt{Prem}(A) = \texttt{Prem}(A_1) \cup \dots \cup \texttt{Prem}(A_n)$; $\texttt{Conc}(A) = \psi$; $\texttt{Sub}(A) = \texttt{Sub}(A_1) \cup \dots \cup \texttt{Sub}(A_n) \cup \{A\}$; $\texttt{TopRule}(A)$ = $\texttt{Conc}(A_1), \dots, \texttt{Conc}(A_n) \rightarrow/\Rightarrow \psi$.

An argument is *strict* if all its inference rules are strict and *defeasible* otherwise, and it is *firm* if all its premises are in $\mathcal{K}_n$ and *plausible* otherwise.

Arguments can be displayed as inference trees. An example argument, A_2, is shown in Figure 1. A_2 has premises P_1, P_2, P_4, and conclusion C_1. A single and double bar stand for, respectively, a strict and defeasible inference. Argument A_2 has four subarguments, namely A_1, which has premises P_1 and P_2 and conclusion P_3, and the formulae P_1, P_2 and P_4 as atomic subarguments.

An argumentation system and a knowledge base are combined with an *argument ordering* into an *argumentation theory*. The argument ordering could be defined in any way, for example, in terms of orderings on $\mathcal{R}_d$ and $\mathcal{K}_p$.

Definition 2.2 [Argumentation theories] An *argumentation theory* is a triple $AT = (AS, \mathcal{K}, \preceq)$ where AS is an argumentation system, $\mathcal{K}$ is a knowledge base in AS and $\preceq$ is an ordering on the set of all arguments that can be constructed on the basis of $\mathcal{K}$ in AS.

Arguments can be attacked in three ways: attacking a conclusion of a defeasible inference, attacking the defeasible inference itself, or attacking a premise. To define how a defeasible inference can be attacked, the function n of an AS can be used, which assigns to each element of $\mathcal{R}_d$ a well-formed formula in $\mathcal{L}$. Recall that informally, $n(r)$ (where $r \in R_d$) means that r is applicable. For our argumentation system, ASPIC+'s definitions of attack can be simplified as follows:[2]

Definition 2.3 [attacks] A *attacks* B iff A *undercuts*, *rebuts* or *undermines* B, where:

- A *undercuts* argument B (on B') iff $\texttt{Conc}(A) = -n(r)$ for some $B' \in \texttt{Sub}(B)$ such that B''s top rule r is defeasible.
- A *rebuts* argument B (on B') iff $\texttt{Conc}(A) = -\varphi$ for some $B' \in \texttt{Sub}(B)$ of the form $B''_1, \dots, B''_n \Rightarrow \varphi$.
- Argument A *undermines* B (on φ) iff $\texttt{Conc}(A) = -\varphi$ for some ordinary premise φ of B.

[2] In the definitions below, $-\neg\varphi$ denotes φ, while if φ does not start with a negation, $-\varphi$ denotes $\neg\varphi$.

In Figure 1, argument A_2 can only be rebutted or undercut on its defeasible subargument A_1.

Undercuts and attacks that are combined with the preferences defined by an argument ordering yield three ways in which an argument may be defeated: undercut (which is independent of preferences), successful undermining and successful rebuttal (both of which do depend on preferences).

Definition 2.4 [Successful rebuttal, successful undermining and defeat]

- A *successfully rebuts* B if A rebuts B on B' and $A \not\prec B'$.
- A *successfully undermines* B if A undermines B on φ and $A \not\prec \varphi$.
- A *defeats* B iff A undercuts or successfully rebuts or successfully undermines B.

The success of rebutting and undermining attacks thus involves comparing the conflicting arguments at the points where they conflict. The definition of successful undermining exploits the fact that an argument premise is also a subargument. For undercutting attack no preferences are needed to make it succeed, since undercutters state exceptions to the rule they attack.

ASPIC+ thus defines a set of arguments with a binary relation of defeat, that is, it defines abstract argumentation frameworks in the sense of [14]. Formally:

Definition 2.5 [Argumentation framework] An *abstract argumentation framework* (AF) *corresponding to an argumentation theory* AT is a pair $< \mathcal{A}$, *Def*$>$ such that:

- $\mathcal{A}$ is the set of arguments on the basis of AT as defined by Definition 2.1,
- *Def* is the relation on $\mathcal{A}$ given by Definition 2.4.

Thus any semantics for abstract argumentation can be applied to ASPIC+.

3 CATO Argumentation Schemes

In this section, formal argumentation schemes for CATO style case-based reasoning are provided. We give a brief overview of CBR as represented in CATO in section 3.1, introduce our running example in section 3.2, present elements of the language in section 3.3, formalise the argumentation schemes in 3.4, and report the results with respect to our example in 3.5. In section 3.6 we prove that our ASPIC+ argumentation theories satisfy [12]'s rationality postulates of strict closure and consistency.

3.1 Case-based Reasoning as in CATO

CATO [1], which we focus on in this section, analyses cases in terms of *factors*, where a factor is a prototypical fact situation that predisposes the decision in favour of one party or the other in the case; for trade secret law, the domain CATO is designed for, the factors concern trade secret misappropriation and are derived from *Restatement of Torts*

First, Sec. 757 and the *Uniform Trade Secret Act* (see [2, 1]). As different precedents have different distributions of factors, finding and reasoning about precedents with respect to a current case requires one to examine the combinations of and counter-balancing between, factors in the cases. In addition to the factors themselves, there is a *factor hierarchy* in which an *abstract factor* has factors as children; in reasoning with the abstract factors and the factors of a case, differences between the cases can sometimes be reconciled. The argumentation schemes discussed in this paper make such reasoning patterns explicit and formal.

A case comparison method for LCBR was introduced in [4], where cases are analysed in terms of *partitions* of case factors. Various distributions of factors amongst the partitions can be used to support or undermine the plaintiff's argument that the current case should be decided in the plaintiff's favour. [28] provided some informally expressed *argumentation schemes* for this partition method, where the schemes are *defeasible reasoning patterns* and the partitions are sets of CATO factors and the factor hierarchy is used. This paper formalises, articulates, and extends this line of research on LCBR.

3.2 Running Example

To clarify the discussion, we provide a running example using *Mason v Jack Daniels Distillery* (indicated with *Mason*) and *M. Bryce and Associates v Gladstone* (indicated with *Bryce*) as analysed in CATO, based on the factors and factor hierarchy in [1].

Mason v Jack Daniels Distillery[3], is a well known case, so well known that an episode of the Simpsons[4] was based on it. A bartender, Tony Mason, invented a cocktail, Lynchburgh Lemonade comprising Jack Daniel's whiskey, Triple Sec, sweet and sour mix, and 7-Up. It proved surprisingly popular. Mason met Winston Randle, a sales representative for Jack Daniel Distillery, and they talked about the drink, and its possible use in a promotion. Approximately one year later the defendants were developing a national promotion campaign for Lynchburg Lemonade. Mason claimed that he had parted with the recipe because he had been told that his band would be used in the promotion. In fact Mason received nothing. The jury found for the plaintiff, but awarded only a dollar in damages. Here we will treat *Mason* as the current case under consideration.

In *M. Bryce and Associates v Gladstone*[5] Bryce was a software company with a product bearing the registered trademark "Pride". "Pride" is a complete methodology for the design, development and implementation of an information system. Bryce made a presentation of "Pride" to the defendants, hoping to make a sale, after which the defendants designed and implemented a manual that duplicated its procedures, forms and standards. *Bryce* thus also involves disclosure in negotiations and was found for the plaintiff, and so can serve as a possible precedent.

We give the factors for each case, as used in CATO, (the factor identifiers, *F1* and so on are those used in [1] and adopted in other work discussing CATO). We also indicate the side favoured by the factors:

[3] 518 So.2d 130, 1987 Ala. Civ. App.

[4] Flaming Moe's, the tenth episode of the third season.

[5] 107 Wis. 2d 241

- *Mason*
  - *F1 Disclosure-In-Negotiations (d)*
  - *F6 Security-Measures (p)*
  - *F15 Unique-Product (p)*
  - *F16 Info-Reverse-Engineerable (d)*
  - *F21 Knew-Info-Confidential (p)*
- *Bryce*
  - *F1 Disclosure-In-Negotiations (d)*
  - *F4 Agreed-Not-To-Disclose (p)*
  - *F6 Security-Measures (p)*
  - *F18 Identical-Products (p)*
  - *F21 Knew-Info-Confidential (p)*

In subsequent sections, we illustrate the formalism with this example.

3.3 Elements of a Language

We begin by defining the language that we shall use to talk about our cases and which will be used in our underlying knowledge base. We assume a many-sorted first-order language with sorts for parties, cases (with subsorts for current cases and precedents), factors and factor sets. We trust that the types of the terms and predicate and function symbols will be clear from the context and wording.

We first discuss some preliminaries. To correctly represent and reason with set-theoretic expressions, the following definitions are assumed to be in $\mathcal{K}_n$:

1. $\forall s, s'.(s \subseteq s' \equiv \forall x(x \in s \implies x \in s'))$[6]
2. $\forall x, s, s'.(x \in s \cap s' \equiv (x \in s \wedge x \in s'))$
3. $\forall x, s, s'.(x \in s \cup s' \equiv (x \in s \vee x \in s'))$
4. $\forall x, s, s'.(x \in s \setminus s' \equiv (x \in s \wedge x \notin s'))$

In expressions like $\mathit{pFactors}(\mathit{Mason}) = \{\mathit{F6}, \mathit{F15}, \mathit{F21}\}$ the brackets $\{$ and $\}$ are together a function symbol operating on the terms $\mathit{F6}$, $\mathit{F15}$ and $\mathit{F21}$. To preserve the meaning of the function symbol the following axiom is added to $\mathcal{K}_n$:

5. $\forall s, x_1, \ldots, x_n.s = \{x_1, \ldots, x_n\} \equiv \forall y(y \in s \equiv (y = x_1 \vee \ldots \vee y = x_n))$

Here the variable s ranges over sets. This definition assumes that sets are finite, which in our domain is a safe assumption.

Factors are in $\mathcal{K}_p$ declared to be either pro-plaintiff or pro-defendant, with:

[6] In this paper the long double arrow $\Longrightarrow$ denotes the material implication.

- $\mathit{pFactor}(\mathit{factor})$, meaning that factor is a pro-plaintiff factor;

- $\mathit{dFactor}(\mathit{factor})$, meaning that factor is a pro-defendant factor.

In our running example we have at least the following formulae in $\mathcal{K}_p$:

- $\mathit{pFactor}(\mathit{F4})$

- $\mathit{pFactor}(\mathit{F6})$

- $\mathit{pFactor}(\mathit{F15})$

- $\mathit{pFactor}(\mathit{F18})$

- $\mathit{pFactor}(\mathit{F21})$

- $\mathit{dFactor}(\mathit{F1})$

- $\mathit{dFactor}(\mathit{F16})$

No factor can be both pro-plaintiff and pro-defendant, expressed by adding to $\mathcal{K}_n$:

6. $\forall \mathit{factor} \neg(\mathit{pFactor}(\mathit{factor}) \wedge \mathit{dFactor}(\mathit{factor}))$

We next turn to the representation of cases. We will not talk about cases directly, but instead refer to cases as analysed for use by the CATO system, which is the system that provides the paradigm on which our argumentation will be based. For CATO, a case has a name, a set of factors in favour of the plaintiff, a set of factors in favour of the defendant, and (if the case is a precedent) an outcome, which is one of *plaintiff* or *defendant*. We describe cases as follows. First for each case the sets of (plaintiff and defendant) factors in the case are specified with the following predicates:

- $\mathit{hasFactor}(\mathit{case}, \mathit{factor})$, meaning that factor is a factor in case.

- $\mathit{hasPfactor}(\mathit{case}, \mathit{factor})$, meaning that factor is a plaintiff factor in case.

- $\mathit{hasDfactor}(\mathit{case}, \mathit{factor})$, meaning that factor is a defendant factor in case.

Moreover, the following definitions are added to $\mathcal{K}_n$.

7. $\forall \mathit{case}, \mathit{factor}. \mathit{hasPfactor}(\mathit{case}, \mathit{factor}) \equiv \mathit{hasFactor}(\mathit{case}, \mathit{factor}) \wedge \mathit{pFactor}(\mathit{factor})$

8. $\forall \mathit{case}, \mathit{factor}. \mathit{hasDfactor}(\mathit{case}, \mathit{factor}) \equiv \mathit{hasFactor}(\mathit{case}, \mathit{factor}) \wedge \mathit{dFactor}(\mathit{factor})$

The idea is that $\mathit{hasFactor}(\mathit{case}, \mathit{factor})$ statements are added to $\mathcal{K}_p$ and that they together with the specifications of the types of factors in $\mathcal{K}_p$ give rise to strict arguments for $\mathit{hasPfactor}(\mathit{case}, \mathit{factor})$ and $\mathit{hasDfactor}(\mathit{case}, \mathit{factor})$ conclusions. In our running example we thus have strict arguments for the following conclusions:

- $\mathit{hasPfactor}(\mathit{Mason}, \mathit{F6})$

- $\mathit{hasPfactor}(\mathit{Mason}, \mathit{F15})$
- $\mathit{hasPfactor}(\mathit{Mason}, \mathit{F21})$
- $\mathit{hasDfactor}(\mathit{Mason}, \mathit{F1})$
- $\mathit{hasDfactor}(\mathit{Mason}, \mathit{F16})$
- $\mathit{hasPfactor}(\mathit{Bryce}, \mathit{F4})$
- $\mathit{hasPfactor}(\mathit{Bryce}, \mathit{F6})$
- $\mathit{hasPfactor}(\mathit{Bryce}, \mathit{F18})$
- $\mathit{hasPfactor}(\mathit{Bryce}, \mathit{F21})$
- $\mathit{hasDfactor}(\mathit{Bryce}, \mathit{F1})$

To ensure that a factor belongs to a case if and only if specified as such, the predicate completions of the predicate $\mathit{hasFactor}$ and the unique-names and domain-closure axioms for objects satisfying these predicates are added to $\mathcal{K}_n$. This makes, for example, the following pairs of formulae mutually inconsistent:

- $\mathit{pFactors}(\mathit{Mason}) = \{\mathit{F6}, \mathit{F15}, \mathit{F21}\}$ and $\mathit{pFactors}(\mathit{Mason}) = \{\mathit{F4}, \mathit{F21}\}$
- $\mathit{pFactors}(\mathit{Mason}) = \{\mathit{F6}, \mathit{F15}, \mathit{F21}\}$ and $\mathit{hasPfactors}(\mathit{Mason}, \mathit{F4})$

In our running example the predicate completion formulae are as follows:

- $\forall \mathit{case}, \mathit{factor}. \mathit{hasFactor}(\mathit{case}, \mathit{factor}) \equiv$
  $((\mathit{case} = \mathit{Mason} \wedge (\mathit{factor} = \mathit{F1} \vee \mathit{factor} = \mathit{F6} \vee \mathit{factor} = \mathit{F15} \vee \mathit{factor} = \mathit{F1} \vee \mathit{factor} = \mathit{F21})) \vee$
  $(\mathit{case} = \mathit{Bryce} \wedge (\mathit{factor} = \mathit{F1} \vee \mathit{factor} = \mathit{F4} \vee \mathit{factor} = \mathit{F6} \vee \mathit{factor} = \mathit{F18} \vee \mathit{factor} = \mathit{F21})))$

The unique-names and domain closure axioms are:

- $\forall \mathit{case}. \mathit{case} = \mathit{Mason} \vee \mathit{case} = \mathit{Bryce}$
- $\mathit{Mason} \neq \mathit{Bryce}$
- $\forall \mathit{factor}. \mathit{factor} = \mathit{F1} \vee \ldots \vee \mathit{factor} = \mathit{F115}$
- $\mathit{F1} \neq \ldots \neq \mathit{F115}$

The following three function expressions are used to denote a case's sets of pro-plaintiff and pro-defendant factors and its outcome:

- $\mathit{pFactors}(\mathit{case}) = \mathit{setOfFactors}.$
- $\mathit{dFactors}(\mathit{case}) = \mathit{setOfFactors}.$
- $\mathit{outcome}(\mathit{case}) = \mathit{party}.$

We then add the following axioms to $\mathcal{K}_n$ to link expressions with the $\mathit{hasPfactor}$, $\mathit{hasDfactor}$ and $\mathit{hasFactor}$ predicates to expressions with the $\in$ symbol. It is these axioms that enable set-theoretic operations on factors and factor sets.

9. $\forall case, f.f \in Factors(case) \equiv \mathit{hasFactor}(\mathit{case}, \mathit{factor})$

10. $\forall case, f.f \in pFactors(case) \equiv \mathit{hasFactor}(\mathit{case}, \mathit{factor}) \wedge \mathit{pFactor}(\mathit{factor})$

11. $\forall case, f.f \in dFactors(case) \equiv \mathit{hasFactor}(\mathit{case}, \mathit{factor}) \wedge \mathit{dFactor}(\mathit{factor})$

With respect to our running example, we then have strict arguments for the following conclusions:

- $\mathit{pFactors}(\mathit{Mason}) = \{\mathit{F6}, \mathit{F15}, \mathit{F21}\}$
- $\mathit{dFactors}(\mathit{Mason}) = \{\mathit{F1}, \mathit{F16}\}$
- $\mathit{pFactors}(\mathit{Bryce}) = \{\mathit{F4}, \mathit{F6}, \mathit{F18}, \mathit{F21}\}$
- $\mathit{dFactors}(\mathit{Bryce}) = \{\mathit{F1}\}$

We also have in $\mathcal{K}_p$:

- $\mathit{outcome}(\mathit{Bryce}) = \mathit{Plaintiff}$

Additionally a feature of CATO is that factors are organised into a factor hierarchy, with factors being the children of more abstract factors. Thus for every factor we can have relations of the form:

- $\mathit{parentFactor}(\mathit{factor}, \mathit{abstractFactor})$

While CATO has some intermediate layers in the factor hierarchy, we omit some of them for our current purposes as well as the label of these higher level factors. The abstract factors are also associated with a side, with the following formulae in $\mathcal{K}_p$:

- $\mathit{pFactor}(\mathit{F102})$
- $\mathit{pFactor}(\mathit{F115})$
- $\mathit{dFactor}(\mathit{F105})$

Note that cases are described only in terms of base level factors: thus $\mathit{pFactors}(\mathit{case})$ and $\mathit{dFactors}(\mathit{case})$ do not return any abstract factors. Similarly $\mathit{hasFactor}$ does not have any associated abstract factors. The factor hierarchy was originally built by Aleven starting from the base level factors, and in principle it would be possible to construct different factor hierarchies, using different abstract factors and/or different $parentFactor$ relations, in which case paternity could even be the subject of dispute, and $\mathit{parentFactor}(\mathit{factor}, \mathit{abstractFactor})$ would need to be the conclusion of some rule, rather than a premise. We will, however, consider the factor hierarchy to be fixed to that used in [1], and use $parentFactor$ only as it is defined there.

- $\mathit{parentFactor}(\mathit{F1}, \mathit{F102})$

- $\mathit{parentFactor}(\mathit{F4}, \mathit{F102})$
- $\mathit{parentFactor}(\mathit{F4}, \mathit{F115})$
- $\mathit{parentFactor}(\mathit{F6}, \mathit{F102})$
- $\mathit{parentFactor}(\mathit{F15}, \mathit{F105})$
- $\mathit{parentFactor}(\mathit{F16}, \mathit{F105})$
- $\mathit{parentFactor}(\mathit{F21}, \mathit{F115})$

A factor hierarchy can be specified by adding a formula of the following form to $\mathcal{K}_p$:

- $\forall \mathit{factor1}, \mathit{factor2}. \mathit{parentFactor}(\mathit{factor1}, \mathit{factor2}) \equiv (\mathit{factor1} = F_{i1} \wedge \mathit{factor2} = F_{j1}) \vee \ldots \vee (\mathit{factor1} = F_{in} \wedge \mathit{factor2} = F_{jn})$

If desired, axioms can be added to $\mathcal{K}_n$ to exclude cycles in the factor hierarchy, but multiple parents must be allowed to represent the hierarchy of [1].

Cases are compared with one another in terms of their factors. This gives rise to a further six functions of the following type:

$$\mathit{commonPfactors} : \mathit{currentcases} \times \mathit{precedents} \longrightarrow 2^{\mathit{factors}}$$

The six functions are defined as follows as elements of $\mathcal{K}_n$.

$\forall f, \mathit{curr}, \mathit{prec}$:

12. $\mathit{commonPfactors}(\mathit{curr}, \mathit{prec}) = \mathit{pFactors}(\mathit{curr}) \cap \mathit{pFactors}(\mathit{prec})$
13. $\mathit{commonDfactors}(\mathit{curr}, \mathit{prec}) = \mathit{dFactors}(\mathit{curr}) \cap \mathit{dFactors}(\mathit{prec})$
14. $\mathit{currPfactors}(\mathit{curr}, \mathit{prec}) = \mathit{pFactors}(\mathit{curr}) \setminus \mathit{commonPfactors}(\mathit{curr}, \mathit{prec})$
15. $\mathit{currDfactors}(\mathit{curr}, \mathit{prec}) = \mathit{dFactors}(\mathit{curr}) \setminus \mathit{commonDfactors}(\mathit{curr}, \mathit{prec})$
16. $\mathit{precPfactors}(\mathit{curr}, \mathit{prec}) = \mathit{pFactors}(\mathit{prec}) \setminus \mathit{commonPfactors}(\mathit{curr}, \mathit{prec})$
17. $\mathit{precDfactors}(\mathit{curr}, \mathit{prec}) = \mathit{dFactors}(\mathit{prec}) \setminus \mathit{commonDfactors}(\mathit{curr}, \mathit{prec})$

With respect to our running example, we have:

- $\mathit{commonPfactors}(\mathit{Mason}, \mathit{Bryce}) = \{\mathit{F6}, \mathit{F21}\}$
- $\mathit{commonDfactors}(\mathit{Mason}, \mathit{Bryce}) = \{\mathit{F1}\}$
- $\mathit{currPfactors}(\mathit{Mason}, \mathit{Bryce}) = \{\mathit{F15}\}$
- $\mathit{currDfactors}(\mathit{Mason}, \mathit{Bryce}) = \{\mathit{F16}\}$
- $\mathit{precPfactors}(\mathit{Mason}, \mathit{Bryce}) = \{\mathit{F4}, \mathit{F18}\}$
- $\mathit{precDfactors}(\mathit{Mason}, \mathit{Bryce}) = \emptyset$

These relations are the building blocks for our arguments. The first two are the basis for a comparison and represent what is common between the two cases. The remaining four represent differences, and their effect will depend on the outcome of the previous case and the side for which we are arguing. Suppose we are arguing for the plaintiff: then we can only use precedents with the outcome plaintiff. For such cases, *currPfactors* and *precDfactors* will strengthen the plaintiff's position, since they represent, respectively, plaintiff reasons in *curr* not available in the *prec* and defendant reasons in the *prec* which are not available in *curr*. On the other hand, *currDfactors* and *precPfactors* weaken the plaintiff's position in *curr*. Similarly, if arguing for the defendant in the *curr*, *currDfactors* and *precPfactors* strengthen the position and *currPfactors* and *precDfactors* weaken it. The precise nature of the strengthening and weakening will be made clear when we consider the argumentation schemes based on these different partitions.

Next we need to express that one set of factors, *factorSet1*, is preferred over another, *factorSet2*.

- *preferred(factorSet1,factorSet2)*.

In our analysis, the preference is the claim of a defeasible argumentation scheme *CS2*, which only appears later. We cannot, then, straightforwardly provide the preference in our running example until section 3.4.

Finally, we need another relation between factors. If factors for a given party share the same ancestor, then both factors get their force from the fact that the same abstract factor is present in the case. This means that, if they favour the same party to the case, it may be possible to substitute one for another. Similarly if they favour different parties, they may cancel each other out so as to remove the abstract factor from the case. Therefore we have two additional predicates:

- $substitutes(factor1, factor2)$
- $cancels(factor1, factor2)$

To define these predicates, the following definition of the ancestor relation between factors is added to $\mathcal{K}_n$:

18. $\forall f_1, f_2.ancestor(f_1, f_2) \equiv parentFactor(f_1, f_2) \vee \exists f_3(ancestor(f_1, f_3) \wedge parentFactor(f_3, f_2))$

We define substitution and cancellation of factors that would benefit the plaintiff as follows, where substitutions apply *between* cases and cancellations apply *within* cases. Substitutions and cancellations for the defendant would be similar, though switching the predicates (and factor sets). The following definitions are in $\mathcal{K}_n$:

19. $\forall f_1, f_2.substitutes(f_1, f_2) \equiv$
$((pFactor(f_1) \wedge pFactor(f_2)) \vee (dFactor(f_1) \wedge dFactor(f_2))) \wedge$
$\exists f_3(ancestor(f_1, f_3) \wedge ancestor(f_2, f_3))$

20. $\forall f_1, f_2.cancels(f_1, f_2) \equiv$
$((pFactor(f_1) \wedge dFactor(f_2)) \vee (dFactor(f_1) \wedge pFactor(f_2))) \wedge$
$\exists f_3(ancestor(f_1, f_3) \wedge ancestor(f_2, f_3))$

In our running example, we have:

- $\mathit{substitutes}(\mathit{F4}, \mathit{F6})$ since $\mathit{precPfactors}(\mathit{Mason}, \mathit{Bryce}) = \{\mathit{F4}, \mathit{F18}\}$ and $\mathit{F4} \in \{\mathit{F4}, \mathit{F18}\}$ and $\mathit{pFactors}(\mathit{Mason}) = \{\mathit{F6}, \mathit{F15}, \mathit{F21}\}$ and $\mathit{F6} \in \{\mathit{F6}, \mathit{F15}, \mathit{F21}\}$ and $\mathit{parentFactor}(\mathit{F4}, \mathit{F102})$ and $\mathit{parentFactor}(\mathit{F6}, \mathit{F102})$.
- $\mathit{cancels}(\mathit{F15}, \mathit{F16})$ since $\mathit{currPfactors}(\mathit{Mason}, \mathit{Bryce}) = \{\mathit{F15}\}$ and $\mathit{F15} \in \{\mathit{F15}\}$ and $\mathit{dFactors}(\mathit{Mason}) = \{\mathit{F1}, \mathit{F16}\}$ and $\mathit{F16} \in \{\mathit{FF1}, \mathit{F16}\}$ and $\mathit{parentFactor}(\mathit{F15}, \mathit{F105})$ and $\mathit{parentFactor}(\mathit{F16}, \mathit{F105})$.

Intuitively, we want to argue that we should decide *Mason* for the plaintiff on the basis of *Bryce*. The argument will be that *Mason* and *Bryce* share several factors (both for plaintiff and defendant), and so, since *Bryce* was decided for the plaintiff, so too should *Mason* be decided, provided that any differences between them can be argued away by substitution and cancellation.

In the next section we will present the argumentation schemes built from this language.

3.4 CATO style Argumentation Schemes

In this section we specify the defeasible inference rules $\mathcal{R}_d$ of our ASPIC+ argumentation system. For readability we will not specify them with the rule symbol $\Rightarrow$ but as *argumentation schemes*, i.e., with a double horizontal inference bar. Rule schemes will be named by expressions $\mathit{Name}(x_1, \ldots, x_n)$ where the predicate Name stands for the informal name of the rule and $x_1, \ldots, x_n$ are all free variables occurring in the scheme. These variables are replaced by ground terms for each instance of the scheme, resulting in closed formulae that are the names of the scheme instances according to the function n mentioned just before Definition 2.3.

In this section we will always suppose that we wish to argue the *curr* for the plaintiff. Arguments for the defendant are similar, except that the strengthening and weakening factor partitions are reversed as discussed above. The argument is that the *curr* should be decided for the plaintiff because the common p factors were preferred to the common d factors in the *prec*.[7]

CS1$(\mathit{curr}, \mathit{prec}, p, d)$:

$$\begin{array}{c} \mathit{commonPfactors}(\mathit{curr}, \mathit{prec}) = p, \\ \mathit{commonDfactors}(\mathit{curr}, \mathit{prec}) = d, \\ \mathit{preferred}(p, d) \\ \hline\hline \mathit{outcome}(\mathit{curr}) = \mathit{Plaintiff} \end{array}$$

Instantiating CS1, where our *curr* is *Mason* and our *prec* is *Bryce*, we have the following argument, indicated with **Mason(Bryce)A1**:

Mason(Bryce)A1

[7] Note that CS1 uses only a subset of the factors from the precedent: this is because CS2 also encapsulates the rule broadening move as discussed in [26], which is necessary to adapt the *prec* so as to match the *curr*.

$$\begin{array}{c} \mathit{commonPfactors}(\mathit{Mason}, \mathit{Bryce}) = \{\mathit{F6}, \mathit{F21}\}, \\ \mathit{commonDfactors}(\mathit{Mason}, \mathit{Bryce}) = \{\mathit{F1}\}, \\ \mathit{preferred}(\{\mathit{F6}, \mathit{F21}\}, \{\mathit{F1}\}) \\ \hline\hline \mathit{outcome}(\mathit{Mason}) = \mathit{Plaintiff} \end{array}$$

Note that strictly speaking some of these premises are derived from $\mathcal{K}_n \cup \mathcal{K}_p$. However, to keep the arguments reasonably readable we will leave strict derivations from the knowledge base implicit.

We will assume at this point that the information about cases in our KB is correct, or at least beyond dispute; this is relaxed in section 4.2. In ASPIC+ terms this makes the case facts *axioms* and so the first two premises cannot be questioned. The third, however, needs to be established, and this will be done using CS2, which we will describe after considering undercutters to CS1.

There may, of course, be rebuttals, using a variety of argumentation schemes, but we need to recognise that even if such a preference has been established in the *prec*, it may not be applicable to the *curr*, because the defendant has arguments in the *curr* that were not available in the *prec*. We therefore have the undercutting attack for arguments using CS1.

$\textbf{U1.1}(\mathit{cur}, \mathit{prec}, p, d)$:

$$\begin{array}{c} f \in \mathit{currDfactors}(\mathit{curr}, \mathit{prec}) \\ \hline\hline \neg\textbf{CS1}(\mathit{cur}, \mathit{prec}, p, d) \end{array}$$

Instantiating U1.1 with *Mason*, *Bryce* and the relevant sets, we have an undercutter argument:

Mason(Bryce)A2:

$$\begin{array}{c} \mathit{F16} \in \mathit{currDfactors}(\mathit{Mason}, \mathit{Bryce}) \\ \hline\hline \neg\textbf{CS1}(\mathit{Mason}, \mathit{Bryce}, \{\mathit{F6}, \mathit{F21}\}, \{\mathit{F1}\}) \end{array}$$

While this presents a challenge to the plaintiff, the argument for the plaintiff can be defended if the distinctions between the cases can be *downplayed*. The undercutting move of U1.1 is one way of distinguishing the two cases, and in CATO the abstract factor hierarchy allows us to downplay distinctions. This downplaying can be done in two ways, *substitution* or *cancellation*, corresponding to the two different kinds of extra strength the *curr* may have. Accordingly we introduce two schemes that can be used to provide undercutters of U1.1:

$\textbf{U1.1.1}(\mathit{curr}, \mathit{prec}, f_1, f_2, p, d)$:

$$\begin{array}{c} f_1 \in \mathit{currDfactors}(\mathit{curr}, \mathit{prec}), \\ f_2 \in \mathit{dFactors}(\mathit{prec}), \\ \mathit{substitutes}(f_1, f_2) \\ \hline\hline \neg\textbf{U1.1}(\mathit{cur}, \mathit{prec}, p, d) \end{array}$$

$\textbf{U1.1.2}(\mathit{curr}, \mathit{prec}, f_1, f_2, p, d)$:

$$\begin{array}{c} f_1 \in \mathit{currDfactors}(\mathit{curr}, \mathit{prec}), \\ f_2 \in \mathit{pFactors}(\mathit{curr}), \\ \mathit{cancels}(f_1, f_2) \\ \hline\hline \neg\mathbf{U1.1}(\mathit{cur}, \mathit{prec}, p, d) \end{array}$$

The idea here is that as the undercutting factor in the *curr* has the same parent as a factor in *prec*, we can substitute for the undercutting factor, where the point is that the abstract factor can be seen to have been applied also in the *prec*; alternatively, the undercutting factor in the *curr* is cancelled out by some other factor in *curr*, so that the abstract factor does not apply. Instantiating U1.1.2 with our running example and given that we previously determined that *cancels(F15, F16)*, we can form the following argument:

Mason(Bryce)A3:

$$\begin{array}{c} \mathit{F16} \in \mathit{currDfactors}(\mathit{Mason}, \mathit{Bryce}), \\ \mathit{F15} \in \mathit{pFactors}(\mathit{Mason}), \\ \mathit{cancels}(\mathit{F15}, \mathit{F16}) \\ \hline\hline \neg\mathbf{U1.1}(\mathit{cur}, \mathit{prec}, p, d) \end{array}$$

We now turn to the argumentation scheme to establish the preference between two sets of factors, required to justify the third premise of CS1.

CS2$(\mathit{cur}, \mathit{prec}, p, d)$:

$$\begin{array}{c} \mathit{commonPfactors}(\mathit{curr}, \mathit{prec}) = p, \\ \mathit{commonDfactors}(\mathit{curr}, \mathit{prec}) = d, \\ \mathit{outcome}(\mathit{prec}) = \mathit{Plaintiff} \\ \hline\hline \mathit{preferred}(p, d) \end{array}$$

Note that CS2 establishes a preference between two particular sets. It might seem natural to add that from $\mathit{preferred}(p, d)$ we should be able to derive $\mathit{preferred}(p', d)$ where $p' \supset p$ and $\mathit{preferred}(p, d')$ where $d' \subset d$, as in, for example [22]. This, however, would be to go beyond CATO. Moreover it would arguably go against the spirit of CATO-style reasoning, which insists that all claims about preferences are based on a *specific* precedent. CATO always argues using a particular precedent, never with a set of precedents. If the current case does in fact contain additional pro-plaintiff factors or fewer pro-defendant factors, these are made use of in different arguments employing the argumentation schemes CS3 and CS4 discussed below.

Instantiating CS2, we have an argument for the preference, as mentioned above:

Mason(Bryce)A4:

$$\begin{array}{c} \mathit{commonPfactors}(\mathit{Mason}, \mathit{Bryce}) = \{\mathit{F6}, \mathit{F21}\}) \\ \mathit{commonDfactors}(\mathit{Mason}, \mathit{Bryce}) = \{\mathit{F16}\}) \\ \mathit{outcome}(\mathit{Bryce}) = \mathit{Plaintiff} \\ \hline\hline \mathit{preferred}(\{\mathit{F6}, \mathit{F21}\}, \{\mathit{F16}\}) \end{array}$$

All of the premises of CS2 are taken from our database, or straightforward set operations on such data and so represent ASPIC+ axioms which cannot be questioned. It is, however, possible to both rebut and to undercut the argument.

R2.1$(cur, prec, prec2, p, d)$:

$$\frac{\begin{array}{c} p \subseteq \mathit{commonPfactors}(curr, \mathit{prec2}), \\ d \subseteq \mathit{commonDfactors}(curr, \mathit{prec2}), \\ \mathit{outcome}(prec) = \mathit{Defendant} \end{array}}{\neg \mathit{preferred}(p, d)}$$

Attacks made using R2.1 offer counter examples in which the same comparison was available in a case decided for the defendant, suggesting that the preference is opposite, and so providing a rebuttal. We do not consider such rebuttals further in this paper, but in arguing a case they would be subject to attacks using the schemes introduced in this paper, just like CS2. The following scheme can be used to undercut CS2.

U2.1$(curr, prec, p, d)$:

$$\frac{f \in \mathit{precPfactors}(curr, prec)}{\neg \mathbf{CS2}(curr, prec, p, d)}$$

Instantiating U2.1 with *Mason* and *Bryce*, we have two arguments, one for each factor in precPfactors:

Mason(Bryce)A5:

$$\frac{\mathit{F4} \in \mathit{precPfactors}(Mason, Bryce)}{\neg \mathbf{CS2}(Mason, Bryce, \{\mathit{F6}, \mathit{F21}\}, \{\mathit{F1}\})}$$

Mason(Bryce)A5':

$$\frac{\mathit{F18} \in \mathit{precPfactors}(Mason, Bryce)}{\neg \mathbf{CS2}(Mason, Bryce, \{\mathit{F6}, \mathit{F21}\}, \{\mathit{F1}\})}$$

U2.1 undercuts the argument by suggesting that it may have been the additional plaintiff factors available in the *prec* that tipped the balance, and so distinguishing the *curr* and the *prec*. Like U1.1, U2.1 can be undercut if we can downplay the distinction.

U2.1.1$(curr, prec, f_1, f_2, p, d)$:

$$\frac{\begin{array}{c} f_1 \in \mathit{precPfactors}(curr, prec), \\ f_2 \in \mathit{pFactors}(curr), \\ \mathit{substitutes}(f_1, f_2) \end{array}}{\neg \mathbf{U2.1}(curr, prec, p, d)}$$

U2.1.2$(curr, prec, f_1, f_2, p, d)$:

$$\begin{array}{c} f_1 \in \mathit{precPfactors}(\mathit{curr}, \mathit{prec}),\\ f_2 \in \mathit{dFactors}(\mathit{curr}),\\ \mathit{cancels}(f_1, f_2) \\ \hline\hline \mathbf{U2.1}(\mathit{curr}, \mathit{prec}, p, d) \end{array}$$

We instantiate U2.1.1, which undercuts U2.1:

Mason(Bryce)A6:

$$\begin{array}{c} \mathit{F4} \in \mathit{precPfactors}(\mathit{Mason}, \mathit{Bryce}),\\ \mathit{F6} \in \mathit{pFactors}(\mathit{Mason}),\\ \mathit{substitutes}(\mathit{F4}, \mathit{F6}) \\ \hline\hline \neg\mathbf{U2.1}(\mathit{Mason}, \mathit{Bryce}, \{\mathit{F6}, \mathit{F21}\}, \{\mathit{F1}\}) \end{array}$$

At this point we have: the main argument for the plaintiff based on a particular *prec*, comprising an application of a preference and an argument for the preference; undercutters of these two subarguments; and undercutters of some of these undercutting arguments. We may still, however, have some strengths of the *curr* unused, and so we can add some supplementary arguments.

CS3$(\mathit{curr}, \mathit{prec}, f_1, p, d)$:

$$\begin{array}{c} \mathit{commonPfactors}(\mathit{curr}, \mathit{prec}) = p,\\ \mathit{commonDfactors}(\mathit{curr}, \mathit{prec}) = d,\\ \mathit{preferred}(p, d),\\ f_1 \in \mathit{currPfactors}(\mathit{curr}, \mathit{prec}),\\ \neg\exists f_2(f_2 \in \mathit{dFactors}(\mathit{curr}) \land \mathit{cancels}(f_2, f_1)),\\ \neg\exists f_3(f_3 \in \mathit{pFactors}(\mathit{prec}) \land \mathit{substitutes}(f_1, f_3)) \\ \hline\hline \mathit{outcome}(\mathit{curr}) = \mathit{Plaintiff} \end{array}$$

CS4$(\mathit{curr}, \mathit{prec}, f_1, p, d)$:

$$\begin{array}{c} \mathit{commonPfactors}(\mathit{curr}, \mathit{prec}) = p,\\ \mathit{commonDfactors}(\mathit{curr}, \mathit{prec}) = d,\\ \mathit{preferred}(p, d),\\ f_1 \in \mathit{precDfactors}(\mathit{curr}, \mathit{prec}),\\ \neg\exists f_2(f_2 \in \mathit{currPFactors}(\mathit{curr}) \land \mathit{cancels}(f_2, f_1)),\\ \neg\exists f_3(f_3 \in \mathit{dFactors}(\mathit{curr}) \land \mathit{substitutes}(f_1, f_3)) \\ \hline\hline \mathit{outcome}(\mathit{curr}) = \mathit{Plaintiff} \end{array}$$

These arguments make use of the factors not used to substitute or cancel factors cited to undercut the arguments for the plaintiff based on the *prec*. Thus CS3 points to additional plaintiff factors in the *curr* that were not used to cancel or substitute for factors otherwise used. $CS4$ does the same thing in terms of factors that made the defendant's case stronger in the *prec*. Note that both require the *preferred(P, D)* as a premise, and so must use CS2 to establish this. This seems, from a logical point of view, somewhat odd, since the premises of CS1 are a subset of CS3 and the conclusion is the

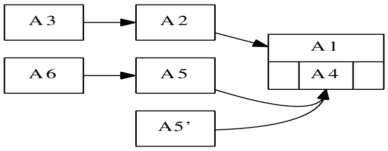

Figure 2: Mason(Bryce) partial Argument Graph

same. Traditionally in work on computational argumentation, arguments are defined so that the premises should be a minimal subset from which the conclusion may be derived [9]. Yet these are presented as arguments in CATO, and so we need schemes for them if we are to reconstruct CATO. Essentially these arguments, which appear in CATO as the move *emphasise strengths*, are intended to have a kind of rhetorical force, rather than a logical force. From a logical point of view, the case is already won, but in order to stress how superior the plaintiff's position is, his advocate adds that not only has the preference been established, but there remains all this unused ammunition which could have countered stronger arguments against the position. The idea seems to be to reassure the judge deciding for the plaintiff that the decision is not a close one, but quite clear and convincing.

3.5 Running Example Result

We have the following defeat relations between arguments, which are represented in Figure 2, where we indicate that **Mason (Bryce) A4** is a subargument of **Mason (Bryce) A1**:

- defeat(**Mason(Bryce)A2**, **Mason(Bryce)A1**)
- defeat(**Mason(Bryce)A3**, **Mason(Bryce)A2**)
- defeat(**Mason(Bryce)A5**, **Mason(Bryce)A4**)
- defeat(**Mason(Bryce)A5′**, **Mason(Bryce)A4**)
- defeat(**Mason(Bryce)A6**, **Mason(Bryce)A5**)

Following [14] and the assumption in ASPIC+ that an attack on a subargument is an attack on the argument, there is a unique extension, containing {**Mason(Bryce)A6**, **Mason(Bryce)A5′** and **Mason(Bryce)A3**}. In particular, **Mason(Bryce)A1** does not appear in any extension as its subargument is defeated by the unattacked **Mason(Bryce)A5′**. While the cases have common factors, *Bryce* was decided in favour of the plaintiff, and the preference for the decision holds, we have not succeeded in eliminating all significant distinctions; in particular, we have not found a substitution for $F18$. Were we to have found such a substitution, then we would have a successful attack on **Mason(Bryce)A5′**, in which case the extension would contain {**Mason(Bryce)A6**, **Mason(Bryce)A4**, **Mason(Bryce)A3**, **Mason(Bryce)A1**}. The extension would not contain **Mason(Bryce)A2**, **Mason(Bryce)A5**, or **Mason(Bryce)A5′**. Given such a substitution, *Bryce* would have been a good precedent for *Mason* as informally discussed previously.

Though this might appear to be a negative result, we can transform it into a positive result by finding an argument against **Mason(Bryce)A5′**, which requires that we substitute or cancel $F18$ based on comparable factors in the factor hierarchy. We might argue that *F18 Identical-Products* holds in both cases, but was too obvious to be explicitly mentioned in *Mason*, and so was omitted from the initial analysis performed for CATO. Alternatively, we could argue that $F18$ should be seen as providing too weak a factor to distinguish the cases. As another possibility, we can argue that **Mason(Bryce)A6** should rest on resolving the relative strength of $F4$ and $F6$, if that becomes an issue. In all three instances, we would need to argue about the factors themselves, which is the subject of the next section.

3.6 Rationality postulates

To prove that our argumentation theory satisfies the rationality postulates of consistency and strict closure, the following properties need to be proven ([21, 20]):

- $\mathcal{R}_s$ is closed under contraposition or transposition.
- Strict consequence is *c-classical* in that if $S \vdash \varphi, \neg\varphi$, then any maximal subset of S strictly implies the negation of the remaining element. (Here $S \vdash \varphi$ means that there exists a strict argument for φ with all premises taken from S.)
- AT is well-formed in that if φ is a contrary of ψ then $\psi \notin \mathcal{K}_n$ and ψ is not the consequent of a strict rule.
- The argument ordering is *reasonable* as defined in [21, 20].
- The closure of $\mathcal{K}_n$ under strict rules is consistent.

The first three properties are immediate from the fact that $\mathcal{L}$ is a first-order language and $\vdash$ in our case corresponds to first-order consequence. Above we used an argument ordering in which all strict-and-firm arguments are preferred over all other arguments and all non-strict-and-firm arguments have equal strength: given this it is easy to show that the argument ordering is reasonable. It remains to show that the closure of $\mathcal{K}_n$ under strict rules is consistent. Since in our case $\mathcal{L}$ is a first-order language and $\mathcal{R}_s$ and $\vdash$ correspond to first-order consequence, this can be proven by specifying a first-order model in which all our axioms are true.

Proposition 3.1 The closure of $\mathcal{K}_n$ under strict rules of the argumentation theory specified above is consistent.

Proof:

We construct a model and then verify that all elements of $\mathcal{K}_n$ are true in the model. Then completeness of first-order logic implies the proposition. The model contains just one factor, one precedent and one current case, where both cases share the factor as a pro-plaintiff factor and both cases are won by the plaintiff[8].

[8] Although this example is minimal it is not unrealistic. It would fit, for example, the representation of *Pierson v Post* in [8].

For ease of notation, we equate below the various model elements with the language elements that denote them, letting the context disambiguate. Capital I stands as usual for the interpretation function of the language in the model.

- The sorts, relations and functions of the model are those corresponding to, respectively, the sorts, predicates and function symbols of $\mathcal{L}$.
- Individuals:
  - *Curr* of sort *curr* and *Prec* of sort *prec* (recall that both are subsorts of the sort *case*)
  - *Plaintiff* and *Defendant* of sort *parties*
  - $F1$ of sort *factors*
  - $\emptyset$ and $\{F1\}$ of sort *sets*
- Interpretation of predicates (those not listed are empty):
  - $I(pFactor) = \{F1\}$
  - $I(outcome) = \{Plaintiff\}$
  - $I(hasPfactor) = \{(Curr, F1), (Prec, F1)\}$
  - $I(hasFactor) = \{(Curr, F1), (Prec, F1)\}$
  - The interpretation of $\subseteq$ and $\in$ is obvious and left implicit.
- Interpretation of functions:
  - $Factors(Curr) = \{F1\}$
  - $pFactors(Curr) = \{F1\}$
  - $Factors(Prec) = \{F1\}$
  - $pFactors(Prec) = \{F1\}$
  - $dFactors(Curr) = \emptyset$
  - $dFactors(Prec) = \emptyset$
  - $commonPfactors(Curr, Prec) = \{F1\}$
  - $commonDfactors(Curr, Prec) = \emptyset$
  - $currPfactors(Curr, Prec) = \emptyset$
  - $currDfactors(Curr, Prec) = \emptyset$
  - $precPfactors(Curr, Prec) = \emptyset$
  - $precDfactors(Curr, Prec) = \emptyset$
  - $outcome(Curr) = Plaintiff$
  - $outcome(Prec) = Plaintiff$
  - The interpretation of $\{\}$, $\cup$, $\cap$ and $\setminus$ is obvious and left implicit.

We now verify that all axioms are true in this model.

- Axiom 1 is clearly true for the following three cases: $I(s) = I(s') = \emptyset$; $I(s) = I(s') = \{\mathit{F1}\}$; $I(s) = \emptyset$ while $I(s') = \{\mathit{F1}\}$. Since these are all cases that can arise, Axiom 1 is universally true. In the same way it is easy to verify that Axioms 2 - 4 are true.

- For axiom 5 note first that this axiom is in fact a scheme for a set of axioms and that in our case we only need to consider the version with two variables s and x_1. Then two cases have to be considered, in both of which we have $I(x_1) = \mathit{F1}$. The first case is when $I(s) = \emptyset$. Then the left-hand side of the equivalence equals to $\emptyset = \{\mathit{F1}\}$, which is false in the model. In this case the right-hand side reduces to $\mathit{F1} \in \emptyset$, which is also false in the model, so the equivalence is true. The second case is when if $I(s) = \{\mathit{F1}\}$. Then the left-hand side of the equivalence equals to $\{\mathit{F1}\} = \{\mathit{F1}\}$, which is true in the model. In this case the right-hand side reduces to $\mathit{F1} \in \{\mathit{F1}\}$, which is also true in the model, so the equivalence is again true.

- Axiom 6 is true since $I(\mathit{dFactor}) = \emptyset$.

- Axiom 7 is true since there is only one individual of sort *factor*, namely $\mathit{F1}$, and we have that $I(\mathit{pFactor}) = \{\mathit{F1}\}$ and $I(\mathit{hasPfactor}) = I(\mathit{hasFactor}) = \{(\mathit{Curr}, \mathit{F1}), (\mathit{Prec}, \mathit{F1})\}$. Likewise, Axiom 8 is true since $I(\mathit{dFactor}) = I(\mathit{hasPfactor}) = \emptyset$.

- Axiom 9 is true since our model contains only one factor and two cases. For the first case we have $\mathit{Factors}(\mathit{Curr}) = \{\mathit{F1}\}$ so $\mathit{F1} \in \mathit{Factors}(\mathit{Curr})$ and we have $(\mathit{Curr}, \mathit{F1}) \in I(\mathit{hasFactor})$. The second case (with *Prec*) is identical. So both sides of the equivalence are true for all f and case. Axiom 10 can be verified in the same way. Axiom 11 is true since the model has no pro-defendant factors.

- For axioms 12-17 note that all three variables can be instantiated in only one way. Then axiom 12 is true since $\mathit{commonPfactors}(\mathit{Curr}, \mathit{Prec}) = \{\mathit{F1}\}$, $\mathit{pFactors}(\mathit{Curr}) = \mathit{pFactors}(\mathit{Prec}) = \{\mathit{F1}\}$ and $\{\mathit{F1}\} \cap \{\mathit{F1}\} = \{\mathit{F1}\}$. Similarly, axiom 12 is true since $\mathit{commonDfactors}(\mathit{Curr}, \mathit{Prec}) = \emptyset$, $\mathit{dFactors}(\mathit{Curr}) = \mathit{dFactors}(\mathit{Prec}) = \emptyset$ and $\emptyset \cap \emptyset = \emptyset$. Axiom 14 is true since $\mathit{currPfactors}(\mathit{Curr}, \mathit{Prec}) = \emptyset$, $\mathit{pFactors}(\mathit{Curr}) = \{\mathit{F1}\}$, $\mathit{commonPfactors}(\mathit{Curr}, \mathit{Prec}) = \{\mathit{F1}\}$ and $\{\mathit{F1}\} \setminus \{\mathit{F1}\} = \emptyset$. Axioms 15, 16 and 17 can be verified in the same way.

- Axioms 18-20 are true since the interpretations of the predicates *ancestor*, *parentFactor*, *substitutes* and *cancels* are all empty.

- Finally, the domain-closure and unique-names axioms must be verified. The relevant domain closure axioms are:

  - $\forall \mathit{case}, \mathit{factor}. \mathit{hasFactor}(\mathit{case}, \mathit{factor}) \equiv$
    $((\mathit{case} = \mathit{Curr} \wedge \mathit{factor} = \mathit{F1}) \vee (\mathit{case} = \mathit{Prec} \wedge \mathit{factor} = \mathit{F1}))$
  - $\forall \mathit{case}. \mathit{case} = \mathit{Curr} \vee \mathit{case} = \mathit{Prec}$

- $Curr \neq Prec$
- $\forall factor.factor = F1$

It is straightforward to show that these are true in the model.

Corollary 3.2 The Dung-style argumentation framework corresponding to the argumentation theory defined above satisfies all four rationality postulates as formulated in [21].

4 Beyond Factor-Based Reasoning

Thus far we have considered reasoning from cases represented as sets of factors to their outcomes. This has been the focus of much of the work on cased-based reasoning in AI and Law, and the understanding of this aspect of reasoning with cases is quite mature. This has enabled us to propose a set of argumentation schemes to capture this reasoning with some confidence, which we believe can serve as a stable, sound and useful basis for determining the nature and properties of these aspects of reasoning with legal cases. There is, however, a lot more to reasoning with legal cases than this: cases do not arrive neatly packaged as bundles of factors, but as rather messy collections of facts or, even worse, as dossiers of conflicting evidence on the basis of which the facts must be established. Unfortunately there is as yet no generally accepted model of how the cases should be analysed into factors that we can use as we used CATO above, and so here we can do no more than provide pointers as to the way forward. Accordingly in this section we will discuss some of the additional aspects of reasoning with cases which we believe need to be further investigated.

Once the facts of a case have been established - and this is rarely straightforward since the move from evidence to facts is often itself the subject of debate - legal reasoning can be seen, following Ross [25] and more recently [19], as a two stage process, first from the established facts to intermediate predicates, and then from these intermediate predicates to legal consequences. CATO has been explicitly identified with the second of these steps (e.g. [11]). Finding these intermediate predicates is by no means simple, and different intermediate concepts require different strategies. Some can be given by listing facts, which supply sufficient, and possibly collectively necessary, conditions while others require consideration of a range of facts, none of which supply sufficient or necessary conditions. Moreover, as argued in [6], which factors hold of a case or which side is favoured by a particular fact may be the whole point. It is even sometimes necessary to argue about what factors there are. To tell more of the story of reasoning with cases, therefore, it is necessary to consider the step from facts to factors. We will first say something about the use of *dimensions* rather than *factors*, and then briefly consider several issues:

- reasoning about what factors hold in a case relative to the facts of a case;
- reasoning about exclusory relations between factors;
- reasoning about factors along dimensions; and

- reasoning about factors and values.

Our preliminary attempts to provide ASPIC+ formalisations of the information and schemes related to these issues can be found in [23].

4.1 Dimensions in Legal Case-Based Reasoning

Dimensions, rather than discrete factors, were used in Rissland and Ashley's HYPO [2], the system from which CATO was developed. Since factors as in CATO predominate in the literature [8, 22, 7], some background discussion on and justification for dimensions is warranted. Dimensions have an *extent* and values along the extent. In contrast to factors, which are either simply present or absent, a dimension, if present, may favour the plaintiff *or* the defendant *to a particular degree*. Dimensions encompass a range of values, with the extreme pro-plaintiff value at one end and the extreme pro-defendant value at the other. Thus, at some initially undetermined point along the range the dimension will cease to favour the plaintiff and start to favour the defendant. Dimensions and factors are, however, related.

In one relationship, factors are intervals along the (continuous) dimension and ordered with respect to one another; in other words as in [7], factors can be taken as the *values* positioned along a dimension[9]. For example, one dimension in HYPO is *Secrets-Voluntarily-Disclosed*, and ranges from 0 to 10,000,000 disclosees, 0 being the pro-plaintiff direction. In CATO, this dimension is expressed as several factors which can be seen as having different degrees of strength. There is a pro-defendant factor *Secrets-Disclosed-Outsiders*, which is present if any disclosure at all had been made, effectively stating that the dimension favours the defendant rather than the plaintiff if a single person is disclosed to, and after that no further force is given to the defendant if there are a million disclosures. In this respect it is a relatively weak factor for the defendant. In addition, there is a *Disclosure-In-Public-Forum* factor, which is intended to cover extensive non-specific disclosure. This is a stronger factor for the defendant. If the latter, stronger, factor applies, then the former does not. Thus, we must reason not only with respect to the factors that hold of a case, but also with the *relative strength* of the factors one to the other. A number of HYPO dimensions are Boolean and counted as present only for one end of the range (e.g. *Common-Employee-Sole-Developer*), and these map straightforwardly to a single CATO factor.

Some other dimensions found in HYPO and used as the basis for CATO factors are not related by a strength ordering relative to some measurable parameter. Most interesting is the HYPO dimension *Security-Measures-Adopted*, which has a range (from pro-defendant to pro-plaintiff):

- *Minimal-Measures*, *Access-To-Premises-Controlled*, *Restrictions-On-Entry-By-Visitors*, *Restrictions-On-Entry-By-Employees*, *Product-Marked-Confidential*, *Employee-Trade-Secrets-Program-Exists*, *Restrictions-On-Hardcopy-Release*, *Employee-Non-Disclosure-Agreements*.

Whereas in HYPO, the fact that, for example, *Employee-Non-Disclosure-Agreements* favours the plaintiff more than *Restrictions-On-Entry-By-Employees*, is indicated by

[9] This is related to a general phenomenon in cognition of *categorial perception* [18].

its position of the dimension, in CATO this is indicated by the presence of several factors. If in HYPO *Employee-Non-Disclosure-Agreements* is satisfied, CATO would have three factor present together: all of *Security-Measures*, *Outsider-Disclosures-Restricted*, and *Agreed-Not-To-Disclose*. If, however, in HYPO only *Restrictions-On-Entry-By-Employees* was reached, in CATO there would be just one factor, *Security-Measures*. That is to say, as we move along the HYPO dimension we collect more and more CATO factors. The increased degree of support is thus given by the cumulative effect of several factors, rather than distinguishing the difference of degree of support given by different positions on the dimension. The connection between the factors is, however, lost.

Although factors dominated thinking about this style of reasoning in AI and Law for some time (e.g. [8, 22, 7]), the need for dimensions was argued for in [6]. Chief amongst the reasons was that the key issue of the case may be about where along the dimension a factor falls and, having situated it, whether the factor favours the plaintiff or defendant. The classic *Pierson v Post* is an example: the dispute turns on when pursuit can be counted as justifying possession, for which different degrees of progress towards bodily possession need to be recognised[10]. Contrast this with the representation based on factors in, for example [8], where the case is assigned the factor *Caught*, and Post is then left without an argument[11].

We can further illustrate the issues using *Pierson v Post* as basis for further reasoning about the factors and schemes that apply in cases. We will take Post as the plaintiff, as in the original action. In [8], the only factors present are *Not-Caught* and *Open-Land*, both of which are pro-defendant. Thus any case found for the defendant where the incident had taken place on open land and the plaintiff had not caught the animal would serve as a precedent; the plaintiff had nothing on which to base the plaintiff's case, all additional factors in the chosen precedent strengthening the defendant's case. In fact the argument put forward for the plaintiff was that Post was sufficiently close to, and sufficiently certain of, taking bodily possession of the fox that it should be *counted as* caught.

Essentially this is an argument against the presence of a factor favouring the defendant, and an argument in favour of the presence of a factor favouring the plaintiff. What this means in ASPIC+ terms is that the status of the factors attributed to the case cease to be axioms and become instead premises requiring justification. What form might this justification take?

In *Pierson v Post*, the defendant's argument was in terms of particular authorities[12]. Tompkins, arguing for the defendant, cites Justinian, Fleca, and Bracton, all of whom seem to say that actual bodily possession is required, and Puttendorf and Barbeyrac, who seem to allow some latitude, but still require mortal wounding. Livingston, arguing for the plaintiff, claims that certain capture would also be enough for Barbeyrac, but also says that it should be so found in this case for the teleological purpose of encour-

[10] In brief the facts were these. Post was chasing a fox with horse and hounds and had cornered it when Pierson intervened and killed it with a fence pole. Post sued Pierson for taking his fox. On appeal, Pierson won on the grounds that only by mortally wounding or seizing the animal can one acquire possession of it, not simply by pursuing it.

[11] For a more recent attempt to represent *Pierson v Post* with factors see [5].

[12] For a detailed reconstruction of the arguments in this case see [16].

aging the destruction of vermin. Neither of these lines of argument are in themselves case-based or reasoning on the basis of the relationship of facts and factors, but make use of generic argumentation schemes such as *Argument from Authority* and *Sufficient Condition Scheme for Practical Reasoning* found in works such as [27].

4.2 Reasoning from Facts to Factors

Suppose an undecided case concerning capturing a wild animal is being argued where the plaintiff claims that the animal was *caught* on the basis of hot pursuit and inevitable capture. Moreover, suppose we take *Pierson v Post* as a *precedent*. In that case the argument that the plaintiff should be counted as having caught the animal was put forward but not sustained. Rather the precedent establishes that the pro-defendant factor *Not-Caught* applies. But to argue that the pursuit should count as possession, even unsuccessfully, requires that cases are not only represented in terms of factors (e.g. *Not-Caught*) but also in terms of the underlying *facts* which, in effect, support the factor assignment (e.g. *Hot-Pursuit* and *Inevitable-Capture*). In addition, it indicates that we must have available ways to argue for relationships between case facts and factors; for example, that *Pierson v Post* can be seen as establishing that *Hot-Pursuit* and *Inevitable-Capture* do not imply *Caught*, despite arguments that they should.

In HYPO there are some procedures that determine whether a dimension applies in terms of facts stored about the case, in effect, providing schemes to reason about the factor category in the case. These schemes are not provided with any justification and no source is given, but are simply hard-coded into the dimension frames. To make this step of the reasoning more accessible, we might identify a set of argumentation schemes so that the relation of factors to facts can be explicitly discussed.

4.3 Reasoning about Factor Incompatibility

In section 3.4, we provided rules for arguing about substituting or cancelling factors in relation to the factor hierarchy. There is, however, another way to reason about factors, done in CATO by the analyst rather than the system. In CATO, when analysing cases the analyst is required to respect the fact that some pairs of factors are incompatible, so that the presence of one factor in a case provides an argument against the presence of another factor in that case. This is obvious in the case of clearly dichotomous factors such as *Caught* and *Not-Caught*, but it is much more widespread than this in CATO. In [1] each factor has a textual explanation of when it does and does not apply. Often the latter includes circumstances where some other factor *does* apply. In [1] we have:

> **F20 Info-Known-To-Competitors(d)**
>
> **Description:** Plaintiff's information was known to competitors.
>
> This factor shows that plaintiff's information was known in the industry or available from sources outside plaintiff's business.
>
> **The factor applies if:** The information plaintiff claims as its trade secret is general knowledge in the industry or trade.

> **The factor does not apply if:** Competitor's knowledge of plaintiff's information results solely from disclosures made by plaintiff. (In this situation, $F10$ applies.) Or if the information could be compiled from publicly available sources, but there was no evidence that competitors had actually done so. (In this situation, $F24$ applies.)

$F10$ is *Secrets-Disclosed-Outsiders* and $F24$ is *Info-Obtainable-Elsewhere*. Thus $F10$ and $F24$ are incompatible with $F20$ and with one another.

If we could make this aspect of the reasoning explicit, using additional predicates to represent which factors exclude one another and additional schemes to enable arguments for and against the presence of factors based on this information, another step in reasoning with legal cases could potentially be transferred from human analyst to the system, and made available for discussion within the system, rather than be necessarily taken as a given.

4.4 Reasoning along Dimensions

In section 4.1, we recognised that factors may favour their party to different extents. In light of this, we might wish to reconsider our notions of cancellation and substitution. Arguments based on substitution and cancellation were used to undercut arguments distinguishing cases. Given that undercutters always defeat the arguments they attack, these are powerful arguments. But suppose the factors in question were $F20$ and $F24$, as defined above. It is clear from the description that $F20$ is intended to be more pro-defendant than $F24$, for $F20$ represents an actual rather than a merely possible state of affairs. Thus if we have $F24$ in a *curr* and $F20$ in a *prec*, there is no problem in substitution: when considering the two under the common abstract factor the plaintiff's case is stronger, because the factor for the defendant is weaker in the *curr*. But if plaintiff attempts to argue that $F24$ in a *prec* substitutes for $F20$ in a *curr*, undercutting an instance of U1.1, the issue is less clear. The defendant can at least argue that $F24$ is not strong enough to be substituted for $F20$. This has been handled in different ways in different applications: CATO indicated different degrees of influence by distinguishing *thin* and *fat* links in the factor hierarchy. IBP [10], which developed from CATO, introduced the idea of knock-out factors, which could be neither substituted for nor cancelled - indeed were entirely decisive with regards to a particular issue.

The most quantitative approach can be found in Chorley's AGATHA [13]. AGATHA constructs theories as defined in [7] by heuristic search over a search space of theories of case law derived from CATO cases constructed using the operators defined in [7]. To support heuristic search, the theories need to assessed using an evaluation function. Part of this evaluation function in AGATHA required every factor to be assigned a weight according to the importance of its dimension and its position on that dimension. This meant that cases could be assigned a score reflecting the weighted sum of the factors present in them. Although good performance was achieved by AGATHA, the assignment of the weights was pragmatic rather than based on any principled analysis.

We believe that a representation making explicit the information that is required and the schemes that can be used to argue about the comparative strength of factors would help to clarify how we should resolve these difficult issues.

4.5 Reasoning about Value Preferences

In work such as [7] the body of precedents used in systems such as CATO are taken as the basis for constructing a theory of the relevant domain intended not only to *record* preferences, but also to *explain* them. Their approach, following [8], was to link factors with social values, the idea being that a factor favoured a particular party because deciding for that party when that factor was present in a case would promote that value. Preferences between cases can then be expressed as preferences between sets of values. The process then becomes one of arguing for value preferences on the basis of the precedents and applying these preferences to new cases. This step of the reasoning could also be captured as a set of argumentation schemes, which we intend to do as future work.

5 Discussion

By articulating the process of reasoning with precedent cases as sets of argumentation schemes and their critical questions, we can see it as a sequence of stages in a dialogue between **P**laintiff and **D**efendant, which are as follows, where every option is available:

1. **P**: Assert that the decision should be in favour of the plaintiff since factors favouring the plaintiff are preferred to factors favouring the defendant;
   (a) **D:** Cite additional points in favour of the defendant
   (b) **P:** Substitute for, dismiss, or cancel these additional strengths
   (c) **D:** Dispute strength of substituting or cancelling factors
2. **P:** Identify a precedent case that justifies a preference applicable to the current case;
   (a) **D:** Cite additional points favouring the plaintiff in the precedent
   (b) **P:** Substitute for, dismiss, or cancel these additional strengths
   (c) **D:** Dispute strength of substituting or cancelling factors
   (d) **D:** Identify a precedent case that justifies a preference for the defendant applicable to the current case
      i. **P:** Cite additional points in favouring the defendant in the precedent:
      ii. **D:** Substitute for, dismiss, or cancel these additional strengths
      iii. **P:** Dispute strength of substituting or cancelling factors
3. **D:** Dispute which factors are present in the current case
   (a) **P:** Defend original factors

Different systems will support more or fewer of these stages. At one extreme we have a neural network style system such as that described in [3] in which the system acts as a black box taking factors (or facts) as an input and expressing a preference based on its internalisation of the set of precedents. Such a system supports only step 1. CATO, from which our discussion began, supports the identification of the preference in 2, the distinguishing moves in 1a and 2a (although it does not discriminate between them), and the counter example move of 2d. CATO also supports the downplaying of 1b and 2b, but does not distinguish between substitution and cancellation. HYPO links facts and dimensions, and so can explain 3, but not support argument about it. Hypothetical arguments in HYPO were intended to explore the issues raised in 1c, 2c, and 2d(iii), but this aspect of HYPO was never fully developed in [2]. These considerations are also used internally in the most advanced version of Chorley's AGATHA [13], although the resulting arguments are not transparent to the user.

Representation in terms of ASPIC+ identifies the underlying knowledge base required by each stage. Given such knowledge in the KB, the specification of the argumentation schemes in this paper would permit straightforward implementation, using a defeasible reasoner to instantiate the schemes from the KB, identifying the attack relations, and then evaluating them as in a Dungian argumentation framework. The first of these steps, identifying the arguments, is achieved for stages 1, 1a, 1b, 2, 2a and 2b, using Prolog as the defeasible reasoner in a program described in [5].

Another benefit from representing these arguments in terms of ASPIC+ is that we can regard cases described under factors as but one source of arguments, as in [17]. At the top level, stage 1, there may be arguments for the defendant rebutting our case-based argument for the plaintiff and these arguments may be based on cases, authority, purpose, or whatever other kind of argument our opponent wishes to advance. Similarly, the premises of our arguments often require other, generic, argumentation schemes, such as authority and purpose, to justify them. By providing a framework in which all kinds of argument can be represented equally, we can readily provide a framework in which reasoning of many different kinds can be deployed. Note that this is done without recasting the various distinctive case-based aspects of CATO style arguments uniformly as ordinary rules, as was the case in e.g. [22].

A final important insight is gained by recognising that the above indicates at which points choice is possible, and at which points the judgement is constrained. Let us relate this to the steps above. At step 1 we may get arguments, constructed with a variety of schemes, for and against deciding for the plaintiff, which conflict through rebuttal and so can be decided through preferences. The attack of 1a, however, cannot be rejected on the grounds of preference, but can only be defeated by 1b, which in turn can only be defeated by an argument from 1c. Arguments in stage 1c itself, however, may be resolved on grounds of preference. Similarly although the rebuttals arising at 2d may be decided by preferences, 2a can only be defended by 2b, and 2b by 2c, at which stage preferences may be used to resolve competing arguments. When considering 2d, only at 2d(iii) do preferences play a role. Thus although we may think of case-based reasoning as involving a choice between the plaintiff and the defendant arguments, in fact, choice operates at a number of quite specific, fine-grained points in the debate.

6 Concluding Remarks

In this paper, we have clarified a range of aspects of legal case-based reasoning with factors using formal defeasible arguments modelled within the ASPIC+ framework.

The choice of ASPIC+ has made it possible to prove consistency and closure results for our formalisation by exploiting the metatheory of ASPIC+. Our formalisation has also illustrated the potential of the ASPIC+ framework for formalising reasoning with argumentation schemes. The schemes reconstructing CATO are proposed as a definitive way of capturing the reasoning from factors to decision. In Section 4 we have discussed some issues relating to further aspects of legal reasoning with cases that have received rather less attention, and about which no consensus as yet exists. In future work, we will look to further extend our approach, by providing formally expressed schemes for these additional aspects, and also for reasoning to and with value preferences as found in [7].

7 Acknowledgement

This work was partially supported by the European IMPACT Project (***247228***). The views expressed are, however, those of one or more of the authors, and should not be taken as necessarily representative of the project.

References

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

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

[3] T. Bench-Capon. Neural networks and open texture. In *Proceedings of the Fourth International Conference on Artificial Intelligence and Law (ICAIL 1993)*, pages 292–297, Amsterdam, 1993. ACM.

[4] T. Bench-Capon. Arguing with cases. In A. Oskamp, R. D. Mulder, C. van Noortwijk, C. Grutters, K. Ashley, T. Bench-Capon, and T. Gordon, editors, *Proceedings of the Twentieth Annual Conference on Legal Knowledge and Information Systems (JURIX 1997)*, pages 85–100, Amsterdam, 1997. The Foundation for Legal Knowledge Systems.

[5] T. Bench-Capon. Representing Popov v. Hayashi with dimensions and factors. *Artificial Intelligence and Law*, 20(1):15–35, 2012.

[6] T. Bench-Capon and E. Rissland. Back to the future: dimensions revisited. In B. Verheij, A. Lodder, R. Loui, and A. Muntjewerff, editors, *Proceedings of the Twentieth Annual Conference on Legal Knowledge and Information Systems (JURIX 2001)*, pages 41–52, Amsterdam, 2001. IOS Press.

[7] T. Bench-Capon and G. Sartor. A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence*, 150(1-2):97–143, 2003.

[8] 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 (ICAIL 1993)*, pages 50–59, Amsterdam, 1993. ACM.

[9] P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128(1-2):203–235, 2001.

[10] S. Brüninghaus and K. Ashley. Predicting outcomes of case-based legal arguments. In *Proceedings of the Ninth International Conference on Artificial Intelligence and Law (ICAIL 2003)*, pages 233–242, Edinburgh, 2003. ACM.

[11] S. Brüninghaus and K. Ashley. A predictive role for intermediate legal concepts. In D. Bourcier, editor, *Proceedings of the Twentieth Annual Conference on Legal Knowledge and Information Systems (JURIX 2003)*, pages 153–162, Utrecht, 2003. IOS Press.

[12] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171:286–310, 2007.

[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] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[15] T. Gordon, H. Prakken, and D. Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence*, 171:875–896, 2007.

[16] T. Gordon and D. Walton. Pierson v. Post revisited - a reconstruction using the Carneades argumentation framework. In P. Dunne and T. Bench-Capon, editors, *Proceedings of the First International Conference on Computational Models of Argumentation (COMMA 2006)*, pages 208–219. IOS Press, 2006.

[17] T. Gordon and D. Walton. Legal reasoning with argumentation schemes. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Law (ICAIL 2009)*, pages 137–146, Barcelona, 2009. ACM.

[18] S. Harnad. Categorical perception. In *Encyclopedia of Cognitive Science*. Nature Publishing Group: Macmillan, December 2003.

[19] L. Lindahl and J. Odelstad. Open and closed intermediaries in normative systems. In T. van Engers, editor, *Proceedings of the Twentieth Annual Conference on Legal Knowledge and Information Systems (JURIX 2006)*, pages 91–99, Paris, 2006. IOS Press.

[20] S. Modgil and H. Prakken. A general account of argumentation with preferences. *Artificial Intelligence*, 195:361–397, 2013.

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

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

[23] H. Prakken, A. Wyner, T. Bench-Capon, and K. Atkinson. A formalisation of argument schemes for case-based reasoning in ASPIC+. Technical Report UU-CS-2013-002, Department of Information and Computing Sciences, Utrecht University, Utrecht, 2013.

[24] E. Rissland, D. Skalak, and T. Friedman. BankXX: Supporting legal arguments through heuristic retrieval. *Artificial Intelligence and Law*, 4(1):1–71, 1996.

[25] A. Ross. Tu-tu. *Harvard Law Review*, 70:812–825, 1961.

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

[27] D. Walton. *Argumentation Schemes for Presumptive Reasoning*. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 1996.

[28] A. Wyner and T. Bench-Capon. Argument schemes for legal case-based reasoning. In A. Lodder and L. Mommers, editors, *Proceedings of the Twentieth Annual Conference on Legal Knowledge and Information Systems (JURIX 2007)*, pages 139–149, Leiden, 2007. IOS Press.

[29] A. Wyner, T. Bench-Capon, and K. Atkinson. Towards formalising argumentation about legal cases. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Law (ICAIL 2011)*, pages 1–10, Pittsburgh, 2011. ACM.