Precedential Constraint and Argument Schemes for Legal Case-based Reasoning

Henry PRAKKEN [a,1]
[a] *Department of Information and Computing Sciences, Utrecht University, The Netherlands*

Abstract. This paper studies the relation between two formal approaches to legal case-based reasoning: argument-schemes models generating abstract argumentation frameworks and formal models of precedential constraint. First it is shown that a CATO-style argument-scheme model proposed in 2015 by Prakken et al. is unsound and incomplete with respect to Van Woerkom et al.'s factor-based hierarchical result model of precedential constraint. Then a new argument-scheme model is presented and shown to be sound and complete with respect to the result model.

Keywords. case-based reasoning, precedential constraint, argument schemes

1. Introduction

Two developments in the formal study of legal case-based reasoning are the formulation of argument schemes for such reasoning in the context of some general formal model of argumentation and the formulation of models of precedential constraint. To date, these two developments have not been formally related. This paper does so for two representative models, the argument-scheme model in *ASPIC*$^+$ of [14] and the model of precedential constraint of [16], which generalises the 'flat' result model of [7]. Both models are for factor-based reasoning with a factor hierarchy as introduced in the CATO system by Aleven (e.g. [1]). First it will be shown that the *ASPIC*$^+$ scheme model is unsound and incomplete with respect to Van Woerkom et al.'s hierarchical model. Then a new argument-scheme model in *ASPIC*$^+$ is presented that is sound and complete for both models of precedential constraint. A benefit of formulating a scheme model for precedential constraint in *ASPIC*$^+$ is that thus the inputs of a model of precedential constraint can be argumentatively derived from further information.

The new *ASPIC*$^+$ model is in the form of an argument game for [5]'s grounded semantics of abstract argumentation frameworks. Such a game is not only of theoretical interest but can also be the basis for explanations of outcomes of AI applications, including LLM applications. For example, [6] encode [17]'s argument game for a simpler form of case-based reasoning in the prompts for an LLM, to let it generate case-based arguments that explain an outcome.

[1] Corresponding Author: Henry Prakken, h.prakken@uu.nl.

This paper is organised as follows. The formal preliminaries on $ASPIC^+$ and precedential constraint are presented in Section 2, after which the scheme model of [14] is summarised in Section 3. The formal results on this model are presented in Section 4. Then the new scheme model is presented, proven sound and complete and illustrated with an example in Section 5, after which the paper concludes.

2. Formal preliminaries

In this section the $ASPIC^+$ framework and [16]'s hierarchical model of factor-based precedential constraint are outlined.

2.1. Sketch of the ASPIC$^+$ *framework*

The **ASPIC$^+$ framework** [8] defines notions of argument, attack and defeat. For present purposes an informal summary, adapted from [10], suffices. Each step in an argument is of the form $P_1, \ldots, P_n$, *therefore* Q where

- each P_i and Q is a statement in some language,
- The inference step is supposed to be based on an inference rule of the form $P_1, \ldots, P_n \rightarrow / \Rightarrow Q$, where a rule with $\rightarrow$ is *strict* and a rule with $\Rightarrow$ is *defeasible*. Strict rules guarantee that their consequent Q is acceptable if their antecedents $P_1, \ldots, P_n$ are acceptable, while defeasible rules in that case only create a presumption for the acceptability of their conclusion.
- Each P_i is either given as an element of the knowledge base $\mathcal{K}$ or a conclusion from a preceding inference in the argument,
- Q is the conclusion of the inference step.

Arguments are then trees of such inference steps, where the leaves are its *premises*, the links are inference steps, the root is the *final conclusion* and all other nodes are *intermediate conclusions*. Subtrees of an argument are its *subarguments*.

In this paper I assume that the elements of $\mathcal{K}$ are certain. Hence, arguments can be *attacked* in two ways: on their defeasible inference steps (*undercutting attack*) and on their defeasibly derived conclusions (*rebutting attack*). Attacks can be *direct* (on an argument's final inference step of conclusion) or *indirect* (on an intermediate inference step or conclusion). In general, $ASPIC^+$distinguishes between attack and defeat relations between arguments, but in this paper I assume like [14] that there are no preferences, so all attacks succeed as defeats. If A defeats B but B does not defeat A we say that A *strictly defeats* B.

$ASPIC^+$ thus generates an **abstract argument framework** ([5]), i.e., a pair $AF = (\mathcal{A}, \mathcal{D})$, with $\mathcal{A}$ a set of arguments and $\mathcal{D}$ a binary relation of defeat on $\mathcal{A}$. A subset S of $\mathcal{A}$ is said to be *conflict-free* if no argument in S defeats an argument in S 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 S and some argument A_2 in $\mathcal{A}$ but not in S defeats A_1, then some argument in S defeats A_2. A semantics of AFs defines further conditions on admissible sets, resulting in subsets of AFs called extensions. Since the results in this paper do not depend on a particular semantics, I will use the simplest one, called *grounded semantics*, which always yields a unique

grounded extension. Instead of giving its definition, I will in this paper use an *argument game* for testing whether an argument is in the grounded extension of an AF. In this game (called the G-game) a proponent and an opponent of an argument A exchange arguments, taking turns after each move and picking all their arguments from AF. The proponent starts with A and then the opponent has to defeat the last argument of the proponent while the proponent has to strictly defeat the last argument of the opponent. Moreover, the proponent is not allowed to repeat its own moves. A game is won if the other player has no replies. It has been shown that an argument A is in the grounded extension just in case the proponent has a winning strategy for A in the G-game [9].

2.2. Precedential constraint

The formal theory of precedential constraint defines to which extent a set of precedents constrains a decision in a new case. In this paper I will study Van Woerkom's [16] hierarchical factor-based result model of precedential constraint (HRM). The following presentation is taken from [13]. Given a set P of factors, a *fact situation* F is a partial function $F : P \rightharpoonup \{t, f\}$, i.e., a function $F : Q \to \{t, f\}$ for some subset $Q \subseteq P$. Intuitively, $F(p) = t$ means that factor p applies in F while $F(p) = f$ means that factor p does not apply in F, and if F is undefined on p then this means it is not yet known whether p applies in F or not. A fact situation defined on all of P is *complete*. We write $F \models p$ for $F(p) = t$ and $F \models \neg p$ for $F(p) = f$. A *case* is a pair (F, s) where F is a fact situation and $s \in \{\pi, \delta\}$. Here s is the outcome, or decision, of the case. The set P of factors is partitioned into two subsets $\texttt{Pro}$ and $\texttt{Con}$. If $p \in \texttt{Pro}$ ($\texttt{Con}$) then p is said to support outcome p (d). Two corresponding functions $\texttt{Pro}, \texttt{Con} : \{\pi, \delta\} \to 2^P$ are defined as follows: $\texttt{Pro}(s) = \texttt{Pro}$ if $s = \pi$ and $\texttt{Pro}(s) = \texttt{Con}$ if $s = \delta$ and, likewise, $\texttt{Con}(s) = \texttt{Con}$ if $s = \pi$ and $\texttt{Con}(s) = \texttt{Pro}$ if $s = \delta$. Finally, a case base CB is a set of cases.

Now **'flat' precedential constraint** is for complete fact situations defined as follows. Let CB be a case base for a set of factors P and $s \in \{p, d\}$. We say that CB *forces* the decision of a complete fact situation F for s, written $CB, F \models s$, iff there exists a case $(G, s) \in CB$ such that (1) for all $q \in \texttt{Pro}(s)$: if $G \models q$ then $F \models q$, and (2) for all $q \in \texttt{Con}(s)$: if $F \models q$ then $G \models q$. In words, F is forced for s if there exists a precedent case (G, s) such that F contains at least all $\texttt{Pro}(s)$ factors that are in G and at most all $\texttt{Con}(s)$ factors that are in G. This definition is equivalent to Horty's [7] original result model of precedential constraint.

To make this hierarchical, fact situations can be partial, to allow for consulting precedents on whether some factor should apply in the new case. Moreover, a factor hierarchy is assumed, defined as follows.

Definition 2.1. A *factor hierarchy* is a pair (P, H) with P a finite set of propositional letters (factors) and H a relation on P satisfying

1. the transitive closure of H is irreflexive;
2. P contains exactly one H-maximal element;
3. H is equal to a disjoint union $\texttt{Pro} \cup \texttt{Con}$ of relations $\texttt{Pro}$ and $\texttt{Con}$.

A factor $p \in P$ is *basic* if there is no $q \in P$ with $H(q,p)$ and *abstract* otherwise. The sets of all basic and all abstract factors are denoted with B and A, respectively. The set $\mathcal{P}$ is defined as $P \cup \{\neg p \mid p \in P\}$. Finally, $\texttt{Pro}(p) = \{q \in F \mid \texttt{Pro}(q,p)\}$, $\texttt{Con}(p) = \{q \in F \mid \texttt{Con}(q,p)\}$, $\texttt{Pro}(\neg p) = \texttt{Con}(p)$ and $\texttt{Con}(\neg p) = \texttt{Pro}(p)$. The unique H-maximal element is denoted by p and its negation ($\neg p$) by d.

Note that in this model fact situations can contain abstract factors and case outcomes are regarded as part of the hierarchy, so they can also be part of a fact situation. The hierarchical result model HRM is then defined recursively as follows. Because of the definitions of negated factors and the functions Pro and Con, this definition equally applies to atomic and negated factors.

Definition 2.2. Let CB be a case base for a factor hierarchy (P,H) and $f \in \mathcal{P}$ a factor. We say that CB *forces* the decision of a fact situation F for f, written $CB, F \models f$, iff either:

1. $F \models f$; or
2. $f \in A$ and there exists a case $G \in CB$ with $G \models f$ such that
   (a) for all $q \in \texttt{Pro}(f)$: if $G \models q$ then $CB, F \models q$, and
   (b) for all $q \in \texttt{Con}(f)$: if $CB, F \models q$ then $G \models q$

The base case (1) checks whether f is in the new fact situation. The inductive clause (2) checks for abstract factors f whether it is in the fact situation of some precedent such that the new case is at least as good for f as the precedent.

3. The schemes from [14]

I next present the argumentation schemes of [14], which in $ASPIC^+$ are all defeasible rules. They all have dual schemes with the roles of p and d reversed. The formal definitions of the predicates used in the schemes are in [14]. In the schemes p stands for 'plaintiff' and d for 'defendant'. Note that [14] do not assume that $outcome(curr) = p$ and $outcome(curr) = d$ contradict each other.

Scheme CS1 is for citing a precedent to argue for the same decision in the new case while CS2 is for establishing the third premise of CS1.

$$\frac{\begin{array}{l}commonPfactors(curr, prec) = x,\\ commonDfactors(curr, prec) = y\\ preferred(x,y)\end{array}}{outcome(curr) = p}\ \textbf{CS1}(curr, prec, p, d, x, y)$$

$$\frac{\begin{array}{l}commonPfactors(curr, prec) = x\\ commonDfactors(curr, prec) = y\\ outcome(prec) = p\end{array}}{preferred(x,y)}\ \textbf{CS2}(cur, prec, p, d, x, y)$$

Here, $commonP(D)factors(curr, prec)$ are the pro-p (pro-d) factors shared between precedent $prec$ and new case $curr$.

Scheme CS1 has an undercutter in case of new pro-d factors.

$$\frac{f \in currDfactors(curr, prec)}{\neg\textbf{CS1}(cur, prec, p, d, x, y)}\ \textbf{U1.1}(cur, prec, p, d, x, y)$$

Here, $curr(proc)D(P)factors(curr, prec)$ denotes the set of pro-d (pro-p) factors in *curr* (*prec*) that are not in *prec* (*curr*).

This undercutting scheme has two undercutters, which can both be used to downplay a distinction.

U1.1.1$(curr, prec, f_1, f_2, p, d, x, y)$:
$f_1 \in currDfactors(curr, prec)$
$f_2 \in dFactors(prec)$
$\underline{substitutes(f_1, f_2)}$
$\neg$**U1.1**$(cur, prec, p, d, x, y)$

U1.1.2$(curr, prec, f_1, f_2, p, d, x, y)$:
$f_1 \in currDfactors(curr, prec)$
$f_2 \in pFactors(curr)$
$\underline{cancels(f_1, f_2)}$
$\neg$**U1.1**$(cur, prec, p, d, x, y)$

Here, $d(p)Factors(c)$ are all pro-d (pro-p) factors in case c. Scheme U1.1.1 can be used for arguing that the new pro-d factor has an abstract factor in common with the missing pro-d factor, so that the new pro-d factor substitutes the missing one. Scheme U1.1.2 can be used for arguing that the new pro-d factor has an abstract factor in common with a pro-p factor in the new case, so that the new pro-p factor cancels the new pro-d factor.

The next two schemes can be used to rebut, respectively undercut arguments using CS2. Scheme R2.1 is meant for citing a counterexample in which a similar comparison was made in favour of the defendant. Scheme U2.1 can be used when a pro-p factor from the precedent is missing in the new case.

R2.1$(cur, prec, prec2, x, y)$:
$x \subseteq commonPfactors(curr, prec2)$
$y \subseteq commonDfactors(curr, prec2)$
$\underline{outcome(prec) = d}$
$\neg preferred(p, d)$

U2.1$(curr, prec, x, y)$:
$\underline{f \in precPfactors(curr, prec)}$
$\neg$**CS2**$(curr, prec, x, y)$

Like U1.1, U2.1 can be undercut in two ways by downplaying the distinction.

U2.1.1$(curr, prec, f_1, f_2, p, d, x, y)$:
$f_1 \in precPfactors(curr, prec)$
$f_2 \in pFactors(curr)$
$\underline{substitutes(f_1, f_2)}$
$\neg$**U2.1**$(curr, prec, p, d, x, y)$

U2.1.2$(curr, prec, f_1, f_2, p, d, x, y)$:
$f_1 \in precPfactors(curr, prec)$
$f_2 \in dFactors(curr)$
$\underline{cancels(f_1, f_2)}$
$\neg$**U2.1**$(curr, prec, p, d, x, y)$

The paper of [14] contains two further schemes for arguing for a decision, but since CS1 applies when they apply and all three have the same undercutters, they do not need to be separately discussed.

4. Results on soundness and completeness of the model of [14]

This section presents results on the soundness and completeness of [14]'s argument-scheme model relative to [16]'s model of precedential constraint, assuming grounded semantics.

Recall that in the HRM, fact situations of a case can contain abstract factors, which can be seen as intermediate decisions or the final decision of the case. In [14] case decisions are presented separately from their fact situation but to be able to

make a comparison, we include these decisions in each case's fact situation. While this is a notational issue only, this is different for intermediate factors, since in [14] cases cannot contain intermediate factors, while its schemes are only about a case's final outcome. Having said so, the counterexamples to soundness and completeness given below do not contain intermediate factors in their fact situations.

Soundness. The scheme model is unsound with respect to the HRM. Consider an FH for p with p_1 pro p and d_1 con p (see Figure 1(a)). Let $F = \{p_1\}$, $G = \{p, p_1\}$ and $G' = \{d_1\}$. Then $CB, F \not\models p$ since $d_1 \in Con(p)$ and $CB, F \models d_1$ since $G' \models d_1$ and d_1 has no children, but $G \not\models d_1$. Consider next the argument A with CS1 and CS2 that cites G for $outcome(curr) = p$.

$commonPfactors(F, G) = \{p_1\}$,
$commonDfactors(F, G) = \emptyset$,
$outcome(G) = p$,
So (by CS2) $preferred(\{p_1\}, \emptyset)$,
So (by CS1) $outcome(curr) = p$

This argument cannot be undercut by applying U1.1, since that requires that d_1 is in both F and G. So P has a trivial winning strategy consisting just of A.

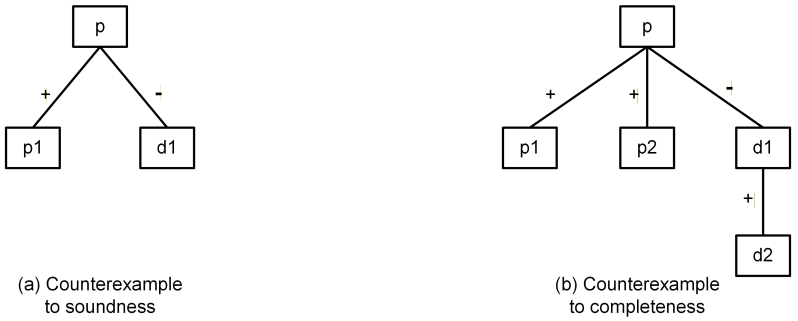


Figure 1. Counterexamples to soundness and completeness

Completeness. The scheme model is incomplete with respect to the HRM. The following is a simple example in which a decision is forced but no argument for it is in the grounded extension. Consider a factor hierarchy for p in which p_1 and p_2 are pro p while d_1 is con p_1 and d_2 is pro d_1 (see Figure 1(b)). Let $F = \{p_1, d_2\}$ and $G = \{p, p_1\}$. Then F is forced for p since , firstly, the only pro-f factors in G are also in F, which satisfies Condition (2a) of Definition 2.2, and, secondly, d_1 is not forced by CB, F (since no case has a decision d_1) and undefined in G so Condition (2b) of Definition 2.2 is also satisfied. In the scheme model, just one argument for $outcome(curr) = p$ can be constructed, with the same form as argument A above. However, now its subargument using CS2 can be undercut with U1.1 applied to d_2, after which neither of the downplaying moves with U1.1.1 and U1.1.2 applies, since there is no pro-p factor in F that can be used to downplay the distinction with d_2. The fact that F is forced for p is a consequence of the design choice in the HRM that decisions on a factor are exclusively based on its children factors (in our example p_1 pro p and d_1 con p). By contrast, in [14] factors are, as in CATO, pro or con a final outcome even if they are not a child of it.

5. A new scheme model

The above negative results for the HRM motivate a new set of argument schemes. The idea is to stay as close as possible to the HRM and to replace downplaying with recursive calls to the scheme for citing precedents on their outcomes. I assume that every case with decision x contains at least one factor that is in $Pro(x)$. Furthermore, p and d are still constants referring to the two possible final decisions of a case.

5.1. The new schemes

The new $ASPIC^+$ modelling is as follows. As for notation, the predicates *pFactor* and *dFactor* are now replaced by *Pro* and *Con*. Note that by the HRM definitions of negations of factors and the functions `Pro` and `Con` the schemes equally apply to atomic and negated factors. The first scheme is for citing a case for one of its decisions.

NCS1$(curr, prec, f)$:
$$\frac{f \in Factors(prec)}{f \in Outcomes(curr)}$$

Here, $x \in Outcomes(curr)$ informally means that x should be a (final or intermediate) decision in *curr*. Furthermore, $\mathcal{K}$ contains $\forall x \forall X(x \in Factors(X) \rightarrow x \in Outcomes(X))$, as well as the following definitions (in which $Parent(x, y)$ refers to the relation between x and y in the factor hierarchy).

$$\begin{array}{ll}
\forall x \forall y(x \in \texttt{Pro}(y) \equiv & \forall x \forall y(x \in \texttt{Con}(y) \equiv \\
(Parent(x, y) \wedge & (Parent(x, y) \wedge \\
\quad (pFactor(x) \wedge pFactor(y)) \vee & \quad (pFactor(x) \wedge dFactor(y)) \vee \\
\quad (dFactor(x) \wedge dFactor(y)))) & \quad (dFactor(x) \wedge pFactor(y))))
\end{array}$$

NCS1 has the following two undercutters:

NU1.1$(curr, prec, f, q)$:
$$\frac{q \in \texttt{Pro}(f) \quad q \in Factors(prec) \quad q \notin Factors(curr)}{\neg\textbf{NCS1}(curr, prec, f)}$$

NU1.2$(curr, prec, f, q)$:
$$\frac{q \in \texttt{Con}(f) \quad q \in Outcomes(curr)}{\neg\textbf{NCS1}(curr, prec, f)}$$

NU1.1 says that NCS1 does not apply since a pro-f factor q that is in *prec* is not in *curr*, while NU1.2 says that NCS1 does not apply since *curr* should be decided for a con-f factor q. Note that NU1.2 avoids counterexample (1b) of Figure 1, since it only requires that $d_1 \in Outcomes(curr)$ and not that both $curr) \models d_1$ and $prec) \models d_1$, unlike U1.1.

These schemes can in turn be undercut as follows:

NU1.1.1$(curr, prec, f, q)$:
$$\frac{q \in Outcomes(curr)}{\neg\textbf{NU1.1}(curr, prec, f, q)}$$

NU1.2.1$(curr, prec, f, q)$:
$$\frac{q \in Factors(prec)}{\neg\textbf{NU1.2}(curr, prec, f, q)}$$

NU1.1.1 says that NU.1.1 does not apply since the pro-f factor missing in *curr* is still forced in *curr*, while NU1.2.1 says that NU1.2 does not apply since the con-f factor forced in *curr* is in *prec*.

Note that the statements $p \in Outcomes(curr)$ and $d \in Outcomes(curr)$ are not contradictory. This is to capture that in the HRM both outcomes can be forced if the case base is consistent.

Proposition 5.1. Any abstract argumentation framework generated by the $ASPIC^+$ model specified above is acyclic.

Proof. (sketch) Consider any $AF = (\mathcal{A}, \mathcal{D})$ and any argument $A \in \mathcal{A}$. Then A's top rule r can be of one of the following forms. No strict argument and no argument applying NU1.1.2 can be defeated. If an argument applying NU1.1 is defeated by using NU1.1.1, then the latter can only be defeated on its (sub)conclusion $q \in Outcomes(curr)$. But such defeaters are all about some $q' \in \texttt{Pro}(q) \cup \texttt{Con}(q)$. Likewise for defeaters of arguments using NCS1 or NU1.2. Then, since the factor hierarchy is acyclic, neither A nor any of its subarguments can defeat some argument in a defeat chain towards A. □

5.2. An example

I now illustrate how the new scheme model could be used to generate dialogues. For reasons of space, the example has to be kept informal. The example is meant to illustrate that the argument game is not just a technical construct but can be the basis for generating arguably realistic dialogues, which can, for instance, explain outcomes of the HRM model of precedential constraint. The dialogue protocol assumed in the example allows the sides to state in one turn all arguments allowed by the G-game at that stage.

P_1: The current case should decide that f since this precedent decided that f.
O_{1a}: But the precedent also decided that q, which is con f and which is not a factor in the current case.
O_{1b}: Moreover, the current case should decide that q, which is con f, since this other precedent decided that q.
P_{2a} (against O_{1a}): But the current case should decide that q precisely since the precedent did so.
P_{2b} (against O_{1b}): Moreover, the other precedent already decided that q, so it took this con f factor into account when deciding that f.
O_3 (against P_{2a}): It is irrelevant that the precedent decided that q, since it also decided that pro-q factor q' holds, which is not in the current case.

Note that since $p \in Outcomes(curr)$ and $d \in Outcomes(curr)$ are not defined to be contradictory, arguments for these outcomes cannot yield rebutting attacks, which might be regarded as less realistic. However, the main point of the game is not to have full correspondence with how humans argue but to have an explainable argument game that is sound and complete with respect to precedential constraint.

5.3. Soundness and completeness

I next prove soundness and completeness of the new $ASPIC^+$ scheme model. As for notation, since $x \in \mathit{Factors}(X)$ is equivalent to $X \models x$, I will in the proofs abbreviate the former to the latter.

Theorem 5.2. If there exists a justified argument for $f \in \mathit{Outcomes}(\mathit{curr})$ then $CB, \mathit{curr} \models f$.

Proof. The proof is with induction on the structure of the factor hierarchy.

Base case: $f \in \mathit{curr}$. Then there is a justified argument for $f \in \mathit{Outcomes}(\mathit{curr})$ that applies $\mathcal{K}$ only. Moreover, since $f \in \mathit{curr}$, $CB, \mathit{curr} \models f$ by condition (1) of Definition 2.2.

Induction hypothesis: For all f and all $q \in \mathtt{Pro}(f)$: if there is a justified argument for $q \in \mathit{Outcomes}(\mathit{curr})$ then $CB, \mathit{curr} \models q$.

Induction step: Suppose there is a justified argument A for $f \in \mathit{Outcomes}(\mathit{curr})$ and consider any winning strategy S for A. If $f \in \mathit{curr}$ then this is the base case. Otherwise, A applies NCS1. Then the following cases must be considered.

(i) If A has no defeaters, then NU.1.1 does not apply, so for all $\mathit{prec} \in CB$ that contain f and all pro-f factors $q \in \mathit{prec} \models q$ it holds that $q \in \mathit{curr}$ so $CB, \mathit{curr} \models q$. But then condition (2a) of Definition 2.2 is satisfied for all such q. Moreover, NU.1.2 does not apply so there is no con-f factor q such that there is an argument for $q \in \mathit{Outcomes}(\mathit{curr})$. But then there is no precedent that contains q, so condition (2b) of Definition 2.2 is satisfied since its antecedent is satisfied, since without any precedent that contains q, it cannot be that $CB, F \models q$.

(ii) If A has a defeater B with a top rule applying NU1.1 for $q \in \mathtt{Pro}(f)$, then B has a defeater C in S with a top rule applying NU1.1.1. Then there is a justified argument for $q \in \mathit{Outcomes}(\mathit{curr})$, so by the induction hypothesis $CB, \mathit{curr} \models q$. But then condition (2a) of Definition 2.2 is satisfied for q.

(iii) If A has a defeater B with a top rule applying NU1.2 where $q \in \mathtt{Con}(f)$, then P replies in S with either an argument applying NU.1.2.1 or with an undercutter C of O's subargument for $q \in \mathit{Outcomes}(\mathit{curr})$. In the first case $q \in \mathit{Prec}$, so condition (2b) of Definition 2.2 is satisfied for q. In the second case, if $CB, \mathit{curr} \models q$. Then $q \in \mathit{prec}$ but then NU1.2.1 applies, while we assumed it does not. So $CB, \mathit{curr} \not\models q$. But then condition (2b) of Definition 2.2 is satisfied for q.

Since (i), (ii) and (iii) cover all q to which condition (2) of Definition 2.2 can be applied, this condition is satisfied, so $CB, \mathit{curr} \models f$. □

Theorem 5.3. If $CB, \mathit{curr} \models f$ then here exists a justified argument for $f \in \mathit{Outcomes}(\mathit{curr})$.

Proof. The proof is with induction on the structure of the factor hierarchy.

Base case: $\mathit{curr} \models f$. Then there is a justified argument for $f \in \mathit{Outcomes}(\mathit{curr})$ that applies $\mathcal{K}$ only.

Induction hypothesis: For all f and all $q \in \mathtt{Pro}(f)$: if $CB, \mathit{curr} \models q$, then there is a justified argument for $f \in \mathit{Outcomes}(\mathit{curr})$.

Induction step: Suppose $CB, curr \models f$ but $f \notin F$. Then there exists a $prec \in CB$ such that $prec \models f$ and conditions (2a) and (2b) of Definition 2.2 are satisfied. Since $prec \models f$, P can move an argument A that applies NCS1.

Suppose A has a defeater B applying NU1.1. Then $prec \models q$, so by condition (2a) of Definition 2.2 it holds that $CB, curr \models q$. Then by the induction hypothesis there is a justified argument for $q \in Outcomes(curr)$. Then P can move a defeater C of B using NU1.1.1 and has a winning strategy for C.

Suppose A has a defeater B applying NU1.2.

If $CB, curr \models q$, then $q \in prec$ so P can move a defeater C of B that applies NU1.2.1, which has no defeaters.

If $CB, curr \not\models q$, then by soundness there is no justified argument for $q \in Outcomes(curr)$. But since AF is acyclic (Proposition 5.1) so every argument is either justified or overruled, the subargument for B's second premise is overruled so it has a justified defeater C, so P can win about C.

We have shown that P can defend A against any defeater with a justified argument, so P has a winning strategy for A. □

Finally, note that since any AF generated by the $ASPIC^+$ model is acyclic, all standard semantics for AF's are equivalent, so the results also hold for the other semantics defined by [5].

6. Conclusion

This paper has made the following contributions:

- The argument-scheme model of [14] in $ASPIC^+$ has been shown to be unsound and incomplete for the factor-based hierarchical result model of [16].
- A new argument-scheme model in $ASPIC^+$ has been presented and shown to be sound and complete for the full hierarchical model.

To the best of my knowledge, there is no other work on formally relating argument-scheme models with models of precedential constraint. The closest comes the grounded argument game of [12] for explaining outcomes of black-box classifiers. This game is motivated by a non-hierarchical model of precedential constraint with dimensions from [11] but proves no formal relations between the two.

Future research could investigate existing or new argument-scheme models for other models of precedential constraint, such as schemes for reason models [4], schemes for models with dimensions, such as [2] for [15], or value-based scheme models, such as [3]. The latter may require the development of a new model of precedential constraint including values.

References

[1] V. Aleven. Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment. *Artificial Intelligence*, 150:183–237, 2003.

[2] K.D. Atkinson, T.J.M. Bench-Capon, H. Prakken, and A.Z. Wyner. Argumentation schemes for reasoning about factors with dimensions. In K.D. Ashley, editor, *Legal Knowledge and Information Systems. JURIX 2013: The Twenty-sixth Annual Conference*, pages 39–48. IOS Press, Amsterdam etc., 2013.

[3] T.J.M. Bench-Capon, H. Prakken, A.Z. Wyner, and K. Atkinson. Argument schemes for reasoning with legal cases using values. In *Proceedings of the 14th International Conference on Artificial Intelligence and Law*, pages 13–22, New York, 2013. ACM Press.

[4] I. Canavotto and J. Horty. Reasoning with hierarchies of open-textured predicates. In *Proceedings of the 19th International Conference on Artificial Intelligence and Law*, pages 52–61, New York, 2023. ACM Press.

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

[6] W. Fungwacharakorn, M.M. Zin, H.T. Nguyen, Y. Kong, and K. Satoh. Argumentative reasoning with language models on non-factorized case bases, 2025. arXiv:2512.12656.

[7] J. Horty. Rules and reasons in the theory of precedent. *Legal Theory*, 17:1–33, 2011.

[8] S. Modgil and H. Prakken. Abstract rule-based argumentation. In P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, editors, *Handbook of Formal Argumentation*, volume 1, pages 286–361. College Publications, London, 2018.

[9] H. Prakken. Dialectical proof theory for defeasible argumentation with defeasible priorities (preliminary report). In J.-J.Ch. Meyer and P.-Y. Schobbens, editors, *Formal Models of Agents*, number 1760 in Springer Lecture Notes in AI, pages 202–215, Berlin, 1999. Springer Verlag.

[10] H. Prakken. An argumentation-based analysis of the Simonshaven case. *Topics in Cognitive Science*, 4:1068–109, 2020.

[11] H. Prakken. A formal analysis of some factor- and precedent-based accounts of precedential constraint. *Artificial Intelligence and Law*, 29:559–585, 2021.

[12] H. Prakken and R. Ratsma. A top-level model of case-based argumentation for explanation: Formalisation and experiments. *Argument and Computation*, 13:159–194, 2022.

[13] H. Prakken and W. van Woerkom. Defending the hierarchical result models of precedential constraint. In R. Markovich, L. Di Caro, A. Rapp, and C. Schifanella, editors, *Legal Knowledge and Information Systems. JURIX 2025: The Thirty-Eighth Annual Conference*, pages 344–349. IOS Press, Amsterdam etc., 2025.

[14] H. Prakken, A.Z. Wyner, T.J.M. Bench-Capon, and K. Atkinson. A formalisation of argumentation schemes for legal case-based reasoning in ASPIC+. *Journal of Logic and Computation*, 25:1141–1166, 2015.

[15] W. van Woerkom, D. Grossi, H. Prakken, and B. Verheij. Hierarchical a fortiori reasoning with dimensions. In G. Sileno, J. Spanakis, and G. van Dijck, editors, *Legal Knowledge and Information Systems. JURIX 2023: The Thirty-Sixth Annual Conference*, pages 43–52. IOS Press, Amsterdam etc., 2023.

[16] W. van Woerkom, D. Grossi, H. Prakken, and B. Verheij. Hierarchical precedential constraint. In *Proceedings of the 19th International Conference on Artificial Intelligence and Law*, pages 333–342, New York, 2023. ACM Press.

[17] K. Čyras, K. Satoh, and F. Toni. Abstract argumentation for case-based reasoning. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference*, pages 549–552. AAAI Press, 2016.