Hierarchical Precedential Constraint

Wijnand van Woerkom
Utrecht University
Department of Information and Computing Sciences
Utrecht, The Netherlands
w.k.vanwoerkom@uu.nl

Davide Grossi
University of Groningen
Bernoulli Institute for Math, CS and AI
Groningen, The Netherlands
University of Amsterdam
Amsterdam Center for Law and Economics
Institute for Logic, Language and Computation
Amsterdam, The Netherlands
d.grossi@rug.nl

Henry Prakken
Utrecht University
Department of Information and Computing Sciences
Utrecht, The Netherlands
University of Groningen
Faculty of Law
Groningen, The Netherlands
h.prakken@uu.nl

Bart Verheij
University of Groningen
Bernoulli Institute for Math, CS and AI
Groningen, The Netherlands
bart.verheij@rug.nl

ABSTRACT

In recent work, theories of case-based legal reasoning have been applied to the development of explainable artificial intelligence methods, through the analogy of training examples as previously decided cases. One such theory is that of precedential constraint. A downside of this theory with respect to this application is that it performs single-step reasoning, moving directly from the case base to an outcome. For this reason we propose a generalization of the theory of precedential constraint which allows multi-step reasoning, moving from the case base through a series of intermediate legal concepts before arriving at an outcome. Our generalization revolves around the notion of factor hierarchy, so we call this hierarchical precedential constraint. We present the theory, demonstrate its applicability to case-based legal reasoning, and perform a preliminary analysis of its theoretical properties.

CCS CONCEPTS

• **Theory of computation** → *Automated reasoning*; • **Information systems** → *Expert systems*; • **Computing methodologies** → *Knowledge representation and reasoning*.

KEYWORDS

case-based reasoning, precedential constraint, factors, factor hierarchy, explainable artificial intelligence

ICAIL 2023, June 19–23, 2023, Braga, Portugal

ACM ISBN 979-8-4007-0197-9/23/06…$15.00
https://doi.org/10.1145/3594536.3595154

ACM Reference Format:
Wijnand van Woerkom, Davide Grossi, Henry Prakken, and Bart Verheij. 2023. Hierarchical Precedential Constraint. In *Nineteenth International Conference on Artificial Intelligence and Law (ICAIL 2023), June 19–23, 2023, Braga, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3594536.3595154

1 INTRODUCTION

There has been a lot of interest over the past years in making artificial intelligence (AI) more interpretable or explainable, as this kind of technology is increasingly being applied to tasks with ethical, social, or legal impact to end-users. This emerging field is often called explainable artificial intelligence (XAI).

It has been argued in the literature, e.g. in [19], that one approach to tackle this problem is by considering the problems and solutions studied in the field of AI & law, of which explainability has always been a core aspect. Indeed, we have seen several XAI methods developed with their roots in argumentation and case-based reasoning, such as [14, 6, 7, 8].

The recently developed method in [14] makes use of the formal result model of *precedential constraint* first proposed by Horty in [11], as the basis of its case-based reasoning mechanism. Examples from the AI's training data are represented in a factor-based approach in the style of HYPO [3], and explanations for specific decisions made by the AI are provided in the form of an argumentative dialogue about the decision at hand, starting with the citation of a similar case from the training set.

In this model of precedential constraint a simplifying assumption is made that the precedential reasoning involved moves directly from the precedent to an outcome in the focus case, rather than moving in multiple steps through intermediate legal concepts. This limits the argumentative discourse that can be based on the theory, which in turn limits the depth of the explanation that a method such as the one developed in [14] can produce. To this end, in the present work we propose a generalization of Horty's theory of precedential constraint which enables multi-step precedential reasoning, primarily in the hope that it can form the basis of more expressive XAI methods. Central to this generalization is a move from a plain factor-based representation of cases to the use of a *factor hierarchy* as used in CATO.

The HYPO program starts with a simple representation of cases as consisting of factors that favor either the plaintiff or the defendant. In its successor, CATO, a factor hierarchy is used instead, in which factors do not simply express a preference towards either of the outcomes but instead support or oppose each other in a hierarchical fashion. Just as CATO allowed multi-step inferences through the use of a factor hierarchy compared to HYPO, we wish to facilitate multi-step inferences for precedential constraint through the use of a factor hierarchy. This leads us to name our generalization *hierarchical precedential constraint*.

Another source of inspiration for us is the work on precedential constraint by Roth and Verheij in [15, 16]. In [15] Roth presents a model which is similar in some ways to that of Horty, in that it contains a comparable definition of what it means for a case base to constrain future decision making; and different in other ways, for instance in that it represents cases in a style that is closer to the hierarchical representation used in the CATO program.

After having presented our generalization, we demonstrate its behaviour through some examples, and we analyse its theoretical properties. We do so in particular by considering whether the form of reasoning thus described is *monotonic* or not; a property which characterizes the defeasability of inferences and plays an important role in the study of reasoning and argumentation [17].

This work is structured as follows. We begin by discussing some related work in Section 2. Then, in Section 3, we describe the theory of precedential constraint of [11]. In particular, we describe the knowledge representation it uses in Section 3.1, the theory itself in Section 3.2, some examples in Section 3.3, and a discussion on the topic of monotonicity in Section 3.4. In Section 4 we then present our generalization, following the same layout as Section 3; in Section 4.1 we present a knowledge representation in the style of the CATO factor hierarchy, in Section 4.2 we describe the notion of constraint for cases in this representation, in Section 4.3 we give some examples, and in Section 4.4 we consider whether this form of reasoning is monotonic. We conclude with a discussion and conclusion in Sections 5 and 6 respectively.

2 RELATED WORK

In the final two paragraphs of [11] Section 7, Horty describes a possible generalization of his theory which allows multi-step reasoning. In this modified version cases would be represented as linked sets of *precedent constituents*, moving from the initial facts of the case through a series of more abstract legal concepts to ultimately arrive at a decision for the plaintiff or the defendant.

A work that has attempted to develop such a more general framework, and which actually predates Horty's [11], is found in Roth's dissertation [15] Chapter 3 and Roth and Verheij's [16]. In this work a formal theory is formulated which aims to describe the conclusions that follow from the precedential reasoning method. Roth's theory notably differs from that of Horty in that cases are not represented as sets of factors but as sentences in a language of [20] with symbols for support and attack between these factors. This way a case cannot only contain factors but can also contain information on the internal relation between the factors in terms of attack and support, inspired by the idea of the factor hierarchy of [1]. A further difference, inspired by [20], is that sentences of the form $p \to (q \to r)$ are also permitted, so that a factor can express support for a support relation, and so on.

Based on these richer case representations a theory is then developed on the way in which precedent cases influence decisions in a novel fact situation. Interestingly, Roth uses attack and support relations to allow the a fortiori reasoning it models to be applied not only to the final outcome of the case but also to intermediate conclusions. This is done by using the attack and support relations to identify components of the case that provide support or opposition to the claim at hand; if there is more support in the novel case but less opposition, then the same decision should be made in the novel case. This is very much akin to the ideas in Horty's work [11]. In fact, some of the essence of Horty's formalization of a fortiori constraint through [11] Definition 10 was already present in Roth's notion of *dialectical support* in [15] Definition 8 and [16] Definition 5, and with some effort one can show that the former definition can be stated in terms of the latter.

In the present work we opt to build directly on Horty's theory rather than follow that of Roth. This has the advantage that we can make use of the theoretical and applied results that followed [9], which has been studied more extensively than Roth's theory. In addition, we do not model Roth's nesting of support and attack relations, and instead focus on the hierarchy of factors.

However, we do take inspiration from Roth's theory. Firstly, we follow Roth by enriching the representation of cases relative to Horty's theory. In contrast to Roth, we do not do so directly, by allowing case representations to contain sentences expressing support or opposition between factors; but indirectly, by keeping the case representation as in Horty's theory, but assuming that the set of factors has an underlying hierarchical structure. Secondly, we follow Roth in allowing the forcing relation to operate not just on the final outcome of cases (i.e. a decision for the plaintiff or the defendant) but also on intermediate decisions. However, in contrast, we do so not by use of what Roth calls a *relevance* relation, but rather by applying recursion on the structure of the factor hierarchy.

3 PRECEDENTIAL CONSTRAINT

In this section we describe Horty's formal account of the result model of precedential constraint, first presented in [11]. We begin by describing the factor-based knowledge representation it uses, pioneered by the HYPO program. We then give Horty's definition of the way in which a case base in this representation constrains future decision making, and demonstrate it through some examples. We conclude with some observations on whether the inferences resulting from this notion of constraint are monotonic.

3.1 Factors, sides, and cases

A concept that permeates the AI & law literature is that of *factors*; facts which are legally relevant in the sense that their presence or absence influence the outcome of cases. A factor is traditionally assumed to provide support for exactly one of the outcomes. This support is defeasible rather than strict, and the weighing of factors against each other is a key component of court cases. Factors were first used as a knowledge representation device in the HYPO program [3], and this use has since become common practice in the field of AI & law. More information about factors and their history may be found in e.g. [3].

Formally the factors of some legal domain are modelled as a finite set P of propositional variables. Additionally, we assume there are two possible *outcomes*; π, for plaintiff; and δ, for defendant. These outcomes may also be referred to as sides, and we write $\mathcal{S} := \{\pi, \delta\}$ for the set of sides. Each factor is assumed to support exactly one of the two outcomes. This is modelled by the assumption that P is equal to a disjoint union $P = \text{Pro} \cup \text{Con}$. If $p \in \text{Pro}$ then p is a factor that provides support for a decision in favor of the plaintiff; if instead $p \in \text{Con}$ then it provides opposition to a decision in favor of the plaintiff. When a factor is pro-π it is assumed to be con-δ and vice versa, so for a uniform treatment of the sides π and δ it is useful to define two functions $\text{Pro}, \text{Con} : \mathcal{S} \longrightarrow 2^P$ by

$$Pro(s) := \begin{cases} \text{Pro} & \text{if } s = \pi, \\ \text{Con} & \text{if } s = \delta, \end{cases} \quad \text{Con}(s) := \begin{cases} \text{Con} & \text{if } s = \pi, \\ \text{Pro} & \text{if } s = \delta, \end{cases} \tag{1}$$

where 2^P denotes the powerset of P.

A *fact situation* is now a *valuation*, or also sometimes called an *interpretation*, of the set of factors P. This means that a fact situation F is a function $F : P \longrightarrow \{\mathbf{t}, \mathbf{f}\}$ that maps the factors to either true ($\mathbf{t}$) or false ($\mathbf{f}$). Intuitively $F(p) = \mathbf{t}$ means the factor p applies in F, and $F(p) = \mathbf{f}$ means the factor p does not apply in F. Instead of writing $F(p) = \mathbf{t}$ we will write $F \vDash p$, and similarly $F \vDash \neg p$ instead of $F(p) = \mathbf{f}$. If $G \subseteq P$ then $F \vDash G$ means $F \vDash p$ for all $p \in P$. A *case* is a fact situation that is decided for one of the two outcomes, modelled as a pair (F, s) with F a fact situation and $s \in \mathcal{S}$ a side. Such a case will be denoted $F{:}s$ for brevity. A set of cases is called a *case base* and usually denoted by CB.

We note that in [11] fact situations are represented as subsets $F \subseteq P$, but formally speaking this is no different from using valuations $F : P \longrightarrow \{\mathbf{t}, \mathbf{f}\}$ (simply regard those factors present in the fact situations as being assigned true, and those absent as being assigned false). We prefer to use valuations as they are more readily generalized to *partial* fact situations, which are not used in the theory of [11], but which will be useful to us in the present work. By partial fact situation we mean a fact situations in which not all factors are true or false, but a third ‘unknown’ or ‘undecided’ option is possible. We will talk about such fact situations in the language of *three-valued* logic, which has a rich history of study; see e.g. [9]. In the present work we follow the notation and conventions of [4].

Formally we can view such a partial fact situation as a function $F : P \longrightarrow \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. Intuitively we interpret $F(p) = \mathbf{u}$ to mean that it is not (yet) decided whether p applies in F or not. We will use the same notation for partial fact situations as for regular fact situations with respect to the entailment symbol $\vDash$.

Example 3.1. Both HYPO and CATO were applied to the domain of trade secret law, and so we will use this domain as a running example in this work. In Figure 1 some example factors are shown for this domain, they are described in [1] Appendix 1 as follows.

F1 Plaintiff disclosed its product information in negotiations with defendant.

F2 Defendant paid plaintiff’s former employee to switch employment, apparently in an attempt to induce the employee to bring plaintiff’s information.

F4 Defendant entered into a nondisclosure agreement with plaintiff.

F6 Plaintiff adopted security measures.

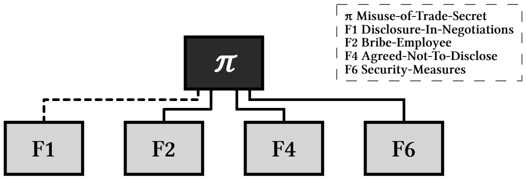

Figure 1: An example of some factors for the domain of trade secret law, from [2] Figure 4. The lines indicate whether the factors support or oppose a decision for the plaintiff; solid lines indicate support and dashed lines indicate opposition.

All aside from F1 are pro-plaintiff factors, meaning that their presence offers defeasible support for a decision for the plaintiff.

3.2 The notion of constraint

Horty’s theory of precedential constraint, first proposed in [11], builds on the factor-based method of case representation to give a precise account of the way in which precedent constrains future decision making. We recall it here, but should note that our presentation differs slightly from the way it is usually done in the literature, so as to emphasize similarities with the generalized version we propose in Section 4.

The view of precedent adopted in Horty’s theory is that the court’s decision in the precedent case represents an assessment of the balance of the applying pro and con factors. Later courts are then constrained to make decisions that are consistent with this assessment. What it means to be consistent is made precise by the theory: if a certain configuration of pro and con factors were decided for a particular outcome, then any fact situation with the same or more pro factors, and the same or fewer con factors, should be decided for that same outcome. This is captured by the following definition, which is [11] Definition 10.

Definition 3.2. Let CB be a case base for a set of factors P, and $s \in \mathcal{S}$ a side. We say *CB forces* the decision of a fact situation F for s, written $CB, F \vDash s$, if and only if there exists $G{:}s \in CB$ such that:

(1) for all $q \in \text{Pro}(s)$: if $G \vDash q$ then $F \vDash q$, and
(2) for all $q \in \text{Con}(s)$: if $F \vDash q$ then $G \vDash q$.

Remark 3.3. The notation we use here for forcing is copied from the one used for the notion of semantic consequence in logic. When a formula ϕ entails another formula ψ this is written as $\phi \vDash \psi$, and if this relation only holds in the presence of an ambient set of formulas Γ then this is denoted by $\Gamma, \phi \vDash \psi$. In our setting the role of Γ is filled by the case base CB, as the presence of these cases may allow us to derive additional conclusions for a given fact situation.

The primary aim of the present work is to define an appropriate version of Definition 3.2 for a variation of this model where the factors form a hierarchy, and this attempt culminates in Definition 4.8.

Before proceeding we look at some examples, and consider whether this form of reasoning is monotonic or not.

3.3 Examples

We briefly illustrate the workings of the forcing relation through two examples, based on the model depicted in Figure 1.

Example 3.4. Let the set of factors P be given by the two sets Pro $= \{\text{F2}, \text{F4}, \text{F6}\}$ and Con $= \{\text{F1}\}$. Now suppose that the fact situation $G \vDash \{\text{F1}, \text{F2}, \text{F4}, \neg\text{F6}\}$ was decided in favor of the plaintiff, meaning it was decided that the pro-π factors F2 and F4 together outweigh the con-π factor F1. Now for any fact situation F we have

$$\begin{aligned} &\{G{:}\pi\}, F \vDash \pi \\ &\text{iff (1) for all } q \in \{\text{F2}, \text{F4}, \text{F6}\}\text{: if } G \vDash q \text{ then } F \vDash q, \text{ and} \\ &\quad\ \text{(2) for all } q \in \{\text{F1}\}\text{: if } F \vDash q \text{ then } G \vDash q \\ &\text{iff } F \vDash \{\text{F2}, \text{F4}\}. \end{aligned}$$

Note that clause (2) is trivially satisfied here because the consequent of the implication, $G \vDash \text{F1}$, is true; it does not matter whether F satisfies F1 or not. So, in the presence of the case $G{:}\pi$ any fact situation F to which at least F2 and F4 apply will have its decision for π forced.

Example 3.5 (Inconsistency). Note that nothing prevents *both* sides π and δ from being forced simultaneously for a particular fact situation F. In [11] a case base within which such a disagreement occurs is said to be *inconsistent*. For example, consider the case $H{:}\delta$ with fact situation $H \vDash \{\text{F1}, \text{F2}, \text{F4}, \text{F6}\}$. Then, with G as in Example 3.4, we have $\{G{:}\pi, H{:}\delta\}, G \vDash \delta$; despite the fact that G was decided for π in the case base $\{G{:}\pi, H{:}\delta\}$.

3.4 Monotonicity

It was remarked by the authors of [12] that this theory of precedential constraint is essentially monotonic. In fact, there are two ways in which this type of precedential reasoning could be monotonic: with respect to the case base or with respect to the fact situation. It is easy to see that the theory is monotonic in the case base, because the addition of more cases to a case base does not obstruct a pre-existing precedent case from acting as a witness to Definition 3.2. This is formalized by the following proposition, which – to the best of our knowledge – does not appear elsewhere in the literature.

Proposition 3.6. *For case bases CB and CB' with $CB \subseteq CB'$, a fact situation F and a side s, if $CB, F \vDash s$ then $CB', F \vDash s$.*

Proof. By the assumption $CB, F \vDash s$ there is $G{:}s \in CB$ satisfying (1) and (2) of Definition 3.2, and so since $CB \subseteq CB'$ we have that $G{:}s \in CB'$ and therefore $CB', F \vDash s$. □

What about the second kind of monotonicity, with respect to the fact situation under consideration? In the theory as we have just described it this question cannot be phrased since a fact situation is defined as assigning true or false to all factors, and so there is no way in which additional information can be added to a fact situation. Note that the same applies to the representation of fact situations as subsets $F \subseteq P$ used in [11]; adding a factor $p \in P$ to such a subset can change a decision made about p. Consider, for instance, the factor F6 from the factors of Figure 1. If it was decided that precautions taken by the plaintiff were insufficient to constitute security measures, so that $\text{F6} \notin F$, then adding F6 to F would not add an assumption but modify an existing assumption.

This changes when we allow fact situations to be partial in the way described in Section 3.1, and apply Definition 3.2 to partial fact situations just as we would to regular fact situations, as information can be added to a partial fact situation by changing the truth value from factors assigned ‘undecided’ to true or false instead. This can be expressed formally as follows. We order the values $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ by the *information order* $<_i$ as in [4]: $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$, while $\mathbf{t}$ and $\mathbf{f}$ are incomparable. The reflexive closure of $<_i$ is denoted by $\leq_i$, which is extended to partial fact situations F, G by $F \leq_i G$ iff $F(p) \leq_i G(p)$ for all $p \in P$. The relation $F \leq_i G$ indicates that if $F(p) \in \{\mathbf{t}, \mathbf{f}\}$ then $F(p) = G(p)$. We now have the following proposition.

Proposition 3.7. *For a case base CB, if F and G are partial fact situations such that $F \leq_i G$, then $CB, F \vDash s$ does not imply $CB, G \vDash s$.*

Proof. For a counter example, consider again the set of factors depicted in Figure 1. Let $H{:}\pi$ be a case of which the fact situation H assigns true only to F2, and false to the other factors. Now let F be a partial fact situation assigning true to F2 and undecided to the other factors. Applying Definition 3.2 we have $\{H{:}\pi\}, F \vDash \pi$ since $F \vDash \text{F2}$ true and $F \nvDash \text{F1}$. Now, let G be the partial fact situation assigning true to F2 and F1 and unknown to the other factors. Then $F \leq_i G$, but $\{H{:}\pi\}, G \nvDash \pi$ since $G \vDash \text{F1}$ while $H \vDash \neg\text{F1}$. □

This proposition shows that, for partial fact situations, we should think of the forcing relation as tentative, and its conclusions as contingent on what has been decided thus far.

4 HIERARCHICAL PRECEDENTIAL CONSTRAINT

An important way in which the CATO program innovated over its predecessor HYPO was through the introduction of hierarchical structure on the set of factors on which it operates. We wish to enrich the representation of factors laid out in Section 3.1 in a similar way and, in doing so, indirectly enrich our case representations.

This section follows the same structure as Section 3. We begin by describing the factor hierarchy used in the CATO program, and then give a formal definition of what a factor hierarchy is composed of in general. For more information about CATO and its factor hierarchy the reader is referred to e.g. [1] or [15] Chapter 4. We then define what it means for a set of cases in this representation to constrain future decision making, demonstrate it by looking at some examples, and then consider whether this form of reasoning is monotonic.

4.1 Factor hierarchy

Like HYPO the CATO program assumes the existence of a set of factors that each provide support for exactly one of the two case outcomes. However CATO further assumes there is an additional case-independent structure on the set of factors which details the way in which the factors relate to each other, in addition to the way in which they relate to the case outcomes. This structure, together with the set of factors, is called the *factor hierarchy*.

Example 4.1. The prototypical example of a factor hierarchy is the one used in the CATO program for the trade-secret domain. It was constructed through knowledge engineering, and a portion of it is depicted in Figure 2. Note that since the hierarchy in Figure 2 depicts only a part of the complete CATO factor hierarchy it is not necessarily the case that what is indicated here as a basic factor is also a basic factor in the original CATO hierarchy.

The links in the hierarchy indicate a generic relation of support, and can be either positive or negative depending on whether the linked factors support the same outcome or not.[1] Factors become increasingly abstract the further they are up in the hierarchy, up to those in the penultimate layer which in [1] are called *legal issues*. The factors that are lowest in the hierarchy are called *basic* or *base-level* factors, and are assumed to be determined directly by the plain facts of the case. The factors above the base-level factors are called *abstract* factors. A positive link indicates (defeasible) support for a more abstract factor, and a negative link indicates (defeasible) opposition. Lastly, links can be strong or weak, indicating their level of support, which is not visualized in Figure 2.

Before we formalize this hierarchical structure we note that we do not attempt to replicate it as closely as possible. For instance, we will not model that links can be weak or strong. We do this for the sake of simplicity and because this distinction does not seem to be directly relevant for our purposes; we simply let the precedent fully decide what the relative strength of the support links is. Furthermore, we drop the assumption that all factors have a preference towards one of the two case outcomes. Instead, we assume only the presence of a hierarchical structure indicating the preference that factors have for each other. This relaxation allows for more structure in the hierarchy, although it remains to be seen whether this offers any benefit in practice. Lastly, we add a layer on top of the legal issues indicating support for either of the case outcomes, so that the hierarchy is a single entity rather than consisting of separate components as in the CATO hierarchy.

To formally define the factor hierarchy, we assume as before that the factors are given by a finite set P of propositional letters. However, as mentioned, we now follow Roth's practice and Horty's suggestion to regard the outcomes π and δ as special factors. This change is not intended to degrade the status of the case outcomes to mere factors, but rather to elevate the status of factors as themselves being subject to decision making. Additionally, we do not assume that P is equal to a disjoint union, meaning that we do not assume that all factors favor exactly one of the two case outcomes. The hierarchical structure on P is given by a relation H on P, where $H(p, q)$ indicates that p is (directly) below q in the hierarchy.

The question now is what further structure we should impose on H. We could, for instance, demand that H (regarded as a graph) is a tree, meaning that between each factor there should be exactly one path to any other factor through the hierarchy. However, looking at Figure 2 we see that it violates this assumption; from F4 there are two paths up to F114, one through F115 and one through F121. To our knowledge there is no exact specification given in the literature on what shape a factor hierarchy should have in general, so we suggest such a specification here.

While the condition that H is a tree is too strict, we would still expect it to have some tree-like properties. For example, the hierarchy should connect all the factors within it rather than consist of separate disconnected components. It should also not be possible for a factor to be above itself in the hierarchy, which is to say the hierarchy should not contain cycles.

Furthermore, the hierarchy should culminate in a single factor representing the binary outcome of the case. The choice for the side corresponding to this factor is arbitrary, in what follows we choose to associate it with the plaintiff. The other side is then represented as the negation of this factor, in our case the defendant.

Lastly, the links between factors in the hierarchy should each have a polarity, indicating whether the link is expressing support (positive) or opposition (negative).

Grouping these conditions yields the following definition, which is foundational to our notion of hierarchical precedential constraint.

Definition 4.2. A *factor hierarchy* is a tuple (P, H) with P a finite set of propositional letters 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 $\mathrm{Pro} \cup \mathrm{Con}$.

Some remarks on the notation and terminology used here in Definition 4.2 are in order.

(1) The transitive closure R^+ of a relation R is the smallest extension of R which satisfies the property that if $R^+(x, y)$ and $R^+(y, z)$ then $R^+(x, z)$. Irreflexivity of R^+ means that $R^+(x, x)$ does not hold for any x, so if R is defined on a finite set then this property is a succinct formal way of saying that R does not contain cycles.
(2) An element x is said to be R-maximal if for no element y the relation $R(x, y)$ holds; in other words there are no elements above x in the relation. In general a set can have more than one maximal element, or none at all, so we require hierarchies to have exactly one. This condition also ensures that H, considered as a graph, is connected.
(3) The edges in the hierarchy must denote either a support or an opposition relation, but not both. This is expressed succinctly by requiring that the relation H (which is formally a set of ordered pairs) is equal to a disjoint union $\mathrm{Pro} \cup \mathrm{Con}$. Note that these are not the sets of propositions as in the factor-based representation of Section 3.1: in this context Pro and Con are both relations which together make up H and do not overlap. We suggestively name them Pro and Con to highlight the very similar purpose they serve compared to the eponymous sets of Section 3.1. So, concretely, if for two factors $p, q \in P$ the relation $H(p, q)$ holds then either $\mathrm{Pro}(p, q)$, meaning p is a pro-q factor; or $\mathrm{Con}(p, q)$, meaning p is a con-q factor; but not both.

Despite grouping all factors into a single set P we can still differentiate them in the usual language of case-based reasoning according to their role in the hierarchy H.

Definition 4.3. A factor $p \in P$ is *basic* if it is H-minimal, meaning there is no $q \in P$ with $H(q, p)$. We write B for the set of basic factors. Factors that are not basic are called *abstract*. We write A for the set of abstract factors, so $A := P \setminus B$.

[1] There is one exception to this rule in [2] Figure 4 in the link between F15 and F111 which are both pro-π factors but have a negative link. As this was likely a typo the link is made positive in Figure 2.

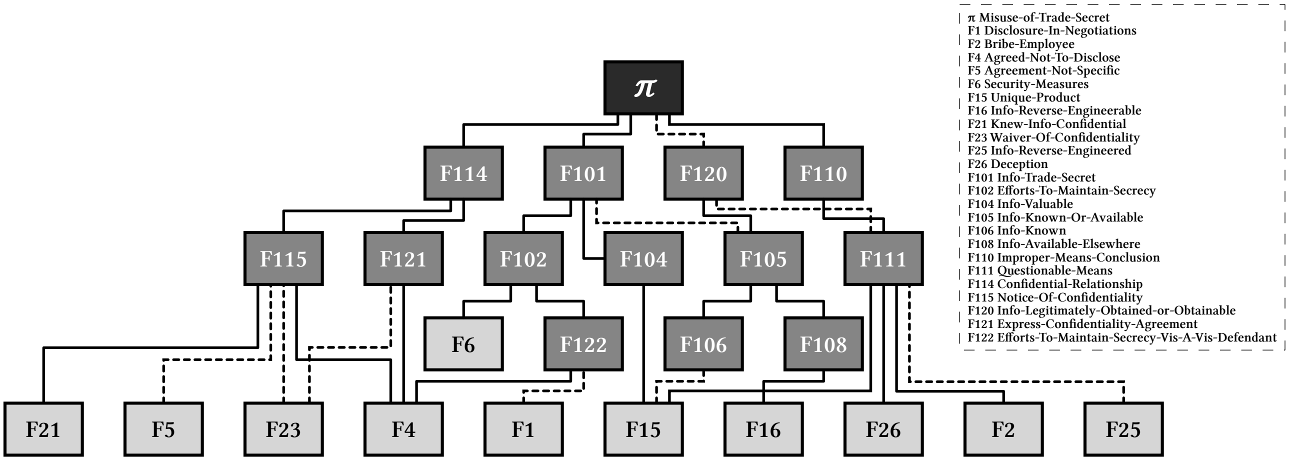
π Misuse-of-Trade-Secret
F1 Disclosure-In-Negotiations
F2 Bribe-Employee
F4 Agreed-Not-To-Disclose
F5 Agreement-Not-Specific
F6 Security-Measures
F15 Unique-Product
F16 Info-Reverse-Engineerable
F21 Knew-Info-Confidential
F23 Waiver-Of-Confidentiality
F25 Info-Reverse-Engineered
F26 Deception
F101 Info-Trade-Secret
F102 Efforts-To-Maintain-Secrecy
F104 Info-Valuable
F105 Info-Known-Or-Available
F106 Info-Known
F108 Info-Available-Elsewhere
F110 Improper-Means-Conclusion
F111 Questionable-Means
F114 Confidential-Relationship
F115 Notice-Of-Confidentiality
F120 Info-Legitimately-Obtained-or-Obtainable
F121 Express-Confidentiality-Agreement
F122 Efforts-To-Maintain-Secrecy-Vis-A-Vis-Defendant
π
F114
F101
F120
F110
F115
F121
F102
F104
F105
F111
F6
F122
F106
F108
F21
F5
F23
F4
F1
F15
F16
F26
F2
F25

Figure 2: A part of the factor hierarchy on trade secret misappropriation, repeated from [2] Figure 4 but with a final layer added expressing support or opposition to the outcome. The basic factors are shown in light gray, the abstract factors in gray, and the outcome in black. Support and opposition edges are indicated by solid and dashed lines respectively.

Definition 4.4. By assumption there is exactly one H-maximal element, which we denote by π. The factor π means the case was decided in favor of the plaintiff.

Remark 4.5. Note that there is no factor directly corresponding to a decision in favor of the defendant. This role will be filled by the negation of π. In other words, a fact situation F is decided for the defendant if it assigns false to π, which is denoted by $F \vDash \neg\pi$.

Now we can define (partial) fact situations and cases just as we did in Section 3.1; a partial fact situation F is a function $F : P \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$, and a case is a partial fact situation F which is defined on π, i.e. such that $F(\pi) \in \{\mathbf{t}, \mathbf{f}\}$. Again we write $F \vDash p$ instead of $F(p) = \mathbf{t}$, $F \vDash \neg p$ instead of $F(p) = \mathbf{f}$, and now $F \vDash ?p$ instead of $F(p) = \mathbf{u}$. If $G \subseteq P$ we write $F \vDash G$ to mean $F \vDash p$ for all $p \in G$. The relation $F \vDash p$ should be understood as saying F was *decided* for p.

Remark 4.6. Notice the difference between $F \vDash \neg p$ and $F \nvDash p$; these statements are only equivalent for a fact situation F that does not assign $\mathbf{u}$ to any factor, i.e. when $F : P \to \{\mathbf{t}, \mathbf{f}\}$.

In what follows we focus on partial fact situations, which we may refer to as just fact situations. This is because we will now assume that precedent may need to be consulted in order to determine whether a factor should apply in the focus situation. Just as how the forcing relation of Definition 3.2 applies to fact situations, which may be considered partial fact situations that lack an answer for the π factor, the relation of forcing in the hierarchical case should be applicable to fact situations that are missing a decision for factors that may be forced by the case base. Allowing fact situations to be partial in this way is common practice in AI & law, see e.g. [13] for a discussion and relevant literature on this topic.

Example 4.7. Let us consider how the factor hierarchy depicted in Figure 2 relates to Definition 4.2. We let P denote the set of factors

$$\{\pi, \text{F114}, \text{F120}, \text{F110}, \text{F115}, \text{F121}, \ldots, \text{F2}, \text{F25}\}.$$

The hierarchy H for this example is given by the lines in the figure, where the direction of the links is determined by vertical order. So for instance, $H(\text{F6}, \text{F102})$ and $H(\text{F23}, \text{F121})$. Each link is expressive of support or of opposition. So we have in particular that $\text{Pro}(\text{F6}, \text{F102})$, indicating that F6 is a pro-F120 factor; and $\text{Con}(\text{F23}, \text{F121})$, indicating that F23 is a con-F121 factor. A case F is now a partial fact situation $F : P \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ which assigns true or false to the outcome factor π, i.e. $F(\pi) \in \{\mathbf{t}, \mathbf{f}\}$. For instance, we may have $F \vDash \{\text{F101}, \text{F120}, ?\text{F110}, \pi\}$, indicating that it has been decided in the case F that F101 and F120 apply, while no decision has been made about whether F110 applies (leaving the decisions in F about the other factors unspecified). Intermediate decisions are modelled in the same way; if we have in addition that $F \vDash \{\text{F102}, \neg\text{F104}, \text{F105}\}$, then this means that it was decided in F that while F104 does not apply, the factor F102 does and outweighs the applying con-F101 factor F105, because the factor F101 does hold.

4.2 The notion of constraint

Having enriched the factor-based representation of cases with hierarchical structure as per CATO's factor hierarchy, we proceed in this section to define a notion of precedential constraint analogous to Definition 3.2, which takes into account this additional structure. In what follows we will frequently compare and contrast our approach to Horty's theory, outlined in Section 3; so to facilitate this discussion we will refer to Horty's theory by the abbreviation PC (for precedential constraint), and to our generalization by the abbreviation HPC (for hierarchical precedential constraint).

In principle we could now formulate a straightforward adaptation of Definition 3.2 to factors, by stating that the decision of a fact situation F for some proposition $p \in P$ is forced if there is some precedent G with $G \vDash p$ containing fewer or equal pro-p factors, as well as more or equal con-p factors. However, this approach has two shortcomings which we will now consider.

Firstly, this naive approach overlooks a subtlety introduced by the levelwise support and opposition that factors can have for each other. For instance, considering the fact situations F and G as before, it may be that F is missing a pro-p factor q which is true in G, but that there is a precedent case H which forces the decision of F for q. We will address this by defining forcing for HPC using recursion.

Secondly, in PC there is an implicit notion of *negation* for sides; if a case is not decided for the plaintiff then it must have been decided for the defendant, and vice versa. This negation operates by switching the roles of the Pro and Con sets. Therefore, to generalize this behaviour we should not just consider the factors in P but also their negations, so that the forcing relation operates not only on factors but also on their negations.

To this end we define $\mathcal{P} := P \cup \{\neg p \mid p \in P\}$. We let $\mathcal{B}$ and $\mathcal{A}$ respectively denote the closures of B and A under the negation operation, so e.g. $\mathcal{B} := B \cup \{\neg p \mid p \in B\}$. We extend the operation of a partial fact situation $F : P \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ to $\mathcal{P}$ in the evident way: $F(\neg p) := \neg F(p)$ where $\neg\mathbf{t} := \mathbf{f}$, $\neg\mathbf{f} := \mathbf{t}$ and $\neg\mathbf{u} := \mathbf{u}$ (again, in accordance with [4]). The negation of π is denoted by $\delta := \neg\pi$.

Finally, as in Eq. (1), we will make use of auxiliary functions $\mathrm{Pro}, \mathrm{Con} : \mathcal{P} \to 2^P$ mapping a factor to its sets of pro and con factors, given by its preimages under the Pro and Con relations respectively, and switching Pro and Con factors for negations:

$$\mathrm{Pro}(p) := \{q \mid \mathrm{Pro}(q, p)\}, \qquad \mathrm{Con}(p) := \{q \mid \mathrm{Con}(q, p)\},$$
$$\mathrm{Pro}(\neg p) := \mathrm{Con}(p), \qquad \mathrm{Con}(\neg p) := \mathrm{Pro}(p).$$

At last we have the required vocabulary to phrase a definition of forcing for HPC, analogous to Definition 3.2 of PC.

Definition 4.8. Let CB be a case base for a factor hierarchy (P, H), and $p \in \mathcal{P}$ a factor. We say *CB forces* the decision of a fact situation F for p, written $CB, F \vDash p$, if and only if either:

- $F \vDash p$, or
- $p \in \mathcal{A}$ and there is $G \in CB$ with $G \vDash p$ such that:
  (1) for all $q \in \mathrm{Pro}(p)$, if $G \vDash q$ then $CB, F \vDash q$, and
  (2) for all $q \in \mathrm{Con}(p)$, if $CB, F \vDash q$ then $G \vDash q$.

The base case of the recursion is that $F \vDash p$. Informally this says that a case base (trivially) forces the decision of F for p if F is already decided for p. The general case is only applied to abstract factors and requires the existence of a precedent case G which forces the decision of the focus case for p in the manner of Definition 3.2, but where a recursive call to $CB, F \vDash p$ takes the place of $F \vDash p$. Since $F \vDash p$ implies $CB, F \vDash p$ as per the base case, the general case can be understood as first expanding the decisions made in F with all possible decisions forced by the case base, before considering whether the existentially quantified G has the same or fewer pro-p factors and more or equal con-p factors.

That the recursion in Definition 4.8 ends – and so that the definition is valid – is ensured by the assumptions that P is finite and that H contains no cycles. Clause (2) applies only to abstract factors $p \in \mathcal{A}$, which prevents the case base from vacuously forcing basic factors: if $p \in \mathcal{B}$ then $\mathrm{Pro}(p) = \mathrm{Con}(p) = \emptyset$ by definition, so any $G \in CB$ with $G \vDash p$ would trivially satisfy conditions (1) and (2). The idea of basic factors is not that they are decided on the basis of precedent, but that they are determined directly by the plain facts brought to the court.

We conclude this section with some lemma's that follow trivially from Definition 4.8 but which are worth stating separately.

LEMMA 4.9. *For any $p \in \mathcal{P}$, if $F \vDash p$ then $CB, F \vDash p$.*

LEMMA 4.10. *For any $p \in \mathcal{B}$, if $CB, F \vDash p$ then $F \vDash p$.*

4.3 Examples

Let us now consider a series of examples to illustrate and justify the components of the model and the Definition 4.8 of constraint. We begin with some examples relating HPC to PC.

Example 4.11 (Revisiting Example 3.4). To show that the essence of PC is retained we can repeat the reasoning of Example 3.4. Consider the factors in Figure 1 to form a hierarchy without any abstract factors; so $P = \{\mathrm{F1}, \mathrm{F2}, \mathrm{F4}, \mathrm{F6}\}$, $\mathrm{Pro} = \{\mathrm{F2}, \mathrm{F4}, \mathrm{F6}\} \times \{\pi\}$, and with $\mathrm{Con} = \{(\mathrm{F1}, \pi)\}$. We consider a case G satisfying $G \vDash \{\mathrm{F1}, \mathrm{F2}, \mathrm{F4}, \pi\}$. Now for any fact situation F which is not defined on π we get:

$\{G\}, F \vDash \pi$
iff (1) for all $q \in \{\mathrm{F2}, \mathrm{F4}, \mathrm{F6}\}$: if $G \vDash q$ then $\{G\}, F \vDash q$, and
(2) for all $q \in \{\mathrm{F1}\}$: if $\{G\}, F \vDash q$ then $G \vDash q$
iff $F \vDash \{\mathrm{F2}, \mathrm{F4}\}$.

The construction used in Example 3.4 can be generalized to show that any reasoning performed in PC can be faithfully replicated in HPC. This is because factor hierarchies without abstract factors aside from π correspond exactly to the knowledge representation used for PC, and, moreover, Definition 4.8 behaves like Definition 3.2 on this fragment. More specifically, we have the following proposition.

PROPOSITION 4.12. *The theory of HPC with the additional axiom $A \subseteq \{\pi\}$ is equivalent to the theory of PC.*

Carefully proving this proposition involves some tedious verifications, see Appendix A for a sketch of the proof.

Example 4.13 (Inconsistency). As in Example 3.5, a case base of HPC can be inconsistent in the sense that there can be some $F \in CB$ with $F \vDash p$ such that $CB, F \vDash \neg p$. Note that this is not immediately as problematic as it would be in a logic where the principle of explosion holds, i.e. the situation above does not imply that $CB, F \vDash q$ for all $q \in P$. A theory of precedential constraint prescribes what it means to make decisions in accordance with a set of previous decisions, but courts can and do deviate from this prescription in practice.

Proposition 4.12 shows that if there are no abstract factors aside from π then HPC behaves exactly as PC. However, the question remains whether HPC is well behaved in terms of its forcing relation on abstract factors, and this is what we examine next.

Example 4.14 (Recursively forcing pro factors). We now consider the factor hierarchy (P, H) as specified by Figure 2. Let F be a fact situation such that $F \vDash \{\mathrm{F114}, ?\mathrm{F101}, \mathrm{F102}\}$, and G a precedent case with $G \vDash \{\mathrm{F114}, \mathrm{F101}, \mathrm{F120}, \neg\mathrm{F110}, \pi\}$. We then get

$\{G\}, F \vDash \pi$
if (1) for all $q \in \{\mathrm{F114}, \mathrm{F101}, \mathrm{F110}\}$: if $G \vDash q$ then $\{G\}, F \vDash q$, and
(2) for all $q \in \{\mathrm{F120}\}$: if $\{G\}, F \vDash q$ then $G \vDash q$
if $\{G\}, F \vDash \{\mathrm{F114}, \mathrm{F101}\}$
if $\{G\}, F \vDash \mathrm{F101}$.

We see that G does not force the decision of F for π as F is undecided on F101. But now suppose that there is a second precedent case H

with $H \vDash \{\text{F102}, \neg\text{F104}, \text{F105}, \text{F101}\}$. Then,

$$\begin{aligned}
&\{G, H\}, F \vDash \text{F101}\\
&\quad \text{if } (1) \text{ for all } q \in \{\text{F102}, \text{F104}\}\text{: if } H \vDash q \text{ then } \{G, H\}, F \vDash q, \text{ and}\\
&\qquad (2) \text{ for all } q \in \{\text{F105}\}\text{: if } \{G, H\}, F \vDash q \text{ then } H \vDash q\\
&\quad \text{if } \{G, H\}, F \vDash \text{F102}.
\end{aligned}$$

Since $F \vDash$ F102 we get $\{G, H\}, F \vDash$ F102, so $\{G, H\}, F \vDash$ F101 and in turn $\{G, H\}, F \vDash \pi$. Note also that this example works even if $H \vDash \delta$: a precedent can be involved in forcing an outcome that it itself was not decided for.

So far we considered examples in which the only con factor involved was satisfied by the precedent case, thus making clause (2) trivially satisfied. To demonstrate the purpose of the recursion in this clause we consider a modified version of Example 4.14.

Example 4.15 (Recursively forcing con factors). Suppose that in the setup of Example 4.14 we swap the truth value of the con-F101 factor F105 in H; i.e. $H \vDash \neg$F105. Suppose, moreover, that there is a precedent I with $I \vDash \{\text{F106}, \neg\text{F108}, \text{F105}\}$, and that $F \vDash$ F106. Now I forces the decision of F for F105, i.e. $\{G, H, I\}, F \vDash$ F105, and since $H \nvDash$ F105 the precedent case H no longer forces the decision of F for F101. Clearly any opponent of a decision of F for π would cite I as forcing F for F105, thereby opposing the citation of H. This example demonstrates that Definition 4.8 accounts for this.

We conclude this section with two examples of similarities between HPC and two of its sources of inspiration: the HYPO and CATO programs, and Roth's model of precedential constraint [15].

Example 4.16 (Downplaying distinctions). In HYPO and CATO a precedent case $G{:}s$ can be cited to argue that a focus fact situation F should also be decided for s. This citation can be opposed by noting *relevant differences*, also known as *distinctions*, between G and F: pro-s factors that apply in G but not in F, or con-s factors that apply in F but not in G. In other words, relevant differences are the factors obstructing $G{:}s$ from forcing the decision of F for s in terms of Definition 3.2.

In CATO such a distinction can be responded to in several ways, on the basis of its factor hierarchy. One such way is to draw an *abstract parallel*; arguing that, in effect, the difference is not relevant because it is subsumed by a more abstract factor on which the precedent and the focus fact situation do agree.

We consider a concrete example of a case G and fact situation F:

$$G \vDash \{\text{F102}, \neg\text{F104}, \text{F101}, \pi\}, \qquad F \vDash \{\neg\text{F102}, \text{F104}, \text{F101}, ?\pi\}.$$

A citation of G to argue that F should be decided for π can be opposed by noting the distinction F102, which says that the plaintiff took efforts to maintain secrecy of their information. This factor supports a decision for π and applies to G, but not to F, and so it is a distinction. However in F the factor F104 does apply, which says that the plaintiff's information was valuable for their business. For both F and G it was decided that the factor F101 applies, which says that plaintiff's information is a trade secret, due to the applying factors F104 and F102 respectively. Therefore, the distinction F102 can be downplayed, by noting that it is subsumed by the more abstract factor F101, so that the difference is not relevant.

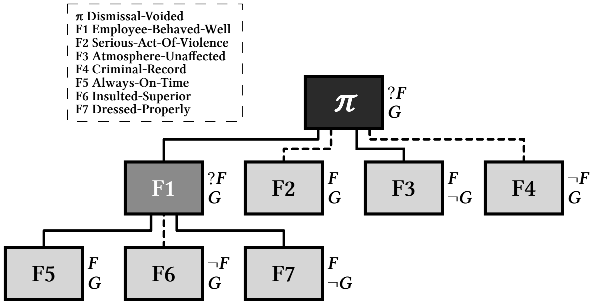

Figure 3: A small factor hierarchy for the domain of Dutch dismissal law, repeated from [16] Section 3.

Downplaying a distinction means that as a precedent, distinctions at a lower level in the hierarchy do not matter so long as the precedent and the focus case agree on more abstract, subsuming factors. In some sense this sentiment is inherent in the definition of forcing for HPC. If we evaluate whether $\{G\}, F \vDash \pi$ in the previous scenario, according to Definition 4.8, what matters is that both G and F satisfy the pro-π factor F101, and not *why* it is satisfied (in this case, for the different reasons F102 and F104).

Example 4.17 (Dialectical arguments). As a final example we demonstrate that HPC is aligned with the work by Roth and Verheij, by formalizing an example from [16] Section 3 on the domain of Dutch dismissal law. In Figure 3 a factor hierarchy for this domain is depicted. The case outcome corresponds to whether or not the dismissal can be voided, and relevant factors are considered such as whether the dismissed person has always behaved like a good employee. We consider two fact situations F and G. The factors that apply to these situations are indicated next to the factors in Figure 3, so for example $F \vDash ?\pi$, $F \vDash$ F2, and $G \vDash \neg$F7. The point of the example is that there is more support in F for F1 than in G, and so since G was decided for F1 so should F. This, in turn, means there is more support in F for π than in G, so that F should be decided for π as well. Applying Definition 4.8 to this scenario we see that it follows exactly this line of reasoning, albeit in reverse:

$$\begin{aligned}
&\{G\}, F \vDash \pi\\
&\quad \text{if } (1) \text{ for all } q \in \{\text{F1}, \text{F3}\}\text{: if } G \vDash q \text{ then } \{G\}, F \vDash q, \text{ and}\\
&\qquad (2) \text{ for all } q \in \{\text{F2}, \text{F4}\}\text{: if } \{G\}, F \vDash q \text{ then } G \vDash q\\
&\quad \text{if } \{G\}, F \vDash \text{F1}\\
&\quad \text{if } (1) \text{ for all } q \in \{\text{F5}, \text{F7}\}\text{: if } G \vDash q \text{ then } \{G\}, F \vDash q, \text{ and}\\
&\qquad (2) \text{ for all } q \in \{\text{F6}\}\text{: if } \{G\}, F \vDash q \text{ then } G \vDash q\\
&\quad \text{if } F \vDash \text{F5}, \text{ which is true by assumption.}
\end{aligned}$$

This example also shows that the same precedent case can be involved in forcing multiple outcomes in a focus case.

4.4 Monotonicity

We saw in Proposition 3.7 that PC is nonmonotonic in the fact situation with respect to the forcing relation, and of course this same counterexample applies to HPC as well.

However, the two theories do not behave the same with regards to monotonicity in the case base: unlike in PC, in HPC decisions made on the basis of precedent can be invalidated by the inclusion of more precedent cases.

PROPOSITION 4.18. *Given case bases $CB \subseteq CB'$, a fact situation F, and a factor $p \in \mathcal{P}$, the statement $CB, F \vDash p$ does not imply $CB', F \vDash p$.*

PROOF. For a counterexample we consider the factor hierarchy in Figure 2, and fact situations F, G, H satisfying

$$F \vDash \{\text{F101}, ?\text{F120}, \text{F110}, \text{F105}, ?\pi\},$$
$$G \vDash \{\text{F101}, \neg\text{F120}, \neg\text{F110}, \pi\},$$
$$H \vDash \{\text{F105}, \text{F120}, \delta\}.$$

Now we have $\{G\}, F \vDash \pi$, because the pro-π factor F101 applies in both F and G while the only con-π factor F120 does not apply in F (or more specifically, $F \nvDash$ F120). However, $\{G, H\}, F \vDash$ F120 because F120 applies in H and both F and H satisfy the pro-F120 factor F105. This means that now $\{G, H\}, F \nvDash \pi$, because $G \nvDash$ F120. □

Intuitively, the reason why PC is monotonic in the case base and HPC is not, is that in PC the added precedent case cannot interfere with the pre-existing forcing inference, whereas in HPC it can. In PC there is no room in the single-step forcing inference for the additional precedent, while in HPC the additional precedent can become part of the pre-existing multi-step forcing inference and alter its final conclusion in doing so.

Remark 4.19. It should be noted that Lemma 4.9 can also be regarded as stating a kind of monotonicity property; stating that a decision made in F cannot be invalidated by precedent. In this sense the decisions that are made in a fact situation take precedence over subsequent decisions forced by a case base.

5 DISCUSSION

In Section 4 we described in general terms how a set of factors can be given hierarchical structure in the manner of CATO's factor hierarchy, and we subsequently defined a notion of precedential constraint for a set of cases with factors based on such a factor hierarchy. We now discuss our findings in some more detail, and touch on gaps and possible expansions which may be filled by future work.

In the CATO factor hierarchy the factors are assumed to favor exactly one of the two possible outcomes, and the polarity of the links between them in the hierarchy is fully determined by this preference: if the two linked factors favor the same outcome then the link is positive, and it is negative otherwise. In the present work we do not assume that factors in the hierarchy have this preference, and thus do not impose any restrictions on the polarity of the links (given in our framework by the Pro and Con relations which together make up the hierarchical structure H). In doing so, we allow certain structures to exist within a factor hierarchy that could not occur otherwise, e.g. there may be a chain of factors $p, q, r \in P$ such that p is pro-q, q is pro-r, but p is con-r. Some further research is necessary to ascertain exactly what kinds of structures this relaxation of the requirements permits, and whether these are actually present in any naturally occurring factor hierarchy.

Another point for future work is on the bookkeeping of the precedent cases involved in Definition 4.8. This makes use of recursion, which enables multiple precedent cases to be involved in forcing the decision of a factor in some focus case. No bookkeeping is performed of the identity of these cases; all that matters to Definition 4.8 is their existence. Of course, for the intended application of XAI this will not suffice, as these cases are intended to form the starting point of an argumentative dialogue on how the system might have reached its decision. Thus, a more constructive version of Definition 4.8 is required. One way of going about this would be to inductively define a set of *precedential arguments*, each containing information about exactly what cases are involved in forcing what factors in the multi-step process. These precedential arguments should serve as witnesses to the forcing relation: meaning they should satisfy the property that a case base forces the decision of a factor if and only if such a precedential argument exist. Code for computing such arguments could then be implemented, and applied to representative machine learning data to assess the theory in practice, as was done in [18] for PC.

A second concern with respect to the goal of XAI is that the factor hierarchy defined in Section 4.1 only contains *binary* factors, i.e. propositional letters. In practice relevant legal facts are often not binary but instead take values in sets like that of the natural numbers. These types of facts are often called *dimensions* in the literature, and in [10] Horty developed an extension of PC which accomodates precedential reasoning on the basis of dimensions. A similar extension of HPC is desirable for the purpose of XAI because often the features of machine learning datasets are not binary but categorical or numerical.

Aside from practical concerns related to XAI there are also avenues for theoretical future research on HPC. For instance, our representation of cases based on a factor hierarchy abstracts from the particular reasons the court used to decide a fact situation for a factor. A more fine-grained representation could include such information. Another example would be to further investigate the problems surrounding case base inconsistency. In the present work we only briefly touched on this matter in Examples 3.5 and 4.13, but it is of primary concern in e.g. [11, 10, 5].

6 CONCLUSION

In this work we set forth a generalization of the theory of precedential constraint [11] which allows multi-step inferences on the basis of precedent cases. To do so we first formally specified what it means for a set of factors to have hierarchical structure in the style of the factor hierarchy used in the CATO program [1]. We then defined a notion of constraint for a set of cases represented using such a factor hierarchy by recursion on the hierarchical structure. The workings of the theory were demonstrated through examples, and a first step in analysing its theoretical properties was taken by demonstrating this type of reasoning to be nonmonotonic.

More broadly our goal is developing argumentative XAI methods, which produce explanations in the form of argumentative dialogues for (or possibly even against) specific decisions made by AI systems. An earlier version of such a method, presented in [14], uses the theory of precedential constraint as its backbone for representing and reasoning about training examples, which may be viewed constituting past decisions and thus, cases. However, the single-step nature of Horty's precedential constraint limits the argumentative discourse that can be had in this manner. Therefore, by generalizing it to enable multi-step inferences we hope to enrich such XAI methods. Clear steps to take in this direction for future work are to extend the theory to operate on dimensions instead of factors, and to subsequently demonstrate the theory in practice.

ACKNOWLEDGMENTS

This research was (partially) funded by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, grant number 024.004.022.

REFERENCES

[1] Vincent Aleven. 1997. *Teaching Case-Based Argumentation Through a Model and Examples.* Ph.D. Dissertation. University of Pittsburgh.

[2] Vincent Aleven and Kevin D. Ashley. 1997. Evaluating a learning environment for case-based argumentation skills. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law*, 170–179.

[3] Kevin D. Ashley. 1991. Reasoning with cases and hypotheticals in HYPO. *International Journal of Man-Machine Studies*, 34, 6, 753–796.

[4] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes Peter Wallner, and Stefan Woltran. 2013. Abstract dialectical frameworks revisited. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 803–809.

[5] Ilaria Canavotto. 2022. Precedential constraint derived from inconsistent case bases. In *Legal Knowledge and Information Systems. JURIX 2022: The Thirty-fifth Annual Conference.* Enrico Francesconi, Georg Borges, and Christoph Sorge, (Eds.) IOS Press, 23–32.

[6] Oana Cocarascu, Andria Stylianou, Kristijonas Čyras, and Francesca Toni. 2020. Data-empowered argumentation for dialectically explainable predictions. In *Proceedings of the 24th European Conference on Artificial Intelligence.* Giuseppe De Giacomo, Alejandro Catala, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, (Eds.) IOS Press, 2449–2456.

[7] Kristijonas Čyras, David Birch, Yike Guo, Francesca Toni, Rajvinder Dulay, Sally Turvey, Daniel Greenberg, and Tharindi Hapuarachchi. 2019. Explanations by arbitrated argumentative dispute. *Expert Systems with Applications*, 127, 141–156.

[8] Kristijonas Čyras, Ken Satoh, and Francesca Toni. 2016. Explanation for case-based reasoning via abstract argumentation. In *Computational Models of Argument, Proceedings of COMMA 2016.* Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, (Eds.) IOS Press, 243–254.

[9] Siegfried Gottwald. 2022. Many-Valued Logic. In *The Stanford Encyclopedia of Philosophy.* (Summer 2022 ed.). Edward N. Zalta, (Ed.) Metaphysics Research Lab, Stanford University.

[10] John Horty. 2019. Reasoning with dimensions and magnitudes. *Artificial Intelligence and Law*, 27, 3, 309–345.

[11] John F. Horty. 2011. Rules and reasons in the theory of precedent. *Legal Theory*, 17, 1, 1–33.

[12] Xinghan Liu, Emiliano Lorini, Antonino Rotolo, and Giovanni Sartor. 2022. Modelling and explaining legal case-based reasoners through classifiers. In *Knowledge and Information Systems. JURIX 2022: The Thirty-fifth Annual Conference.* Enrico Francesconi, Georg Borges, and Christoph Sorge, (Eds.) IOS Press, 83–92.

[13] Henry Prakken. 2021. A formal analysis of some factor- and precedent-based accounts of precedential constraint. *Artificial Intelligence and Law*, 29, 4, 559–585.

[14] Henry Prakken and Rosa Ratsma. 2022. A top-level model of case-based argumentation for explanation: formalisation and experiments. *Argument & Computation*, 13, 2, 159–194.

[15] Abraham Cornelis Roth. 2003. *Case-based reasoning in the law. A formal theory of reasoning by case comparison.* Dissertation. Universiteit Maastricht.

[16] Bram Roth and Bart Verheij. 2004. Dialectical arguments and case comparison. In *Legal Knowledge and Information Systems. JURIX 2004: The Seventeenth Annual Conference.* T.F. Gordon, (Ed.), 99–108.

[17] Christian Strasser and G. Aldo Antonelli. 2019. Non-monotonic Logic. In *The Stanford Encyclopedia of Philosophy.* (Summer 2019 ed.). Edward N. Zalta, (Ed.) Metaphysics Research Lab, Stanford University.

[18] Wijnand van Woerkom, Davide Grossi, Henry Prakken, and Bart Verheij. 2022. Landmarks in case-based reasoning: from theory to data. In *HHAI2022: Augmenting Human Intellect.* Stefan Schlobach, María Pérez-Ortiz, and Myrthe Tielman, (Eds.) IOS Press, 212–224.

[19] Bart Verheij. 2020. Artificial intelligence as law. *Artificial Intelligence and Law*, 28, 2, 181–206.

[20] Bart Verheij. 2003. Deflog: on the logical interpretation of prima facie justified assumptions. *Journal of Logic and Computation*, 13, 3, 319–346.

A PROOF SKETCH OF PROPOSITION 4.12

Definition A.1. A factor hierarchy (P, H) is called *single-layer* if it satisfies $A \subseteq \{\pi\}$.

Definition A.2. A *disjoint factor union* is a finite set P of propositional variables which is equal to a disjoint union $P = \mathrm{Pro} \cup \mathrm{Con}$.

A disjoint factor union is the type of knowledge representation used for models of PC, as described in Section 3.1. Central to the proof of Proposition 4.12 is the following Lemma.

LEMMA A.3. *If (P, H) is a single-layer factor hierarchy then P is given by $P = \mathrm{Pro}(\pi) \cup \mathrm{Con}(\pi) \cup \{\pi\}$.*

PROOF. It is easy to see that $\mathrm{Pro}(\pi) \cup \mathrm{Con}(\pi) \cup \{\pi\} \subseteq P$, so we only prove the other inclusion $P \subseteq \mathrm{Pro}(\pi) \cup \mathrm{Con}(\pi) \cup \{\pi\}$. To this end, consider some $p \in P$. If $p = \pi$ we are done, so assume that $p \neq \pi$. As per Definition 4.2 this means p is not maximal, meaning there is $q \in P$ such that $H(p, q)$. In general $P = B \cup A$ and so since $A \subseteq \{\pi\}$ we have $P = B \cup \{\pi\}$. Hence $q \in B \cup \{\pi\}$, but $q \in B$ contradicts $H(p, q)$ and so $q = \pi$. This means $H(p, \pi)$ and $p \in \mathrm{Pro}(\pi) \cup \mathrm{Con}(\pi)$ as desired. □

LEMMA A.4. *There is a one-to-one correspondence between single-layer factor hierarchies and disjoint factor unions.*

PROOF. Given a single-layer factor hierarchy (P, H) we define $P' := P \setminus \{\pi\}$. Furthermore, let $\mathrm{Pro}' := \mathrm{Pro}(\pi)$ and $\mathrm{Con}' := \mathrm{Con}(\pi)$. Now by Lemma A.3 we have that P' is a disjoint factor union $P' = \mathrm{Pro}' \cup \mathrm{Con}'$.

Conversely, given a disjoint factor union $P = \mathrm{Pro} \cup \mathrm{Con}$ we define a single-layer factor hierarchy (P', H) by $P' := P \cup \{\pi\}$ and

$$\mathrm{Pro}' := \mathrm{Pro} \times \{\pi\}, \quad \mathrm{Con}' := \mathrm{Con} \times \{\pi\}, \quad H := \mathrm{Pro}' \cup \mathrm{Con}' .$$

It is easy to see that this hierarchy satisfies the requirements of Definition 4.2 and the additional axiom $A \subseteq \{\pi\}$, and moreover, that the constructions thus described are one another's inverse. □

PROPOSITION A.5 (4.12). *The theory of HPC with the additional axiom $A \subseteq \{\pi\}$ is equivalent to the theory of PC.*

PROOF. The constructions from Lemma A.4 come with translations for cases from one representation to the other. For instance, any case $F{:}s$ based on a disjoint factor union $P = \mathrm{Pro} \cup \mathrm{Con}$ can be translated to a case F' of its corresponding HPC representation:

$$F' : P' \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\} : p \mapsto \begin{cases} F(p) & \text{if } p \in P, \\ \mathbf{t} & \text{if } p = \pi = s, \\ \mathbf{f} & \text{if } p = \pi \neq s. \end{cases}$$

Let $f : F{:}s \mapsto F'$ denote this operation. Similarly, a fact situation $F : P \longrightarrow \{\mathbf{t}, \mathbf{f}\}$ as in Section 3.1 can be extended to operate on $P' = P \cup \{\pi\}$ by $F(\pi) := \mathbf{u}$. Let g denote the function performing this translation. Now, writing $\Vdash$ for the forcing relation of Definition 4.8 to avoid confusion with that of Definition 3.2, we have

$$CB, F \vDash s \text{ implies } f[CB], g(F) \Vdash s,$$

where CB is some case base of PC, $f[CB] = \{f(G{:}s) \mid G{:}s \in CB\}$, and $s \in \mathcal{S}$. An analogous result holds for the other direction. □