Justification, stability and relevance for case-based reasoning with incomplete focus cases

Daphne Odekerken*
d.odekerken@uu.nl
Utrecht University
National Police Lab AI
The Netherlands

Floris Bex
f.j.bex@uu.nl
Utrecht University
Tilburg University
The Netherlands

Henry Prakken
h.prakken@uu.nl
Utrecht University
University of Groningen
The Netherlands

ABSTRACT

We define and study the notions of stability and relevance for precedent-based reasoning, focusing on Horty's result model of precedential constraint. According to this model, precedents constrain the possible outcomes for a focus case, which is a yet undecided case, where precedents and the focus case are compared on their characteristics (called dimensions). In this paper, we refer to the enforced outcome for the focus case as its *justification* status. In contrast to earlier work, we do not assume that all dimension values of the focus case have been established with certainty: rather, each dimension is assigned a set of possible values. We define a focus case as *stable* if its justification status is the same for every choice of the possible values. For focus cases that are not stable, we study the task of identifying *relevance*: which possible values should be excluded to make the focus case stable? We show how the tasks of identifying justification, stability and relevance can be exploited for human-in-the-loop decision support. Finally, we discuss the computational complexity of these tasks and provide efficient algorithms.

CCS CONCEPTS

• **Theory of computation → Theory and algorithms for application domains**; • **Applied computing → Law**.

KEYWORDS

case-based reasoning, stability, relevance, complexity, algorithms, human-in-the-loop, decision support

ACM Reference Format:
Daphne Odekerken, Floris Bex, and Henry Prakken. 2023. Justification, stability and relevance for case-based reasoning with incomplete focus cases. 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/10.1145/3594536.3595136

1 INTRODUCTION

Modelling reasoning with legal cases has been an important topic in the computational study of legal argument. This type of reasoning is applied to problems that are not decided by a legal rule but by comparing the characteristics of cases. In particular, the characteristics of a new case for which the outcome still has to be decided (called *focus case*) are compared to the characteristics of precedent cases for which the outcome has already been assigned. In the seminal work on the HYPO system for US trade secrets law [17], these characteristics were multi-valued and were called *dimensions*. Later, in influential work on the CATO system [1], the characteristics of a case were considered to be boolean and were called *factors*. Each factor or dimension has some direction: a factor's presence favours one of the two parties in a case and some dimension values are more favourable to one of the parties than other values.

Studies on reasoning with legal cases have multiple applications. The first systems were designed to generate arguments in the legal domain [17], for example for teaching purposes [1]. Later work, initiated by Horty in [8], addressed the question how precedents constrain the possible decisions for a focus case, thereby providing multiple models, including a *result model* of *precedential constraint*. Recently, this model has been used for explaining machine-learning-based decisions (e.g. [13, 15]). In this paper, we study an alternative application of the result model from [8], namely the use of the model itself as a classifier for human-in-the-loop decision support. We implemented a simplified version of this model for the classification of mala fide webshops. This is a classification problem where it is important that decisions are taken consistently. That is, if a decision is taken once, then in similar (or more extreme) situations, the same decision should be taken – which is exactly the kind of reasoning modelled by the result model of precedential constraint.

We propose to extend the result model for precedential constraint with a component that provides the possibility to express uncertainties in the dimension values of a focus case.[1] This is motivated by the situation that may occur in practical applications in which the values of some dimensions have not yet been determined with certainty. Typically, investigation into these dimensions requires time and effort. It is therefore useful to identify cases for which the same outcome is enforced, regardless of the way the uncertainties in dimension values are resolved: in those cases, there is no need for investment in further investigation.

Interestingly, the suggestion of studying hypothetical variations to the focus case by varying dimension values was already put forward for HYPO in [4], motivated by similar reasons as ours: as explained in [2], suggestions for hypothetical variations are useful in lawyers' preparation for trial, because they cannot assume to

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

ACM ISBN 979-8-4007-0197-9/23/06…$15.00
https://doi.org/10.1145/3594536.3595136

[1] In order to enable the classification of input cases with both boolean and multi-valued characteristics, we will treat factors as a special case of (two-valued) dimensions. This is the same as the approach in [15], see Section 5.2.

be certain of all elements of the fact situation, and hypotheticals suggest dimensions that they can elicit from a client or prepare to rebut. Whereas many ideas in HYPO have greatly influenced the AI and Law field [5], this aspect seems to have attracted less attention. In particular, in existing work on the result model (e.g. [8, 9]), it is assumed that the characteristics of a focus case are fully investigated.

In order to enable precedent-based reasoning with incomplete focus cases, we introduce the notion of *incomplete fact situations*, which express not only the certain characteristics of a case, but also the uncertain ones. An incomplete fact situation has a set of *completions*, which are the fact situations that can be obtained by resolving all uncertainties. We refer to the enforced outcome of the precedential constraint for a particular completion as its *justification status*. Then, we define *stability* for case-based reasoning, where incomplete fact situations are called *stable* if each completion has the same justification status. For those incomplete fact situations that are not stable, we study the task of identifying *relevance*: which uncertainties should be resolved in order to make the case stable?[2]

The notions of justification, stability and relevance can be exploited in human-in-the-loop decision support systems: if an incomplete fact situation is stable, then no further investigation is required. Otherwise, relevant updates reveal which dimensions should be investigated further. This application requires that the tasks of identifying stability status and relevant uncertainties are performed efficiently, even if there is a lot of uncertainty on the value of dimensions. This is not trivial: we will show that these tasks are situated in high complexity classes. However, under a particular assumption, the source of this complexity disappears and the tasks can be performed in polynomial time. As a final contribution, we describe efficient and scalable algorithms for identifying justification, stability and relevance. Implementations of the algorithms and full proofs can be found at https://git.science.uu.nl/D.Odekerken/lcbr.

Outline We start by recalling some preliminaries in Section 2. Subsequently, we define the notions of justification, stability and relevance for precedent-based reasoning in Section 3. We then show in Section 4 how algorithms for these tasks can be applied in practice in human-in-the-loop decision support. Practical applications require efficient algorithms, which we discuss in Section 5. Related work is discussed in Section 6, after which we conclude in Section 7.

2 PRELIMINARIES

Before formally defining the notions of justification, stability and relevance, we recall the definitions of fact situations, cases and the notion of precedential constraint. In order to define these formally, we introduce the notion of a domain. This is the set of all dimensions related to the cases about which we want to reason.

Definition 1 (Domain). *Let $\mathcal{D}$ be a set of dimensions, where a dimension is a tuple $d = (V, \leq_{\mathsf{PRO}}, \leq_{\mathsf{CONTRA}})$ such that:*

- *V is a set of values that can be assigned to d; and*
- *$\leq_{\mathsf{PRO}}$ and $\leq_{\mathsf{CONTRA}}$ are two partial orders on V such that $v \leq_{\mathsf{PRO}} v'$ iff $v' \leq_{\mathsf{CONTRA}} v$.*

[2]The idea of defining a stability status and relevance notion, based on possible future information updates, was developed in other settings as well: [11] study stability for ASPIC$^+$ argumentation theories with an incomplete knowledge base, whereas [12] define stability and relevance for incomplete (abstract) argumentation frameworks. [7] define stability for Defeasible Denotic Logic theories.

We refer to $\mathcal{D}$ as the domain. The notation $V(\mathcal{D}, d)$ refers to the set of values that can be assigned to the dimension d in domain $\mathcal{D}$.

Example 1. *An example of a dimension related to the classification of web shops is* `complaints` $= (V, \leq_{\mathsf{PRO}}, \leq_{\mathsf{CONTRA}})$*, the number of complaints submitted against a web shop, where* `complaints` *can take any natural number (including 0) as a value and the higher the value assigned to* `complaints` *in a case, the more CONTRA this case. Formally: $V = \mathbb{N}$, $\leq_{\mathsf{PRO}} = \geq$ and $\leq_{\mathsf{CONTRA}} = \leq$. Another example of a dimension, which can only have two values, is* `trustmark` $= (\{0, 1\}, \leq, \geq)$*. Note that high values for* `trustmark` *should be interpreted differently than high values for* `complaints`*: for* `trustmark`*, higher values are considered more PRO whereas higher values for* `complaints` *are considered more CONTRA, as registration at a trustmark company makes a web shop more trustworthy.*

Note that in this approach, factors are seen as a special case of dimensions. A fact situation is then a combination of dimensions with corresponding values, as defined next. Note that we require each fact situation to be assigned a value for each dimension.

Definition 2 (Fact situation). *A fact situation c within a domain $\mathcal{D}$ is a set D of value assignments to all dimensions in $\mathcal{D}$. A value assignment is a pair (d, v) where $d \in \mathcal{D}$ and $v \in V(\mathcal{D}, d)$. The notation $v(d, c)$ denotes the value of dimension d in fact situation c.*

Based on the dimensions of the fact situation, an outcome can be assigned. A fact situation paired with the outcome is a case.

Definition 3 (Case). *Within a domain $\mathcal{D}$, a case is a (c, o)-pair, where c is a fact situation within the domain $\mathcal{D}$ and o is the assigned outcome of the case, which can be either PRO or CONTRA.*

We then refer to a set of cases as a case base.

Definition 4 (Case base). *A case base $\mathcal{C}$ within a domain $\mathcal{D}$ is a set of cases within $\mathcal{D}$.*

Example 2. *Table 1 gives an example of a case base, consisting of three cases, in the web shop classification domain.*

Based on the ordering of dimension values, cases can be compared to each other under some conditions, which are specified in the following definition, adapted from [9]'s Definition 12. The idea is that a case with dimensions with values that are "at least as PRO" as a the values in case that is already assigned a PRO outcome is constrained to be PRO. Similarly, a case that is at least as CONTRA as a case with CONTRA outcome is assigned to be CONTRA.

Definition 5 (Strength for a side). *Let c_1 and c_2 be two fact situations within some domain $\mathcal{D}$. Then c_1 is at least as strong as c_2 for the side PRO – written $c_1 \geq_{\mathsf{PRO}} c_2$ – iff for all $d \in \mathcal{D}$: $v(d, c_1) \geq_{\mathsf{PRO}} v(d, c_2)$. Analogously, c_1 is at least as strong as c_2 for the side CONTRA ($c_1 \geq_{\mathsf{CONTRA}} c_2$) iff for all $d \in \mathcal{D}$: $v(d, c_1) \geq_{\mathsf{CONTRA}} v(d, c_2)$.*

Using the cases from the case base, in specific situations the outcome of a new fact situation can be predicted ("forced") based on the so-called *a fortiori* constraint [8].

Definition 6 (A fortiori constraint). *Let $\mathcal{C}$ be a case base and c a fact situation. Then c is forced towards outcome PRO by $\mathcal{C}$ iff there is some (c', PRO) in $\mathcal{C}$ such that $c \geq_{\mathsf{PRO}} c'$; c is forced towards outcome CONTRA by $\mathcal{C}$ iff there is some $(c', CONTRA)$ in $\mathcal{C}$ such that $c \geq_{\mathsf{CONTRA}} c'$.*

	trustmark $(\{0,1\},\leq,\geq)$	**terms** $(\{0,1\},\leq,\geq)$	**fake_payment_option** $(\{0,1\},\geq,\leq)$	**complaints** $(\mathbb{N},\geq,\leq)$	**outcome**
c_1	1	1	1	10	PRO
c_2	0	0	1	5	CONTRA
c_3	0	1	1	2	CONTRA

Table 1: Example case base in the web shop classification domain, where columns represent dimensions and rows represent cases. The values in the table correspond to the dimension values assigned to the cases.

Example 3. *Consider the case base $\mathcal{C}$ from Table 1 and suppose that we have a new case c_4 which is the same as case c_1, except that only 3 complaints have been submitted (so $v(\texttt{complaints}, c_4) = 3$). As* `complaints` *is a dimension with $\geq_{\texttt{PRO}} = \leq$, c_4 is at least as strong as c_1 for the side PRO, so c_4 is forced towards outcome PRO by $\mathcal{C}$.*

Note that a new case only has a forced outcome if there is a precedent in the case base for which the a fortiori constraint applies.

3 JUSTIFICATION, STABILITY AND RELEVANCE

We introduce the notions of justification, stability and relevance in this and the following two sections.

3.1 Justification

The notion of justification directly uses the a fortiori constraint to assign an outcome to focus cases, i.e. any (new) fact situation that is not yet assigned an outcome, based on the case base.

Definition 7 (Justification status). *Let $\mathcal{C}$ be a case base and c a fact situation. Then the justification status of c given $\mathcal{C}$ is:*

- *PRO if c is forced towards outcome PRO by $\mathcal{C}$;*
- *CONTRA if c is forced towards outcome CONTRA by $\mathcal{C}$;*
- *UNDECIDED otherwise.*

Note that there is always only one justification status under the assumption that the case base is consistent. In general, this is quite a strong assumption [13], but for applications in human-in-the-loop decision support we consider it to be reasonable as users have control on the case base; for further discussion, see Section 4.

For any focus case c, the justification status w.r.t. the case base $\mathcal{C}$ can be determined in polynomial time using the following procedure: iterate over all cases in the case base, and for each case (c', o) in $\mathcal{C}$ check if c is forced towards outcome o by c'. If this applies, return the justification status o; otherwise, return UNDECIDED.

3.2 Stability

The algorithm from the previous section can be used for deciding the justification status of new cases for which the dimension values of the case are certain. In practice, this is not always the case. Returning to our running example on the web shop classification problem, consider an application in which some, but not all, of the dimensions are extracted automatically with reasonable accuracy.

Example 4. *Consider the dimension* `terms` $= (\{0,1\},\leq,\geq)$*, stating that a web shop contains a terms and conditions page. This dimension can be extracted automatically by searching for specific links in the html code. However, it is possible that this automatic procedure has failed to detect some page and therefore assiged the wrong value.*

Other dimensions cannot be extracted automatically at all. This is, for example, the case for the dimension `fake_payment_option` $= (\{0,1\},\geq,\leq)$*, which states that the web shop displays a safe payment option on one of its web pages, but it is not actually possible to pay for a product with this method. This dimension cannot be determined automatically, as it requires starting the payment procedure.*

This example shows that an imperfect (but pragmatic) analysis can cause incomplete knowledge of the fact situation, in which we would like to keep multiple values into account for specific dimensions. In order to enable explicit representation of these possible values, we introduce the notion of an incomplete fact situation.

Definition 8 (Incomplete fact situation). *An incomplete fact situation c within a domain $\mathcal{D}$ is a tuple $c = (D, D^?)$ where:*

- *D is the assignment of a value for each dimension: for each $d = (V, \leq_{\texttt{PRO}}, \leq_{\texttt{CONTRA}})$ in $\mathcal{D}$ it contains at most one tuple (d, v) where $v \in V(\mathcal{D}, d)$ is the current value assignment. The notation $v(d, c)$ denotes the current value of dimension d in incomplete fact situation c, which may be undefined; and*
- *$D^?$ is the assignment of a possible value set for each dimension: for each $d = (V, \leq_{\texttt{PRO}}, \leq_{\texttt{CONTRA}})$ in $\mathcal{D}$ it contains a tuple (d, V_p) where $V_p \subseteq V(\mathcal{D}, d)$ is the set of possible values. The notation $V_p(d, c)$ denotes the set of possible dimension values of dimension d in incomplete fact situation c. For each dimension $d \in \mathcal{D}$, $V_p(d, c)$ must contain at least one value and if $v(d, c)$ is defined then $v(d, c) \in V_p(d, c)$.*

Example 5. *An example of an incomplete fact situation within domain $\mathcal{D}$ is $c = (D, D^?)$ with assignments $D = \{(\texttt{trustmark}, 1), (\texttt{terms}, 0), (\texttt{complaints}, 5)\}$ and $D^? = \{(\texttt{trustmark}, \{1\}), (\texttt{terms}, \{0,1\}), (\texttt{fake\_payment\_option}, \{0,1\}), (\texttt{complaints}, 5)\}$.*

This expresses the situation where the web shop is registered at the trustmark company, no terms and conditions page has been found (yet) and 5 complaints were submitted. There may be a fake payment option, but this has not yet been investigated. In addition, further investigation may result in finding terms and conditions. The other dimensions cannot change, as they have a single possible value.

By obtaining more information on an incomplete fact situation, the set of possible values can be reduced. At the point where all these uncertainties are resolved, the fact situation no longer has to be considered incomplete. Next, we define completions of an incomplete fact situation, which are (complete) fact situations that can be obtained by selecting a single value from the set of possible values for each dimension.

Definition 9 (Completions). *Given an incomplete fact situation $c = (D, D^?)$, a completion is a fact situation $c' = D^*$ such that for each $(d, v) \in D^* : v \in V_p(d, c)$.*

Example 6. *A completion of the incomplete fact situation in Example 5 is* $c' = \{(\texttt{trustmark}, 1), (\texttt{terms}, 1), (\texttt{fake\_payment\_option}, 0), (\texttt{complaints}, 5)\}$.

Even though a focus case may have an incomplete fact situation, sometimes it is still possible to establish the justification status for any completion, since all completions have the same justification status. This is interesting as it implies that it is not necessary to further investigate the yet uncertain dimension values. To identify these situations, we introduce the notion of stability status.

Definition 10 (Stability status). *Let $\mathcal{C}$ be a case base and c an incomplete fact situation. Then the stability status of c given $\mathcal{C}$ is:*

- *Stable-PRO if the justification status of each completion c' of c given $\mathcal{C}$ is PRO;*
- *Stable-CONTRA if the justification status of each completion c' of c given $\mathcal{C}$ is CONTRA;*
- *Stable-UNDECIDED if the justification status of each completion c' of c given $\mathcal{C}$ is UNDECIDED; and*
- *Unstable otherwise.*

Example 7. *Consider in the web shop classification domain $\mathcal{D}$ the incomplete fact situation* $c = (D, D^?)$ *where* $D^? = \{(\texttt{trustmark}, \{1\}), (\texttt{terms}, \{1\}), (\texttt{fake\_payment\_option}, \{0, 1\}), (\texttt{complaints}, 4)\}$.

Let $\mathcal{C}$ be the case base from Table 1. Then c is Stable-PRO w.r.t. $\mathcal{C}$, because both completions of c are PRO w.r.t. $\mathcal{C}$.

3.3 Relevance

If an incomplete fact situation is Stable-PRO, Stable-CONTRA or Stable-UNDECIDED, it is not necessary to investigate the possible dimension values to be certain on the justification status. Another possibility is that the incomplete fact situation is Unstable. Then there are some dimensions for which it is still relevant to investigate if some of their possible values can be excluded. In this section, we define these relevant updates.

In order to do so, we first need the additional notions of partial completions and minimal stable partial completions. A partial completion is, just like a completion, some notion of refinement of an incomplete focus case. However, a difference is that in a partial completion, part of the information may still be uncertain, whereas in a completion the exact value for each of the dimensions is certain.

Definition 11 (Partial completion). *Within a domain $\mathcal{D}$, given an incomplete fact situation* $c = (D, D^?)$*, a partial completion is an incomplete fact situation* $c' = (D', D^{?\prime})$ *such that:*

- *for each $(d, v) \in D'$: $v \in V_p(d, c)$; and*
- *for each $(d, V'_p) \in D^{?\prime}$: $V'_p \subseteq V_p(d, c)$.*

We denote all partial completions of some incomplete fact situation c by $F(c)$. Note that for each c, $c \in F(c)$.

Partial completions and completions are related in the following way: for each completion $c' = D'$ of some incomplete focus case c, there is some partial completion $(D', D^?) \in F(c)$ where $D^? = \{(d, \{v\}) \mid (d, v) \in D'\}$. Using this notion of partial completions, we then introduce minimal stable partial completions.

Definition 12 (Minimal stable partial completions). *Let $\mathcal{C}$ be a case base and let c be an incomplete fact situation. Let j be a justification status (PRO, CONTRA or UNDECIDED). Then $c' \in F(c)$ is a minimal stable-j partial completion w.r.t. $\mathcal{C}$ and c iff:*

- *c' is Stable-j given $\mathcal{C}$; and*
- *there is no partial completion $c'' \in F(c) \setminus \{c'\}$ such that c'' is Stable-j given $\mathcal{C}$ and $c' \in F(c'')$.*

Example 8. *Consider in the web shop classification domain $\mathcal{D}$ the incomplete fact situation* $c = (D, D^?)$ *where:* $D^? = \{(\texttt{trustmark}, \{0\}), (\texttt{terms}, \{0, 1\}), (\texttt{fake\_payment\_option}, \{0, 1\}), (\texttt{complaints}, \{6\})\}$.

Let $\mathcal{C}$ be the case base from Table 1. Then the incomplete fact situation $c' = (\emptyset, D^{?\prime})$ *where* $D^{?\prime} = \{(\texttt{trustmark}, \{0\}), (\texttt{terms}, \{0\}), (\texttt{fake\_payment\_option}, \{1\}), (\texttt{complaints}, \{6\})\}$ *is a partial completion that is Stable-CONTRA. Another partial completion that is Stable-CONTRA is* $c'' = (\emptyset, D^{?\prime\prime})$ *where* $D^{?\prime\prime} = \{(\texttt{trustmark}, \{0\}), (\texttt{terms}, \{0, 1\}), (\texttt{fake\_payment\_option}, \{1\}), (\texttt{complaints}, \{6\})\}$.

Given that $c' \in F(c'')$ and $c' \neq c''$, c'' is a minimal stable-j partial completion w.r.t. $\mathcal{C}$ and c, while c' is not.

Next, we define relevant updates, which are those updates that lead towards a minimal stable partial completion.

Definition 13 (Relevance). *Within some domain $\mathcal{D}$, let $\mathcal{C}$ be a case base and let* $c = (D, D^?)$ *be an incomplete fact situation. Let j be a justification status (PRO, CONTRA or UNDECIDED). Then for a given dimension $d \in \mathcal{D}$, the removal of possible value $v \in V_p(d, c)$ is j-relevant w.r.t. $\mathcal{C}$ and c iff there is some minimal stable-j partial completion* $c' = (D', D^{?\prime})$ *w.r.t. $\mathcal{C}$ and c such that $v \notin V_p(d, c')$.*

Example 9. *We reconsider the domain $\mathcal{D}$, case base $\mathcal{C}$, incomplete fact situation c and minimal stable-j partial completions from Example 8. Given that c'' was a minimal stable partial completion where* $0 \notin V_p(\texttt{fake\_payment\_option}, c'')$*, the removal of* 0 *for* `fake_payment_option` *is CONTRA-relevant w.r.t. $\mathcal{C}$ and c.*

Updates can have multiple relevance statuses at the same time. We illustrate this in the following example.

Example 10. *Let $\mathcal{D} = \{d_1\}$ be a domain with* $d_1 = (\{1, 2, 3\}, \leq, \geq)$ *and $\mathcal{C}$ is a case base* $\{(c_1, \texttt{PRO}), (c_2, \texttt{CONTRA})\}$ *where* $c_1 = \{(d_1, 3)\}$ *and* $c_2 = \{(d_1, 1)\}$*. Let* $c = (D, D^?)$ *be an incomplete fact situation with* $D^? = \{(d_1, \{1, 2, 3\})\}$*. Then the minimal stable-PRO partial completion is* $(\{(d_1, 3)\}, \{(d_1, \{3\})\})$*; the minimal stable-CONTRA partial completion is* $(\{(d_1, 1)\}, \{(d_1, \{1\})\})$ *and the minimal stable-UNDECIDED partial completion is* $(\{(d_1, 2)\}, \{(d_1, \{2\})\})$*. Consequently, the removal of value 1 for d_1 is both PRO- and UNDECIDED-relevant, whereas the removal of value 3 is both CONTRA- and UNDECIDED-relevant and the removal of value 2 is both PRO- and CONTRA-relevant.*

However, for specific dimensions the removal of a value cannot be both PRO- and CONTRA-relevant. This applies in particular for dimensions that can have two values that are comparable to each other by the given ordering. We formally prove this in the following proposition.

Proposition 1. *For any case base $\mathcal{C}$, incomplete fact situation c and dimension d such that* $d = (\{v_1, v_2\}, \leq_{\texttt{PRO}}, \leq_{\texttt{CONTRA}})$ *and* $\leq_{\texttt{PRO}} = \{(v_1, v_1), (v_1, v_2), (v_2, v_2)\}$*, the removal of some possible value v for d cannot be both PRO- and CONTRA-relevant.*

Proof. Suppose that we are given a case base $\mathcal{C}$, incomplete fact situation c and dimension d such that $d = (\{v_1, v_2\}, \leq_{\texttt{PRO}}, \leq_{\texttt{CONTRA}})$ and value $v \in \{v_1, v_2\}$.

(1) Suppose that removal of v for d is PRO-relevant w.r.t. $\mathcal{C}$ and c. First, we show that $v = v_1$: suppose, towards a contradiction, that $v = v_2$. Then there is a minimal stable-PRO partial completion $c_1 = (D, D^?)$ w.r.t. $\mathcal{C}$ and c such that $v_2 \notin V_p(d, c_1)$. Then $V_p(d, c_1) = \{v_1\}$ and for each completion c_2 of c_1, there is some case $(c', \text{PRO}) \in C$ such that $c' \leq_{\text{PRO}} c_2$. Now construct c_3 that equals c_1, except for the possible values for d: $V_p(d, c_3) = \{v_1, v_2\}$. Note that c_3 must be in $F(c)$, as c_1 is a partial completion of c and c_3 is almost the same, except that the possible values for d also include v_2, which must have been in $V_p(d, c)$ as we assumed that the removal of v_2 is PRO-relevant for c. Given that $v_1 \leq_{\text{PRO}} v_2$, it must be that for each completion c_4 of c_3, there is some completion c_2 of c_1 such that $c_2 \leq_{\text{PRO}} c_4$, hence there is some case $(c', \text{PRO}) \in C$ such that $c' \leq_{\text{PRO}} c_4$. But then c_1 was not minimal, as c_1 is a partial completion of c_3; contradiction.
(2) If the removal of v for d is CONTRA-relevant w.r.t. $\mathcal{C}$ and c then $v = v_2$ (analogous to the proof in item 1).

Given that only the removal of v_1 for d can be PRO-relevant, while only the removal of v_2 for d can be CONTRA-relevant w.r.t. $\mathcal{C}$ and c and $V_p(d, c) = \{v_1, v_2\}$, there can be no v such that the removal of v is both PRO- and CONTRA-relevant w.r.t. $\mathcal{C}$ and c. □

Finally, we prove that iteratively performing j-relevant updates to any incomplete fact situation c eventually results in a minimal j-stable partial completion, provided that c has a completion with justification status j.

Proposition 2. *For any case base $\mathcal{C}$, incomplete fact situation c and justification status j, let ϕ_j be an arbitrary function from $F(c)$ to $F(c)$ such that, for a given $c' \in F(c)$:*

(1) *if there is some d for which there is some v such that removing v is j-relevant w.r.t. $\mathcal{C}$ and c', then $\phi_j(c')$ returns a partial completion c'' of c' such that $v \notin V_p(d, c'')$ while for all d' such that $d' \neq d : V_p(d', c'') = V_p(d', c')$.*
(2) *otherwise, $\phi_j(c') = c'$.*

Then for any partial completion $c' \in F(c)$ that has a completion that is j w.r.t. $\mathcal{C}$, there is some minimal stable-j partial completion c'' that can be obtained within a finite number n of applications of ϕ_j (so $\phi_j^n(\phi_j^{n-1}(\ldots \phi_j^1(c') \ldots)) = c''$).

Proof. For any case base $\mathcal{C}$ in domain $\mathcal{D}$, incomplete fact situation c and justification status j, let ϕ_j be an arbitrary function as specified above and suppose that $c' \in F(c)$ has a completion that is j w.r.t. $\mathcal{C}$. An upper bound on the number of iterations to reach a fixed point is $T = \Sigma_{d \in \mathcal{D}}(|V_p(d, c')| - 1)$: after T iterations, no value can be removed from any dimension, as there is only one possible value for each dimension. We prove by induction that, after the i'th iteration, $\phi_j^i(c')$ has a completion that is j w.r.t. $\mathcal{C}$. For $i = 0$, this follows from the assumption that c' has a completion that is j w.r.t. $\mathcal{C}$. Assuming the induction hypothesis for i, the proof for $(i+1)$ follows from the definition of ϕ_j: given that $c_i = \phi_j^i(\phi_j^{i-1}(\ldots \phi_j^1(c') \ldots))$ has a completion that is j, there must be some minimal stable-j partial completion w.r.t. $\mathcal{C}$ and c'. Then there are two possibilities:

- c_i is a minimal stable-j partial completion w.r.t. $\mathcal{C}$ and c. Then $\phi_j^{i+1}(c_i) = c_i$. From the induction hypothesis, it follows that $\phi_j^{i+1}(c_i)$ has a completion that is j w.r.t. $\mathcal{C}$.
- Alternatively, there is some d and v such that removal of v for d is j-relevant w.r.t. $\mathcal{C}$ and c and $\phi_j^{i+1}(c_i)$ is obtained by removing v from the possible values for d. Given that removing v is j-relevant, there is some minimal stable-j partial completion of c_i that does not have v in its possible values. Then this partial completion is still a partial completion of $\phi_j(c_i)$, so $\phi_j^{i+1}(c_i)$ has a completion that is j w.r.t. $\mathcal{C}$.

After T iterations, $\phi_j^T(c_i)$ has a completion that is j w.r.t. $\mathcal{C}$ and there can be no value that is j-relevant to remove for any dimension. Then $\phi_j(c_i)$ is a minimal stable-j completion w.r.t. $\mathcal{C}$. □

In the next section, we show how the property proven in Proposition 2 can be used in human-in-the-loop decision support systems.

4 APPLICATION IN HUMAN-IN-THE-LOOP DECISION SUPPORT

Having defined justification, stability and relevance, we now illustrate how these notions can be exploited in decision support systems that can be used by human analysts to make consistent decisions on incomplete focus cases. This has been implemented in a human-in-the-loop decision support system for web shop classification at a national police force [10] (albeit in a simplified version). From practical experience, we know that for many, though not all, dimensions the values can be extracted automatically. In modelling the incomplete fact situation $c = (D, D^?)$, we take the availability and quality of automatic extraction methods into account:

- For dimensions d, the value v can be extracted accurately. For example, the value of `complaints` is obtained from the complaint registration system and the value of `trustmark` can be requested directly from the trustmark company. In these cases, we include (d, v) to D and $(d, \{v\})$ to $D^?$.
- For other dimensions, the automatic extraction procedure is imperfect (e.g. `terms`). Then we include (d, v) to D, but also add alternative possible values to $V_p(d, c)$.
- Alternatively, no automatic procedure is possible for a dimension (e.g. `fake_payment_option`). In that case, D does not contain a value for d and $V_p(d, c) = V(\mathcal{D}, d)$.

In the remainder of this section, we show how the analyst uses the system to make a consistent and well-informed decision on c, without spending more resources than necessary. The implemented system has more than four dimensions, but for brevity we only refer to the domain $\mathcal{D}$ of our running example and case base $\mathcal{C}$ from Table 1.

A high-level overview of the decision support system is illustrated in Figure 1. If all dimensions have a value in D, the analyst can obtain the justification status, which can be seen as an initial advice. In situations where some $d \in \mathcal{D}$ still has multiple possible values ($|V_p(d, c) > 1|$), the analyst also obtains the stability status. If this status turns out to be Unstable, then there is at least one update that would cause a change in the justification (and stability) status. All updates causing such a change are then identified using the algorithm for relevance and presented to the analyst. The analyst can then choose to further investigate (some of) the dimensions related to these updates and, after this investigation, update the incomplete fact situation. If the analyst obtains more information on the possible values V_p' to be assigned to some dimension d where

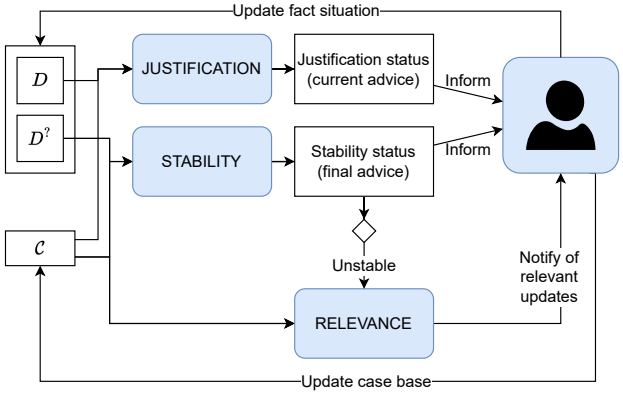

Figure 1: High-level overview of the proposed human-in-the-loop decision-making process, involving algorithms for justification, stability and relevance identification.

$(d, V_p) \in D^?$, then in the updated fact situation $c' = (D', D^{?\prime})$, (d, V_p) is replaced by (d, V_p'). Additionally, the value (d, v_c') in D' may have to be changed.

After this update, the algorithms for justification and stability (and, if necessary, relevance) can be executed again, until a stable situation has been reached (which, by Proposition 2, will eventually happen if only relevant dimension values are removed) or the analyst decides that further investigation on the fact situation is no longer opportune given the available resources. At this point, the analyst makes a decision based on D', that is, on the current dimension values of c', the most updated version of the incomplete fact situation – which may or may not contain any uncertain elements.

Suppose, for example, that $D = \{(\texttt{trustmark}, 0), (\texttt{terms}, 0), (\texttt{complaints}, 6)\}$ and that $D^? = \{(\texttt{trustmark}, \{0\}), (\texttt{terms}, \{0, 1\}), (\texttt{fake\_payment\_option}, \{0, 1\}), (\texttt{complaints}, \{6\})\}$. Then c has CONTRA and UNDECIDED completions. Recall from Example 9 that removal of 0 for `fake_payment_option` is CONTRA-relevant w.r.t. $\mathcal{C}$ and c. In addition, removal of 1 for `fake_payment_option` is UNDECIDED-relevant. The analyst starts the payment procedure and detects a fake payment option. Then $D' = \{(\texttt{trustmark}, 0), (\texttt{terms}, 0), (\texttt{fake\_payment\_option}, 1), (\texttt{complaints}, 6)\}$ and $D^{?\prime} = \{(\texttt{trustmark}, \{0\}), (\texttt{terms}, \{0, 1\}), (\texttt{fake\_payment\_option}, \{1\}), (\texttt{complaints}, \{6\})\}$. Given that c' is Stable-CONTRA, further investigation is not necessary.

Subsequently, the analyst can choose to add D' and the decided outcome to the case base. Given our assumption that the analyst decides consistently, we assume that the analyst's decision is in line with the justification status of D': if this outcome is PRO, we assume that the analyst decides PRO; if the justification status is CONTRA, we expect the analyst to decide CONTRA. If the justification status is UNDECIDED, the analyst can choose either PRO or CONTRA. For the latter situation, it is particularly interesting to add D' to the case base: it could force other focus cases, which would initially have the justification status UNDECIDED, to become PRO or CONTRA.

On a final note, it may occur that the analyst's decision is incompatible with the justification status obtained from the case base: that is, either the analyst decides on outcome o = PRO while the justification status o' is CONTRA, or o' is PRO while o is CONTRA. Such a situation can have multiple causes, including concept drift in the interpretation of dimension values, a mistake in the decision for an earlier case, or the new case being an outlier. In any case, the analyst is not allowed to add the fact situation D' with outcome o to the case base: that would cause the case base to become inconsistent. The analyst then has four options:

(1) Do not add D' with outcome o to the case base;
(2) Remove all cases with outcome o' that force D' to be o' from the case base and then (optionally) add D' with outcome o to the case base;
(3) Correct the fact situations of all cases with outcome o' that force D' to be o' from the case base and then (optionally) add D' with outcome o to the case base; or
(4) Correct D', so it becomes D'' (which is not forced to be o') and then add D'' with outcome o to the case base.

The choice between these four options is up to the analyst. By always choosing one of these options after an incompatible decision is made, the analyst ensures that the case base remains consistent.

5 COMPLEXITY AND ALGORITHMS

At this point we have formally introduced the notions of justification, stability and relevance and we have shown how these notions are applicable for human-in-the-loop decision support systems. In Section 3.1, we described a polynomial-time procedure for determining the stability status. In this section, we will focus on the complexity of stability and relevance. In addition, we will give algorithms for these tasks and prove under which conditions these algorithms run in polynomial time.

5.1 Computing stability

To determine the stability status of an incomplete fact situation, one could construct all possible completions and run the justification algorithm for each of them. However, this approach would be very inefficient, as we illustrate with the following example.

Example 11 (Exponential blowup of naive stability algorithm). *Suppose that we are given some case base $\mathcal{C}$ within the domain $\mathcal{D}$ and some incomplete fact situation c within this domain. Assume that for each $d_i \in \mathcal{D}$, $d_i = (\{1, \ldots, 6\}, \leq, \geq)$, so each dimension has six possible values and higher values are considered to be more PRO. Now we consider the number of calls to the justification algorithm for a varying number of dimensions. First suppose that $|\mathcal{D}| = 1$ and that $c = (D, D^?)$ where $D^? = \{(d_1, \{1, \ldots, 6\})\}$. From a computational perspective, this is a worst case scenario, as the naive stability algorithm requires 6 calls to the justification algorithm: one for each of the 6 completions of c. As the number of dimensions increases, the number of calls to the justification algorithm grows exponentially. Suppose, for example, that $|\mathcal{D}| = 3$ and consider a focus case $c' = (D', D^{?\prime})$ where $D^{?\prime} = \{(d_i, \{1, \ldots, 6\}) \mid i \in \{1, 2, 3\}\}$. As c' has $6^3 = 216$ completions, the naive stability algorithm requires 216 calls to the justification algorithm. Similarly, if we have 6 dimensions that can take 6 different values, the number of calls is 46656.*

This naive approach is clearly impractical for actual applications, in which there could be a large number of dimensions, each of which could have many possible values. Fortunately, it is possible

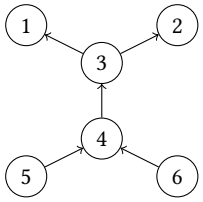

Figure 2: Hasse diagram of the ordering $\leq'_{\text{CONTRA}} = \{(6,6),(6,4),(6,3),(6,2),(6,1),(5,5),(5,4),(5,3),(5,2),(5,1),(4,4),(4,3),(4,2),(4,1),(3,3),(3,2),(3,1),(2,2),(1,1)\}$ on the set $\{1,\ldots,6\}$.

to compute the stability status in a faster way, strongly limiting the number of calls to the justification algorithm. The problem of deciding if an incomplete fact situation is Stable-UNDECIDED can be solved in polynomial time, as we show next.

Proposition 3. *Deciding if an incomplete fact situation is Stable-UNDECIDED w.r.t. some case base is in P.*

Proof. For a given incomplete fact situation c and case base $\mathcal{C}$, c is Stable-UNDECIDED w.r.t. $\mathcal{C}$ iff for each case $(c', o) \in \mathcal{C}$ there is some dimension d with value $v \in v(d, c')$ such that there is no $v_p \in V_p(d, c)$ such that $v_p \geq_o v$. This can be checked in polynomial time, by iterating over the cases in $\mathcal{C}$. □

For deciding if an incomplete fact situation c is Stable-PRO, Stable-CONTRA or Unstable, we use a different strategy. As we will prove later in this section, for these stability statuses it suffices to only consider those completions that are most likely to have a different justification status. Informally, this works as follows:

- To determine if c is Stable-PRO, construct the "most CONTRA completions" by choosing the "most CONTRA" values for each dimension within the set of possible values. If the justification algorithm classifies each of the resulting fact situations as PRO, then c was Stable-PRO.
- To determine if c is Stable-CONTRA, construct the "most PRO completions". If these are all assigned CONTRA, then c was Stable-CONTRA.
- In order to decide if c is Unstable, construct both all most PRO and all most CONTRA completions and compute their justification statuses. If not all cases are assigned the same status, then c was Unstable.

Formally, these most PRO or CONTRA completions are constructed as follows:

Definition 14 (Extreme completions). *Let $c = (D, D^?)$ be an incomplete fact situation within a domain $\mathcal{D}$. For each of the dimensions $d_i \in \mathcal{D}$, let the most PRO values of this dimension be $MPD(d_i, c) = \{V'_p \subseteq V_p(d,c) \mid$ for each $v'_p \in V'_p$: for each $v_p \in V_p$: if $v_p \geq_{\text{PRO}} v'_p$ then $v'_p \geq_{\text{PRO}} v_p\}$. The set of most PRO completions of c is $MP(c) = \{(d_i, v_i) \mid d_i \in \mathcal{D}, v_i \in MPD(d_i, c)\}$ Similarly, for each of the dimensions $d_i \in \mathcal{D}$, let the most CONTRA values be $MCD(d_i, c) = \{V'_p \subseteq V_p(d,c) \mid$ for each $v'_p \in V'_p$: for each $v_p \in V_p$: if $v_p \geq_{\text{CONTRA}} v'_p$ then $v'_p \geq_{\text{CONTRA}} v_p\}$. The set of most CONTRA completions of c is $MC(c) = \{(d_i, v_i) \mid d_i \in \mathcal{D}, v_i \in MCD(d_i, c)\}$.*

Example 12. *Consider a dimension with six possible values $d = (\{1,\ldots,6\}, \leq'_{\text{PRO}}, \leq'_{\text{CONTRA}})$ where $\leq'_{\text{CONTRA}}$ is the ordering illustrated in Figure 2. Let $c = (\{(d,1)\}, \{(d,\{1,\ldots,6\})\})$ be an incomplete fact situation in domain $\{d\}$. Then $MPD(d,c) = \{5,6\}$ and $MCD(d,c) = \{1,2\}$; $MP(c) = \{\{(d,5)\},\{(d,6)\}\}$ and $MC(c) = \{\{(d,1)\},\{(d,2)\}\}$.*

In the following proposition, we prove that checking the justification status of these extreme completions suffices to determine the stability status of any incomplete fact situation.

Proposition 4. *Given a case base $\mathcal{C}$ and an incomplete fact situation $c = (D, D^?)$:*

1. *c is Stable-PRO given $\mathcal{C}$ iff each $MC(c)$ is PRO given $\mathcal{C}$;*
2. *c is Stable-CONTRA given $\mathcal{C}$ iff each $MP(c)$ is CONTRA given $\mathcal{C}$; and*
3. *c is Unstable given $\mathcal{C}$ iff the justification status of some $MP(c)$ does not equal the justification status of some $MC(c)$ given $\mathcal{C}$.*

Proof. The proof for items 1 and 2 from left to right follow directly from Definition 10 for stability: for a given justification status j (PRO or CONTRA) such that c is Stable-j given $\mathcal{C}$, it must be that each completion, including each completion in $\text{MC}(c)$ and $\text{MP}(c)$, has the status j given $\mathcal{C}$.

For item 1 from right to left, suppose that each c^* in $\text{MC}(c)$ is PRO given $\mathcal{C}$ and consider an arbitrary completion c' of c. By Definition 14, there must be some $c^* \in \text{MC}(c)$ such that $c^* \leq_{\text{PRO}} c'$. Since c^* is PRO given $\mathcal{C}$, there must be some $(c'', \text{PRO}) \in C$ such that $c'' \leq_{\text{PRO}} c^*$. By transitivity of $\leq_{\text{PRO}}$, $c'' \leq_{\text{PRO}} c'$, so c' is PRO given $\mathcal{C}$. Since c' was chosen arbitrarily, c must be Stable-PRO given $\mathcal{C}$. The proof for item 2 from right to left is analogous.

Finally consider item 3. The right to left part follows directly from Definition 10 for stability. For the proof from left to right, suppose that c is Unstable given $\mathcal{C}$. Then at least one of the following is true:

- There is some completion c_1 of $\mathcal{C}$ such that c_1 is PRO given $\mathcal{C}$ and there is some completion c_2 of $\mathcal{C}$ such that c_2 is not PRO given $\mathcal{C}$. As there must be some $c^* \in \text{MP}(c)$ such that $c_1 \leq_{\text{PRO}} c^*$, this c^* must be PRO given $\mathcal{C}$. Given that c_2 is not PRO and there must be some $c^- \in \text{MC}(c)$ such that $c^- \leq_{\text{PRO}} c_2$, this c^- cannot be PRO given $\mathcal{C}$. So the justification statuses of $c^* \in \text{MP}(c)$ and $c^- \in \text{MC}(c)$ are not equal.
- There is some completion c_1 of $\mathcal{C}$ such that c_1 is CONTRA given $\mathcal{C}$ and there is some completion c_2 of $\mathcal{C}$ such that c_2 is not CONTRA given $\mathcal{C}$. Analogously, there must be some $c^- \in \text{MC}(c)$ such that $c^- \leq_{\text{PRO}} c_1$, which must be CONTRA and some $c^* \in \text{MP}(c)$ such that $c_2 \leq_{\text{PRO}} c^*$, which cannot be CONTRA given $\mathcal{C}$. So $c^* \in \text{MP}(c)$ and $c^- \in \text{MC}(c)$ do not have equal justification statuses. □

Using this result, the stability status of an incomplete fact situation c can be computed in a more efficient way, since it suffices to only call the justification algorithm for the most extreme completions in $\text{MP}(c) \cup \text{MC}(c)$.

Example 13. *Reconsider the case base $\mathcal{C}$ and domain $\mathcal{D}$ from Example 11, but now suppose that for each $d_i \in \mathcal{D}$, $d_i = (\{1,\ldots,6\}, \leq'_{\text{PRO}}, \leq'_{\text{CONTRA}})$ where $\leq'_{\text{CONTRA}}$ is defined as in Example 12 (illustrated in Figure 2). Then the number of extreme completions $|MP(c)| + |MC(c)|$ of an incomplete focus case with all possible values for each dimension*

is $2^{|\mathcal{D}|} + 2^{|\mathcal{D}|}$, so the number of calls to the justification algorithm is greatly reduced compared to the $6^{|\mathcal{D}|}$ calls of the naive approach.

If the number of completions in $\text{MP}(c) \cup \text{MC}(c)$ is small compared to the total number of completions of c, then this stability algorithm is much faster than the naive version. In fact, if for each dimension d, the set $V_p(d, c)$ has a single minimum and a single maximum value, then the stability algorithm runs in polynomial time.

Proposition 5. *Given a domain $\mathcal{D}$ and some incomplete fact situation c, if for each $d \in \mathcal{D}$ the set $V_p(d, c)$ has a single minimum and a single maximum value, then the stability algorithm based on extreme completions runs in polynomial time for c.*

Proof. Suppose that for each $d \in \mathcal{D}$, the set $V_p(d, c)$ has a single minimum and a single maximum; then c has a single element in $\text{MP}(c)$ that is obtained by selecting the maximum possible value for each dimension. Similarly, $\text{MC}(c)$ has a single element, obtained by selecting the minimum possible values. Since $|\text{MP}(c)|+|\text{MC}(c)| = 2$, at most two calls to the polynomial algorithm for justification are required, so the stability algorithm based on extreme completions runs in polynomial time for c. □

Proposition 5 implies that computing stability is fast, even for very big inputs, if the possible values of each dimension have single minimum and maximum. This is, for example, always the case if the ordering on each of the dimensions is $\leq$ or $\geq$, like in our example in the web shop classification domain.

Example 14. *Consider in the web shop classification domain $\mathcal{D}$ the incomplete fact situation $c = (D, D^?)$ where: $D^? = \{(\texttt{trustmark}, \{0\}), (\texttt{terms}, \{0, 1\}), (\texttt{fake\_payment\_option}, \{0, 1\}), (\texttt{complaints}, \{6\})\}$. Then the set of most PRO completions of c contains a single element $D' = \{(\texttt{trustmark}, 0), (\texttt{terms}, 1), (\texttt{fake\_payment\_option}, 0), (\texttt{complaints}, 6)\}$. Similarly, the set of most CONTRA completions $MC(c)$ consists of a single element $D'' = \{(\texttt{trustmark}, 0), (\texttt{terms}, 0), (\texttt{fake\_payment\_option}, 1), (\texttt{complaints}, 6)\}$.*

However, the exponential blowup that occurs if the possible dimension values have multiple maxima or minima cannot be circumvented in general, because the problem of deciding if a case is Stable-PRO/Stable-CONTRA is CoNP-complete, as we prove next.

Proposition 6. *Deciding if an incomplete fact situation is Stable-PRO/Stable-CONTRA w.r.t. some case base is CoNP-complete.*

Proof sketch. Membership in CoNP follows from the fact that a negative instance can be verified in polynomial time, given a completion with a different justification status as a certificate. Next, we prove CoNP-hardness for deciding if an incomplete fact situation is Stable-PRO; the proof for Stable-CONTRA is analogous. We reduce from the CoNP-complete problem TAUTOLOGY. Let (ϕ, X) be an instance of this problem, where ϕ is a formula in DNF on the variables in X. Now let $(\mathcal{D}, C, c)$ be the result of a transformation $T(\phi, X)$ that is constructed as follows: for each $x_i \in X$, construct a dimension d_i with three values: $\{v_\text{p}, v_\text{n}, v_\text{u}\}$ and ordering $\leq_{\text{PRO}=} \{(v_\text{u}, v_\text{u}), (v_\text{u}, v_\text{p}), (v_\text{u}, v_\text{n}), (v_\text{p}, v_\text{p}), (v_\text{n}, v_\text{n})\}$ and let $\mathcal{D}$ be the set of all dimensions. Construct an incomplete case c such that for each $d_i \in \mathcal{D}$: $V_p(d_i, c) = \{v_\text{p}, v_\text{n}\}$. Construct case base $\mathcal{C}$ with a case (c^*, PRO) for each consistent clause c' in ϕ, where $v(d_i, c^*) = v_\text{p}$ if $x_i \in c'$; $v(d_i, c^*) = v_\text{n}$ if $\neg x_i \in c'$ and $v(d_i, c^*) = v_\text{u}$ otherwise. $T(\phi, X)$ can be constructed in polynomial time and (ϕ, X) is a positive instance of TAUTOLOGY iff c is Stable-PRO w.r.t. $\mathcal{C}$. □

To complete the complexity analysis of stability statuses, we claim that the problem of deciding if an incomplete fact situation is Unstable is NP-complete.

Proposition 7. *Deciding if an incomplete fact situation is Unstable w.r.t. some case base is NP-complete.*

Proof sketch. Analogous to the proof of Proposition 6, membership in NP follows from the fact that a positive instance can be verified in polynomial time given two completions with different justification statuses as certificates. For the NP-hardness proof, the transformation from Proposition 6 can be used to reduce from the NP-complete problem NON-TAUTOLOGY. □

To summarise this section, the tasks of deciding on the statuses PRO-stable, CONTRA-stable and Unstable are inherently complex. This could yield problems for applications with a lot of possible dimension values. We have provided an algorithm for deciding on these stability statuses in a relatively efficient way and proved that this algorithm is polynomial for specific inputs. In addition, we provided an exact polynomial algorithm for UNDECIDED-stability.

5.2 Computing relevance

In this section, we consider computational aspects of the relevance problem. First, we will study its complexity. In order to prove an upper bound on the complexity, we will use the property that relevant operations can be verified by a single completion c' of c for which the value v for d causes a difference in justification status: c has the justification status j, but replacing d's value by v results in a completion with a different justification status. Before formally proving this property in Lemma 1, we illustrate it with an example.

Example 15. *Reconsider from Example 8 the domain $\mathcal{D}$, case base $\mathcal{C}$ and incomplete fact situation $c = (D, D^?)$, where: $D^? = \{(\texttt{trustmark}, \{0\}), (\texttt{terms}, \{0, 1\}), (\texttt{fake\_payment\_option}, \{0, 1\}), (\texttt{complaints}, \{6\})\}$. The removal of 0 for* `fake_payment_option` *is CONTRA-relevant w.r.t. $\mathcal{C}$ and c. Indeed, c has a completion c' that is CONTRA, while replacing* `fake_payment_option`*'s value by 0 results in a completion that is not CONTRA: $c' = \{(\texttt{trustmark}, 0), (\texttt{terms}, 0), (\texttt{fake\_payment\_option}, 1), (\texttt{complaints}, 6)\}$.*

This property holds in general, as we show in Lemma 1.

Lemma 1. *Let $\mathcal{C}$ be a case base in domain $\mathcal{D}$ and let $c = (D, D^?)$ be an incomplete fact situation. Let j be a justification status (PRO, CONTRA or UNDECIDED). Then for a given dimension $d \in \mathcal{D}$, the removal of possible value $v \in V_p(d, c)$ is j-relevant w.r.t. $\mathcal{C}$ and c iff there is some completion $c' = D'$ of c such that:*

- *c' is j w.r.t. $\mathcal{C}$; and*
- *$c'' = D''$ is **not** j w.r.t. $\mathcal{C}$ where D'' is equal to D' except for the value of d, which is v. Formally: $D'' = \{(d', v(d', c')) \mid d' \in \mathcal{D} \setminus \{d\}\} \cup \{(d, v)\}$.*

Proof. First we prove the lemma from left to right:

(1) Suppose that the removal of v is j-relevant w.r.t. $\mathcal{C}$ and c.
(2) Then there is a minimal stable-j partial completion $c^* = (D^*, D^{?*})$ w.r.t. $\mathcal{C}$ and c such that $v \notin V_p(d, c^*)$.

(3) Now construct the partial completion $c^{*\prime}$ from c^* by including v to the possible values of d and replacing the current value of d by v: $c^{*\prime} = (D^{*\prime}, D^{?*\prime})$ where $D^{*\prime} = \{(d', v(d', c^*) \mid d' \in \mathcal{D} \setminus \{d\}\} \cup \{(d, v)\}$ and $D^{?*\prime} = \{(d', V_p(d', c^*)) \mid d' \in \mathcal{D} \setminus \{d\}\} \cup \{(d, V_p(d, c^*) \cup \{v\})\}$. Given that c^* was a minimal stable-j partial completion, that $c^* \neq c^{*\prime}$ and $c^* \in F(c^{*\prime})$, $c^{*\prime}$ cannot be stable-j w.r.t. $\mathcal{C}$.
(4) So there is some completion c'' of $c^{*\prime}$ that is not j w.r.t. $\mathcal{C}$. Then it must be that $v(d, c'') = v$: otherwise, c'' would also be a completion of $c^{*\prime}$ (and have the justification status j).
(5) Finally construct c', which equals c'' except for the value of d: $c' = \{(d', v(d', c'')) \mid d' \in \mathcal{D} \setminus \{d\}\} \cup \{(d, v(d, c^*))\}$. Then c' is a completion of c^*, so c' is j w.r.t. $\mathcal{C}$.

Then c' (in item 5) and c'' (in item 4) fulfill the left-to-right part of the lemma. Next, we prove the lemma from right to left:

(1) Suppose that there is some completion $c' = D'$ of c such that c' is j w.r.t. $\mathcal{C}$ and $c'' = D''$ is **not** j w.r.t. $\mathcal{C}$ where $D'' = \{(d', v(d', c')) \mid d' \in \mathcal{D} \setminus \{d\}\} \cup \{(d, v)\}$.
(2) Let $c^{*\prime}$ be an incomplete fact situation for which c' is the only completion as each dimension has one possible value: the actual value of that dimension in c'. Formally: $c^{*\prime} = (D', D^{?\prime})$ where $D^{?\prime} = \{(d, \{v'(d, c')\}) \mid d \in \mathcal{D}\}$. Then $c^{*\prime}$ is a partial completion of c (as c' is a completion of c) and it is stable-j w.r.t. $\mathcal{C}$ and $\mathcal{D}$ (as there is a single completion which is j).
(3) This implies that there must be some minimal stable-j partial completion c^* (possibly $c^{*\prime}$ itself) such that $c^{*\prime} \in F(c^*)$. Note that $v \notin V_p(d, c^*)$: if $v \in V_p(d, c^*)$ then c'' would be a completion of c^* and therefore be j w.r.t. $\mathcal{C}$, which would contradict our assumption. Then by Definition 13, removal of v is j-relevant w.r.t. $\mathcal{C}$ and c. □

Next, we use Lemma 1 to prove that relevance is in NP.

Proposition 8. *For each justification status j (PRO, CONTRA or UNDECIDED), deciding if the removal of a possible value for a given dimension of an incomplete fact situation is j-relevant w.r.t. some case base is NP-complete.*

Proof sketch. Membership in NP follows from Lemma 1: in order to verify that the removal of a value for a dimension d of an incomplete fact situation is PRO/CONTRA/UNDECIDED-relevant w.r.t. some case base (i.e. a positive instance of the problem), a suitable certificate would be some completion c' such that c' is j w.r.t. the case base, while replacing d's value to v results in a completion c'' that is not j. This can be validated in polynomial time using the algorithm for justification on both c' and c''. Together with the NP-hardness proof, which we omit here due to space restrictions, this implies that deciding relevance is NP-complete. □

Having identified the complexity of the relevance problem, we propose an algorithm for enumerating all PRO-and CONTRA-relevant removals, given an incomplete fact situation c and case base $\mathcal{C}$ in domain $\mathcal{D}$. Informally, this algorithm works as follows:

(1) For each case $(c', o) \in \mathcal{C}$, construct the completions of c that "just suffice to match" c' and collect these cases in an alternative case base $\mathcal{C}^*$.
(2) Let $\mathcal{C}'$ be a minimised version of $\mathcal{C}^*$, where cases forced by another case in $\mathcal{C}'$ are removed.
(3) For each case (c', o) in $\mathcal{C}'$, for each $d \in \mathcal{D}$ and for each $v \in V_p(d, c)$ such that $v \not\geq_o v(d, c')$: construct an alternative case c'' by replacing the value of d in c' by v. Store v as an o-relevant removal if the justification status is no longer o.

To give some intuition on this procedure: constructing $\mathcal{C}'$ is convenient as it allows us to efficiently find the completions c' and c'' mentioned in Lemma 1, where the justification status switches between PRO/CONTRA and an alternative status. Next, we formally define the minimal matching completions and minimised case base.

Definition 15 (Minimal matching values and completions). *Given an incomplete fact situation c, case (c', o) and d in domain $\mathcal{D}$, the set of minimal matching values is $V_m(d, (c', o))$, which consists of all values v in $V_p(d, c)$ such that $v \geq_o v(d, c')$ and there is no $v' \in V_p(d, c)$ such that $v \geq_o v', v' \not\geq_o v$ and $v' \geq_o v(d, c')$. The set of minimal matching completions consists of all combinations of minimal matching values for each dimension: $C_m(c') = \bigtimes_{d \in \mathcal{D}} V_m(d, c')$.*

Definition 16 (Minimised case base). *Given a case base $\mathcal{C}$, the minimised version is $\mathtt{min}(\mathcal{C})$ where $\mathtt{min}(\mathcal{C}) = \{(c', o) \in \mathcal{C} \mid$ there is no c'' such that $(c'', o) \in \mathcal{C}, c'' \leq_o c'$ and $c' \not\leq_o c''\}$.*

Example 16. *Reconsider from Example 8 the domain $\mathcal{D}$, case base $\mathcal{C}$ and incomplete fact situation $c = (D, D^?)$, where: $D^? = \{(\mathtt{trustmark}, \{0\}), (\mathtt{terms}, \{0, 1\}), (\mathtt{fake\_payment\_option}, \{0, 1\}), (\mathtt{complaints}, \{6\})\}$. For c_1, there is no minimal matching completion. For c_2, the only minimal matching completion is $c' = \{(\mathtt{trustmark}, 0), (\mathtt{terms}, 0), (\mathtt{fake\_payment\_option}, 1), (\mathtt{complaints}, 6)\}$. For c_3, the minimal matching completion is $c'' = \{(\mathtt{trustmark}, 0), (\mathtt{terms}, 1), (\mathtt{fake\_payment\_option}, 1), (\mathtt{complaints}, 6)\}$. The minimised version of $\{(c', \mathit{CONTRA}), (c'', \mathit{CONTRA})\}$ is $C' = \{(c'', \mathit{CONTRA})\}$.*

Then the remainder of the procedure to obtain PRO- and CONTRA-relevant updates from this minimal case base $\mathcal{C}'$ is as follows:

(1) For each $(c', \text{PRO}) \in \mathcal{C}'$, for each $d \in \mathcal{D}$ and for each $v \in V_p(d, c)$ such that $v \not\geq_{\text{PRO}} v(d, c')$:
   (a) Construct c'' by replacing the value of d in c' by v.
   (b) If the justification status of c'' is not PRO, then removal of v is PRO-relevant.
(2) For each $(c', \text{CONTRA}) \in \mathcal{C}'$, for each $d \in \mathcal{D}$ and for each $v \in V_p(d, c)$ such that $v \not\geq_{\text{CONTRA}} v(d, c')$:
   (a) Construct c'' by replacing the value of d in c' by v.
   (b) If the justification status of c'' is not CONTRA then removal of v is CONTRA-relevant.

Example 17. *Building on Example 16, given $C' = \{(c'', \mathit{CONTRA})\}$, for CONTRA-relevance we only check $\{(\mathtt{trustmark}, 0), (\mathtt{terms}, 1), (\mathtt{fake\_payment\_option}, 0), (\mathtt{complaints}, 6)\}$. This is UNDECIDED, so removal of value 0 for* `fake_payment_option` *is CONTRA-relevant.*

Proposition 9. *Within some domain $\mathcal{D}$, let $\mathcal{C}$ be a case base and let c be an incomplete focus case. The algorithm explained above, applied on c, $\mathcal{D}$ and $\mathcal{C}$, returns for each dimension $d \in \mathcal{D}$ all PRO- and CONTRA-relevant values to remove.*

Proof sketch. Given a case base $\mathcal{C}$ and incomplete focus case c, let $\mathcal{C}^*$ be the set of cases (c', o) where $c' \in C_m(c'')$ for some $(c'', o) \in \mathcal{C}$. Let $\mathcal{C}' = \mathtt{min}(\mathcal{C}^*)$. For each completion of c, its justification status given $\mathcal{C}$ equals its justification status given $\mathcal{C}'$, thanks

to the way $\mathcal{C}'$ is constructed. Then by Lemma 1, for any given dimension $d \in \mathcal{D}$ and $j \in \{\text{PRO}, \text{CONTRA}\}$, the removal of possible value $v \in V_p(d, c)$ is j-relevant w.r.t. $\mathcal{C}$ and c iff there is some completion $c' = D'$ of c such that c' is j w.r.t. $\mathcal{C}'$; and $c'' = \{(d', v(d', c')) \mid d' \in \mathcal{D} \setminus \{d\}\} \cup \{(d, v)\}$ is not j w.r.t. $\mathcal{C}'$. This is the case iff there is some $(c^*, j) \in \mathcal{C}'$ such that $c^{*\prime} = \{(d', v(d', c^*)) \mid d' \in \mathcal{D} \setminus \{d\}\} \cup \{(d, v)\}$ is not j w.r.t. $\mathcal{C}'$: from right to left this follows immediately, as c^* must be a completion of c. From left to right: since c' is j w.r.t. $\mathcal{C}'$, there must be some $(c^*, j) \in \mathcal{C}'$ such that $c' \geq_{\text{PRO}} c^*$. Let $c^{*\prime} = \{(d', v(d', c^*)) \mid d' \in \mathcal{D} \setminus \{d\}\} \cup \{(d, v)\}$; then $c'' \geq_{\text{PRO}} c^{*\prime}$, so $c^{*\prime}$ cannot be j. Therefore all j-relevant updates are returned. □

Similar to the stability algorithm from Section 5.1, the proposed algorithm has a worst-case exponential runtime but runs in polynomial time under specific conditions on the dimension ordering.

Proposition 10. *If for each dimension d of an incomplete fact situation c the set $V_p(d, c)$ has a single minimum and a single maximum value, then the relevance algorithm runs in polynomial time.*

Proof sketch. Under these conditions, there is at most one minimal matching value for each dimension for any case in $\mathcal{C}$, so each case in $\mathcal{C}$ has a single minimal matching completion. □

In this section, we have proposed efficient algorithms for the problems of justification, stability and relevance. Although some of these problems are in high complexity classes, there are conditions on the dimensions for which the algorithms run in polynomial time.

6 RELATED WORK

The research area of reasoning with legal cases originated from Rissland and Ashley's work on the HYPO system for US trade secret's law [17]. Compared to HYPO, our approach differs in, for instance, the evaluation of cases, the selection of the citeable case(s) and the strategy for generating hypothetical variations. Whereas HYPO focused on generating debates as they can take place between lawyers, we focus on classifying the outcome of legal cases, in particular those cases for which the outcome is forced by a precedent. Related work in this area includes [3, 16]. In particular, we are interested in the models for precedential constraint proposed by Horty [8, 9], especially the result model. Whereas this result model has been applied for explaining outcomes of machine-learning-based decision-making applications [15], we propose an alternative application for human-in-the-loop decision support. Moreover, whereas the models for precedential constraint in [8, 9] are defined on cases that either contain only factors or only dimensions, in our work we also consider cases that combine factors and dimensions in a similar way as [15]. An alternative way of combining factors and dimensions in a single model of precedential constraint is presented in [14].

Another area of related research is factor ascription [6], which provides argumentation schemes for the presence of factors in a case. This could be used for instantiating possible dimension values.

Finally, the idea of defining and computing a stability status and relevance notion, based on possible future information updates, was developed in other settings as well. Our definitions for stability and relevance are inspired by the definitions presented for argumentation-based reasoning in [11, 12].

7 CONCLUSION

We defined and studied the notions of justification, stability and relevance for Horty's result model of precedential constraint. In contrast to earlier work on this model, we do not assume that all factors and dimension value assignments of the focus case have been established with certainty. In order to account for this, we have introduced the notion of incomplete fact situations. We have defined an incomplete fact situation as *stable* if its justification status, that is, the enforced outcome by the precedential constraint, does not change, regardless of any change in the uncertain elements. For incomplete fact situations that are not stable, we have studied the task of identifying *relevance*: which updates on dimensions can be performed to make the resulting incomplete fact situation stable? We have described how these tasks can be exploited in a human-in-the-loop decision support system: if an incomplete fact situation is stable, then further investigation is not required. Otherwise, relevant updates reveal which dimensions should be investigated further. Finally, we showed that the tasks of identifying the justification and stability status of a focus case can be performed efficiently, just like the task of identifying relevant uncertainties.

REFERENCES

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

[2] Kevin Ashley. 1991. Reasoning with cases and hypotheticals in HYPO. *International journal of man-machine studies* 34, 6 (1991), 753–796.

[3] Kevin Ashley and Stefanie Brüninghaus. 2009. Automatically classifying case texts and predicting outcomes. *Artificial Intelligence and Law* 17, 2 (2009), 125–165.

[4] Kevin Ashley and Edwina Rissland. 1988. A case-based approach to modeling legal expertise. *IEEE Intelligent Systems* 3 (1988), 70–77.

[5] Trevor Bench-Capon. 2017. Hypo's legacy: introduction to the virtual special issue. *Artificial Intelligence and Law* 25, 2 (2017), 205–250.

[6] Trevor Bench-Capon and Katie Atkinson. 2022. Argument Schemes for Factor Ascription. In *Computational Models of Argument. Proceedings of COMMA 2022.* 68 – 79.

[7] Guido Governatori, Francesco Olivieri, Antonino Rotolo, and Matteo Cristani. 2022. Stable Normative Explanations. In *Legal Knowledge and Information Systems.* IOS Press, 43–52.

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

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

[10] Daphne Odekerken and Floris Bex. 2020. Towards Transparent Human-in-the-Loop Classification of Fraudulent Web Shops. In *Proceedings of the 33rd International Conference on Legal Knowledge and Information Systems*, Serena Villata, Jakub Harasta, and Petr Kremen (Eds.). IOS Press, 239–242.

[11] Daphne Odekerken, Floris Bex, AnneMarie Borg, and Bas Testerink. 2022. Approximating Stability for Applied Argument-based Inquiry. *Intelligent Systems with Applications* (2022), 200110.

[12] Daphne Odekerken, AnneMarie Borg, and Floris Bex. 2022. Stability and Relevance in Incomplete Argumentation Frameworks. In *Computational Models of Argument. Proceedings of COMMA 2022*, Francesca Toni, Sylwia Polberg, Richard Booth, Martin Caminada, and Hiroyuki Kido (Eds.). 272 – 283.

[13] Joeri Peters, Floris Bex, and Henry Prakken. 2022. Justifications derived from inconsistent case bases using authoritativeness. In *Proceedings of the 1st International Workshop on Argumentation for eXplainable AI*, Vol. 3209. CEUR WS, 1–13.

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

[15] Henry Prakken and Rosa Ratsma. 2022. A Top-level Model of Case-based Argumentation for Explanation: Formalisation and Experiments. *Argument & Computation* 13, 2 (2022), 159–194.

[16] Henry Prakken and Giovanni Sartor. 1998. Modelling reasoning with precedents in a formal dialogue game. *Artificial Intelligence and Law* 6 (1998), 231–287.

[17] Edwina Rissland and Kevin Ashley. 1987. A case-based system for trade secrets law. In *Proceedings of the 1st international conference on Artificial intelligence and law.* 60–66.