International Journal of Approximate Reasoning 152 (2023) 357–389

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

The possibilistic horn non-clausal knowledge bases

Gonzalo E. Imaz

Artificial Intelligence Research Institute (IIIA) - CSIC, Barcelona, Spain

ARTICLE INFO

Article history:
Received 9 November 2021
Received in revised form 3 November 2022
Accepted 3 November 2022
Available online 10 November 2022

Keywords:
Possibilistic logic
Horn
Non-clausal
Inconsistency
Resolution
Satisfiability testing

ABSTRACT

Non-clausal deduction in classical logic is one of the oldest areas in artificial intelligence. It first appeared in the sixties and consequently a large body of research has been devoted to it. Within the last decades, computing with non-clausal formulas has been considered in several fields, and in particular, in answer set programming, wherein non-clausal or nested logic programs were conceived in 1999.

Possibilistic logic is the most extended approach to handle uncertain and partially inconsistent information. Here, we generalize some well-known clausal outcomes in possibilistic reasoning to the non-clausal setting, concretely the objective of our proposal is: (i) to extend available insights from clausal to non-clausal form; (ii) to show that possibilistic reasoning admits feasible classes also at the non-clausal level; (iii) to combine the high expressiveness of non-clausal possibilistic logic with the highest efficient (polynomial) reasoning mechanisms; and (iii) to suggest that some meaningful subclasses of possibilistic nested programs can be efficiently processed.

Firstly, we define the class of *Possibilistic Horn Non-Clausal formulas*, or $\overline{\mathcal{H}}_\Sigma$, which covers the classes: possibilistic Horn and propositional Horn-NC. $\overline{\mathcal{H}}_\Sigma$ is shown to be non-clausal, analogous to the standard Horn class.

Secondly, we define *Possibilistic Non-Clausal Unit-Resolution*, or $\mathcal{UR}_\Sigma$, and prove that $\mathcal{UR}_\Sigma$ correctly computes the inconsistency degree of Horn-NC bases. $\mathcal{UR}_\Sigma$ is formulated in a clausal-like manner, which eases its understanding, formal proofs and future extension towards full non-clausal resolution.

Thirdly, we prove that computing the inconsistency degree of Horn-NC bases takes polynomial time. Although there already exist tractable classes in possibilistic logic, all of them are clausal, and thus, $\overline{\mathcal{H}}_\Sigma$ turns out to be the first characterized polynomial non-clausal class within possibilistic reasoning.

We discuss that our approach serves as a starting point to developing uncertain non-clausal reasoning on the basis of both methodologies: DPLL and resolution.

1. Introduction

Possibilistic logic is the most popular approach to represent and reason with uncertain and partially inconsistent knowledge, and the vast research carried out for several decades in the field has led to numerous theoretical discoveries and pragmatic progress. Regarding normal forms, important research has been developed and notable improvements have been accomplished in the standard clausal form, however little effort has been devoted so far to possibilistic reasoning in non-clausal form, on which our contributions are centered.

E-mail address: gonzalo@iiia.csic.es.

https://doi.org/10.1016/j.ijar.2022.11.002
0888-613X/

We emphasize, however, that our approach is not aimed at being an alternative to methods or developments based on clausal form. On the contrary, it is complementary to them, showing that some nice properties enjoyed by clausal possibilistic reasoning are generalizable to the non-clausal level. Simultaneously, we suggest that non-clausal possibilistic reasoning is, despite its scientific and technological interest, seldom studied, and so, it remains an open domain wherein research efforts are worthwhile.

Our new results, as aforesaid, stem from recognized clausal ones, and specifically, we lift to the non-clausal level: the Horn possibilistic class as well as possibilistic unit-resolution. More specifically, this means that we: (i) characterize the possibilistic Horn non-clausal (Horn-NC) bases and prove that obtaining their inconsistency degree is polynomial; and (ii) design possibilistic non-clausal unit-resolution and demonstrate its correctness for calculating the inconsistency degree of Horn-NC bases.

Reasoning in classical logic and non-clausal form is one of the oldest fields in artificial intelligence, as it took off in the 1960's, and is an actual concern in the principal fields of classical logic, namely, satisfiability solving [81,73,59,43], theorem proving [40,72], quantified boolean formulas [39,15] and answer set programming [19,16]; and also in many other automated reasoning areas such as symbolic model checking [83], formal verification [27], supervisory control [18], circuit verification [26], knowledge compilation [23], heuristics [7], encodings [66], deductive databases [24], linear constraints [80], constraint handling rules [45], dynamic systems [76], constrained Horn clauses [56], Horn clause verification [82], planning [77], stochastic search [63] and hard problems [66].

We stress the relevance that processing with non-clausal formulas has in the field of answer set programming [6,60], a current and prominent problem-solving artificial intelligence methodology. In fact, answer sets were enriched to non-clausal or nested programs by Lifschitz et al. [61] in 1999, and since then, they have been determining in many theoretical issues and practical applications. It is worthwhile mentioning here that the authors in [68,69] extended the nested programs to possibilistic logic and, moreover, showed their practical utility by successfully solving real-world problems from the medical domain. Besides, the approach in [68,69] is, to the best of our knowledge, the only existing one to deal with possibilistic non-clausal formulas.

Switching to non-classical logic, non-clausal formulas with different functionalities have been studied in a profusion of languages: signed many-valued logic [65,10,79], Łukasiewicz logic [58], Levesque's three-valued logic [17], Belnap's four-valued logic [17], M3 logic [1], fuzzy logic [48], fuzzy description logic [47], intuitionistic logic [74], modal logic [74], lattice-valued logic [84] and regular many-valued logic [54].

When the formula issued from modeling a real problem is naturally expressed in non-clausal form, technical drawbacks summarized below make it inadvisable to employ clausal-form translators and so advocate computing the formula in its original non-clausal structure. The drawbacks below have been gathered by many authors across several fields, e.g. the authors in [38,78,39] report them within the area of quantified boolean formulas.

Two kinds of clausal form transformation are known. The first one, applying distributivity to the non-clausal formulas until obtaining an equivalent clausal formula, is clearly infeasible as it leads to an exponential increase of the size of the clausal formula.

The second one, Tseitin-transformation, usually produces an increase of formula size and number of variables, and also a loss of information about the formula's original structure. Besides, the normal form is not unique in most cases and, however, deciding how to perform the transformation heavily influences the solving process and it is usually impossible to predict which strategy is going to be the best, as this depends on the concrete solver used and on the kind of problem to be solved. Further, Tseitin-transformation preserves satisfiability but loses logical equivalence, which impedes its usage in many applications.

In view of such drawbacks, we abandon the assumption that the input formula should be transformed to clausal form and directly process it in its original nested structure. We allow an arbitrary nesting of conjunctions and disjunctions and only limit the scope of the negation connective that applies to exclusively literals. The non-clausal form considered here is popularly called negation normal form (NNF) and can be obtained deterministically, causing only a negligible increase of the formula size.

The second cornerstone of our proposal is the class of Horn clausal formulas. Horn formulas are recognized as central for deductive databases, declarative programming, and more generally, for rule-based systems. In fact, Horn formulas have received a great deal of attention since 1943 [62,51] and, at present, there is a broad span of areas within artificial intelligence relying on them, and their scope covers a fairly large spectrum of realms spread across many logics and a variety of reasoning settings (see [52] for details).

Computing the inconsistency degree of possibilistic Horn formulas is a tractable problem [57] and Section 5 states that $O(n \times \log m)$ is its best worst-case complexity known up to now. Related to this class but going beyond clausal form, we present the novel possibilistic class Horn-NC that is non-clausal and which we will denote by $\overline{\mathcal{H}}_\Sigma$. We will demonstrate that $\overline{\mathcal{H}}_\Sigma$ is non-clausal, analogous to the Horn clausal class, and that besides the latter, $\overline{\mathcal{H}}_\Sigma$ also subsumes the class of propositional Horn-NC formulas which have been previously characterized in propositional [52] and regular many-valued [54] logics.

Computationally, we will prove that determining the inconsistency degree of Horn-NC bases is a polynomial problem. This result signifies that polynomiality is preserved when upgrading both from clausal to non-clausal form and from propositional to possibilistic logic: in the former upgrading, because the possibilistic Horn [57] and possibilistic Horn-NC classes are both tractable, and in the latter, because the propositional Horn-NC [52] and possibilistic Horn-NC classes are both tractable as well.

Firstly, we determine the syntactical Horn-NC restriction by lifting the Horn clausal restriction "a formula is Horn if all its clauses have any number of negative literals and at most one positive literal", to the non-clausal level as follows: *"a propositional NC formula is Horn-NC if all its disjunctions have any number of negative disjuncts and at most one non-negative disjunct"*. By extending such definition to possibilistic logic, we establish straightforwardly that a possibilistic NC formula is Horn-NC only if its propositional formula is Horn-NC, and denote the *class of possibilistic Horn-NC formulas* by, as said above, $\overline{\mathcal{H}}_\Sigma$. Note that $\overline{\mathcal{H}}_\Sigma$ naturally subsumes the standard possibilistic Horn clausal class.

Example 1.1. Below we give a specific possibilistic non-clausal base Σ expressed in suffix notation (explained in detail in Section 2). $P, Q, \ldots$ and $\neg P, \neg Q, \ldots$ are positive and negative literals, respect., and ϕ_1, ϕ_2 and ϕ_3 are NC propositional formulas. $(\vee_k\ \varphi_1 \ldots \varphi_k)$ and $[\wedge_k\ \varphi_1 \ldots \varphi_k]$ express the disjunction and conjunction, respect., of formulas φ_1 to φ_k

$$\varphi = [\wedge_3\ P\ (\vee_2\ \neg Q\ [\wedge_3\ (\vee\ \neg P\ \neg Q\ R\,)\ (\vee_2\ \phi_1\ [\wedge\ \phi_2\ \neg P\,]\,)\ Q\,]\,)\ \phi_3\,]$$

$$\Sigma = \{\,\langle \varphi : \mathbf{0.8}\rangle\quad \langle P : \mathbf{0.8}\rangle\quad \langle \neg Q : \mathbf{0.6}\rangle\quad \langle R : \mathbf{0.6}\rangle\quad \langle \phi_1 : \mathbf{0.3}\rangle\quad \langle \phi_3 : \mathbf{1.0}\rangle\,]$$

and subindex k in connectives $\vee_k$ and $\wedge_k$ indicates their arity. We will show that Σ is Horn-NC only if ϕ_1, ϕ_2 and ϕ_3 are Horn-NC and at least one of ϕ_1 or ϕ_2 is negative.

We will demonstrate the following relationships of $\overline{\mathcal{H}}_\Sigma$ with (1) its subclass $\mathcal{H}_\Sigma$ of Horn-clausal formulas and with (2) its superclass of non-clausal formulas: (1) $\overline{\mathcal{H}}_\Sigma$ subsumes syntactically $\mathcal{H}_\Sigma$ but both classes are semantically equivalent; and (2) $\overline{\mathcal{H}}_\Sigma$ contains all non-clausal formulas whose clausal form is Horn. In view of (1) and (2), one can conclude that $\overline{\mathcal{H}}_\Sigma$ is non-clausal, analogous to the standard Horn class.

Secondly, we establish the calculus *Possibilistic Non-Clausal Unit-Resolution*, or $\mathcal{UR}_\Sigma$, and then prove that it correctly computes the inconsistency degree of the Horn-NC bases. $\mathcal{UR}_\Sigma$ is the generalization of non-clausal unit-resolution from propositional [52] to possibilistic logic and we formulate it here in a clausal-like fashion, which contrasts with the functional-like fashion of the existing non-clausal (full) resolution [64].

Thirdly, we prove that computing the inconsistency degree of Horn-NC bases has polynomial complexity. There indeed exist polynomial classes in possibilistic logic but all of them are clausal [57], and so, the tractable non-clausal fragment was empty so far. We think that this is just a first tractable result in possibilistic non-clausal reasoning and that the approach presented here will serve as a starting point to find further classes, and so to widen the tractable possibilistic non-clausal fragment.

Summing up, the list of properties of $\overline{\mathcal{H}}_\Sigma$ is given below (the last two properties have been shown in [52] for propositional logic but are preserved in $\overline{\mathcal{H}}_\Sigma$), where $\mathcal{H}_\Sigma$ denotes the classical possibilistic Horn clausal class:

- Computing the inconsistency-degree of $\overline{\mathcal{H}}_\Sigma$ bases is tractable.
- $\overline{\mathcal{H}}_\Sigma$ subsumes syntactically $\mathcal{H}_\Sigma$.
- $\overline{\mathcal{H}}_\Sigma$ and $\mathcal{H}_\Sigma$ are semantically equivalent.
- $\overline{\mathcal{H}}_\Sigma$ contains all non-clausal formulas whose clausal form is Horn.
- $\overline{\mathcal{H}}_\Sigma$ is linearly recognizable [52].
- $\overline{\mathcal{H}}_\Sigma$ is strictly more succinct[1] than $\mathcal{H}_\Sigma$ [52].

Given that Horn formulas are suitable for many applications as pointed out above, Horn-NC formulas are potentially useful for generalizing such applications. We outline the potential applicability of Horn-NC formulas in nested logic programming [61] as follows.

Just as the Horn-clausal class underpins standard definite (heads are literals) logic programming [14], it can be proven [53] (ongoing work) that the Horn-NC class underpins definite nested logic programming. In a nutshell, Horn-NC programs are definite nested programs, i.e. they are the non-clausal counterpart of definite or Horn clausal programs. In fact, it is proven in [53] that Horn-NC programs also enjoy many recognized properties of Horn programs, for instance, Horn-NC programs have only one minimal model.

In order to illustrate the utility of the Horn-NC class in the nested logic programming field, we provide an example below. Further examples can be found in Appendix (and also in [53]). As nested programs are up to an exponential factor more succinct than their equivalent unnested programs, their interpreter requires to evaluate substantially less literals and connectives, and so it achieves higher efficiency. The extension of $\overline{\mathcal{H}}_\Sigma$ to possibilistic nested logic programming is briefly discussed also in Appendix. Yet, as can be seen there, it requires additional research that will be reported in a forthcoming work.

[1] Succinctness was defined in [46].

Example 1.2. The twelve traditional definite rules in the first two lines are equivalent to the Horn-NC rule in the third line. This nested rule is Horn-NC because, as it will be shown, its nested head and the whole rule are both Horn-NC formulas. Note the contrast between the 33 literals in the twelve non-nested rules and the 9 literals plus the constant $\top$ in the unique nested rule; besides in this particular case, no literal repetition is needed.

$$b \leftarrow a \quad b \leftarrow c \quad b \leftarrow m \quad c \leftarrow a, b \quad c \leftarrow e, b \quad c \leftarrow m, b$$

$$g \leftarrow a, d \quad g \leftarrow e, d \quad g \leftarrow m, d \quad g \leftarrow a, f \quad g \leftarrow e, f \quad g \leftarrow m, f$$

$$(\vee_2\ [\wedge\ \neg a\ \neg e\ \neg m]\ [\wedge_3\ (\vee\ \neg b\ c)\ b\ (\vee_2\ g\ [\wedge\ \neg d\ \neg f])\,]\,) \leftarrow \top$$

In the context of answer set programming, the first two lines would represent a classical reduct, while the second line would correspond to its equivalent nested reduct. □

Altogether, Horn-NC or definite nested programs can compact traditional definite programs up to an exponential factor while preserving their nice semantical properties (proven in [53]), and besides, Horn-NC programs: (1) mitigate redundancies inherent to traditional logic programs; and (2) form a highly flexible and expressive logic programming language, which helps users to more naturally represent their expertise.

Possibilistic logic can be seen as an encapsulation of propositional logic. Thus the inconsistency degree of Horn-NC, and in general of non-clausal bases, can be calculated through a number (at most logarithmic) of calls to a non-clausal SAT solver or through non-clausal propositional queries. Hence, this principle remains sufficiently general to take advantage of any potential progress made at the level of non-clausal propositional logic.

The presented approach serves as a base to develop approximate non-clausal reasoning based on the DPLL and resolution schemes: (1) it paves the way to define DPLL in non-clausal form as its procedure, unit-propagation for non-clausal formulas, is based on $\mathcal{UR}_\Sigma$; and (2) the existing non-clausal resolution [64] presents some deficiencies derived from its functional-like formalization, such as not precisely defining the potential resolvents. The given clausal-like formalization of $\mathcal{UR}_\Sigma$ skips such deficiencies and signifies a step forward towards defining non-clausal resolution for at least those uncertainty logics for which clausal resolution is already defined, e.g. possibilistic logic [30,31].

This paper is organized as follows. Section 2 and 3 present background on propositional non-clausal formulas and on possibilistic logic, respectively. Section 4 defines the class $\overline{\mathcal{H}}_\Sigma$. Section 5 introduces the calculus $\mathcal{UR}_\Sigma$. Section 6 provides examples illustrating how $\mathcal{UR}_\Sigma$ computes Horn-NC bases. Section 7 provides the formal proofs. Section 8 is devoted to both related and future work. Last section summarizes the main contributions.

A supplied appendix includes further examples of the equivalence between Horn and Horn-NC programs, and also, discusses the relationships between Horn-NC programs and both answer sets with aggregates [41] and nested programs with preferences [19].

2. Propositional non-clausal logic

This section presents the terminologies used in this paper and background on non-clausal (NC) propositional logic (see [11] for a complete background).

The NC language is formed by: constants $\{\bot, \top\}$, propositions $\mathcal{P} = \{P, Q, \ldots\}$, connectives $\{\neg, \vee, \wedge\}$ and auxiliary symbols "(", ")", "[", and "]". $X \in \mathcal{P}$ (resp. $\neg X$) is a positive (resp. negative) literal. $\mathcal{L}$ is the set of literals. $\bot, \top$ and the literals are atoms. $(\vee\ \ell_1 \ldots \ell_k)$, with $\ell_i \in \mathcal{L}$, is a clause. A clause with at most one positive literal is Horn. $[\wedge\ C_1 \ldots\ C_n]$, the C_i's being clauses (resp. Horn clauses) is a clausal (resp. Horn) formula.

Note. For the sake of readability of non-clausal formulas, we will employ: (1) prefix notation as it requires only one $\vee/\wedge$-connective per formula, while infix notation requires $k-1$, k being the arity of $\vee/\wedge$; (2) two formula delimiters, $(\vee_k\ \varphi_1 \ldots \varphi_k)$ for disjunctions and $[\wedge_k\ \varphi_1 \ldots \varphi_k]$ for conjunctions, to better distinguish them inside non-clausal formulas.

Our first definition is that of non-clausal formulas, whose differential feature is that the connective $\neg$ can only occur in front of propositions, i.e. at atomic level.

Definition 2.1. $\mathcal{NC}$ is the smallest set such that the following conditions hold:

- $\{\bot, \top\} \cup \mathcal{L} \subset \mathcal{NC}$.
- If $\varphi_1, \ldots, \varphi_k \in \mathcal{NC}$ then $[\wedge_k\ \varphi_1 \ldots \varphi_i \ldots \varphi_k] \in \mathcal{NC}$.
- If $\varphi_1, \ldots, \varphi_k \in \mathcal{NC}$ then $(\vee_k\ \varphi_1 \ldots \varphi_i \ldots \varphi_k) \in \mathcal{NC}$.

We will omit the subindex k in the connectives when $\varphi_1 \ldots \varphi_k$ are exclusively literals.

Example 2.2. φ_1 and φ_2 below are NC formulas, while $\neg(\vee\ \varphi_1\ \varphi_2)$ is not.

- $\varphi_1 = [\wedge_2\ \ (\vee\ \ \neg P\ \ Q\ \ \bot)\ \ (\vee_2\ \ Q\ \ [\wedge\ \neg R\ \ S\ \ \top])]$
- $\varphi_2 = (\vee_3\ \ [\wedge\ \ \neg P\ \ \top]\ \ [\wedge_2\ \ (\vee\ \ \neg P\ \ R)\ \ [\wedge_2\ \ Q\ \ (\vee\ \ P\ \ \neg S)]]\ \ [\wedge\ \ \bot\ \ Q])$ □

Definition 2.3. The unique sub-formula of an atom is the atom itself. The sub-formulas of $\varphi = [\odot_k\ \varphi_1 \ldots \varphi_i \ldots \varphi_k], \odot_k \in \{\wedge_k, \vee_k\}$, are φ itself plus the sub-formulas of the φ_i's.

Definition 2.4. NC formulas are modeled by trees in the following way: (i) each atom is a *leaf* and each occurrence of a connective is an *internal node*; and (ii) each sub-formula $[\odot_k\ \varphi_1 \ldots \varphi_k], \odot_k \in \{\wedge_k, \vee_k\}$ is a k-ary hyper-arc linking the node of $\odot_k$ with, for every i, the node of φ_i if φ_i is an atom and with the node of its connective otherwise.

Remark. Our approach also applies when NC formulas are represented and implemented by DAGs. Nevertheless, for simplicity, we will use only formulas represented by trees.

Definition 2.5. An interpretation ω maps $\mathcal{NC}$ into $\{0, 1\}$ and is extended from $\{\bot, \top\} \cup \mathcal{P}$ to $\mathcal{NC}$ via the rules below, where $X \in \mathcal{P}$ and $\varphi_i \in \mathcal{NC}$.

- $\omega(\bot) = \omega((\vee)) = 0$ and $\omega(\top) = \omega([\wedge]) = 1$.
- $\omega(X) + \omega(\neg X) = 1$.
- $\omega((\vee_k\ \varphi_1 \ldots \varphi_i \ldots \varphi_k)) = 1$ iff $\exists i$, $\omega(\varphi_i) = 1$.
- $\omega([\wedge_k\ \varphi_1 \ldots \varphi_i \ldots \varphi_k]) = 1$ iff $\forall i$, $\omega(\varphi_i) = 1$.

Definition 2.6. Let $\varphi, \varphi' \in \mathcal{NC}$. An interpretation ω is a model of φ if $\omega(\varphi) = 1$. If φ has a model then it is consistent and otherwise inconsistent. φ and φ' are equivalent, or $\varphi \equiv \varphi'$, if $\forall \omega$, $\omega(\varphi) = \omega(\varphi')$. φ' is consequence of φ, or $\varphi \models \varphi'$, if $\forall \omega$, $\omega(\varphi) \leq \omega(\varphi')$.

Trivially $\bot \equiv (\vee)$ and $\top \equiv [\wedge]$. Formulas can be simplified by recursively using the equivalences: $[\wedge\ \top\ \varphi] \equiv \varphi \equiv (\vee\ \bot\ \varphi)$, $(\vee\ \top\ \varphi) \equiv \top$, and $[\wedge\ \bot\ \varphi] \equiv \bot$.

Note. For simplicity and since constants are easily removed as indicated above, the unique formula with constants handled in this paper will be $\bot$.

3. Necessity-valued possibilistic logic

Let us have a brief refresher on necessity-valued possibilistic logic [29,32,33].

3.1. Semantics

At the semantic level, possibilistic logic is defined in terms of a *possibilistic distribution* π on the universe Ω of interpretations, i.e. an $\Omega \to [0, 1]$ encodes for each $\omega \in \Omega$ to what extent it is plausible that ω is the actual world. $\pi(\omega) = 0$ means that ω is impossible, $\pi(\omega) = 1$ means that nothing prevents ω from being true, whereas $0 < \pi(\omega) < 1$ means that ω is only somewhat possible to be the real world. Possibility degrees are interpreted qualitatively: when $\pi(\omega) > \pi(\omega')$, ω is considered more plausible than ω'. A possibilistic distribution π is *normalized* if $\exists \omega \in \Omega, \pi(\omega) = 1$, i.e. at least one interpretation is plausible.

A possibility distribution π induces two uncertainty functions from $\mathcal{NC}$ to $[0, 1]$, called possibility and necessity functions and denoted by Π and N, respectively, which allow us to rank formulas. Π is defined by Dubois et al. (1994) [29] as:

$$\Pi(\varphi) = \max\{\pi(\omega) \mid \omega \in \Omega, \omega \models \varphi\},$$

and evaluates the extent to which φ is consistent with the beliefs expressed by π. The dual *necessity measure* N is defined by:

$$N(\varphi) = 1 - \Pi(\neg\varphi) = \inf\{1 - \pi(\omega) \mid \omega \in \Omega, \omega \not\models \varphi\},$$

and evaluates the extent to which φ is entailed by the available beliefs [29]. So the lower the possibility of an interpretation that makes φ False, the higher the necessity degree of φ. $N(\varphi) = 1$ means φ is totally certain, whereas $N(\varphi) = 0$ expresses the complete lack of knowledge of priority about φ. Note that $N(\top) = 1$ for any possibility distribution, while $\Pi(\top) = 1$

(and, related, $N(\bot)$=0) only holds when the possibility distribution is normalized, i.e. only normalized distributions can express consistent beliefs [29].

A major property of N is Min-Decomposability: $\forall \varphi, \psi, N(\varphi \wedge \psi) = \min(N(\varphi), N(\psi))$. However, for disjunctions only $N(\varphi \vee \psi) \geq \max(N(\varphi), N(\psi))$ holds. Further, one has $N(\varphi) \leq N(\psi)$ if $\varphi \models \psi$, and hence, $N(\varphi) = N(\psi)$ if $\varphi \equiv \psi$.

3.2. Syntactics

A possibilistic formula is a pair $\langle \varphi : \alpha \rangle \in \mathcal{NC} \times (0, 1]$, where $\alpha \in (0, 1]$ expresses the certainty that φ is the case, and it is interpreted as the semantic constraint $N(p) \geq \alpha$. So formulas $\langle \varphi : 0 \rangle$ are excluded. A *possibilistic base* Σ is a collection of possibilistic formulas $\Sigma = \{\langle \varphi_i : \alpha_i \rangle \mid i = 1, \ldots, k\}$ and corresponds to a set of constraints on possibility distributions. The propositional knowledge base of Σ is denoted as Σ^*, namely $\Sigma^* = \{\varphi | \langle \varphi : \alpha \rangle \in \Sigma\}$. Σ is consistent if and only if Σ^* is consistent. It is noticeable that, due to Min-Decomposability, *a possibilistic logic base can be easily put in clausal form.*[2]

Typically, there can be many possibility distributions that satisfy the set of constraints $N(\varphi) \geq \alpha$ but we are usually only interested in the *least specific possibility distribution*, i.e. the possibility distribution that makes minimal commitments, namely, the *greatest possibility distribution* w.r.t. the following ordering: π is a least specific possibility distribution compatible with Σ if for any π', $\pi' \neq \pi$, compatible with Σ, one has $\forall \omega \in \Omega, \pi(\omega) \geq \pi'(\omega)$. Such a least specific possibility distribution always exists and is unique [29].

Thus, for a given $\langle \varphi : \alpha \rangle$, possibilistic distributions should consider that an ω that makes φ True is possible at the maximal level, say 1, while an ω that makes φ False is possible at most at level $1 - \alpha$. Thus the semantic counterpart of a base Σ, or the least specific distribution π_Σ is defined by, $\forall \omega, \omega \in \Omega$:

$$\pi_\Sigma(\omega) = \begin{cases} 1 & \text{if } \forall \langle \varphi_i, \alpha_i \rangle \in \Sigma, \omega \models \varphi_i \\ \min\{1 - \alpha_i \mid \omega \not\models \varphi_i,\ \langle \varphi_i, \alpha_i \rangle \in \Sigma\} & \text{otherwise} \end{cases}$$

Proposition 3.1. *Let Σ be a possibilistic base. For any possibility distribution π on Ω, π satisfies Σ if and only if $\pi \leq \pi_\Sigma$.*

Proposition 3.1 says that π_Σ is the least specific possibility distribution satisfying Σ and it has been shown in reference [29].

3.3. Syntactic deduction

This subsection introduces some few notions about deduction in possibilistic logic and starts by the well-known possibilistic inference rules to be handled in this article:

Definition 3.2. We define below three rules, where $\ell \in \mathcal{L}; \varphi, \psi \in \mathcal{NC}$ and $\alpha, \beta \in (0, 1]$. The first is possibilistic resolution [30,31]; the second rule is Min-Decomposability; and the third rule, Max-Necessity, follows from the semantic constraint meaning of $\langle \varphi : \alpha \rangle$.

- Resol : $\langle (\vee\ \ell\ \varphi) : \alpha \rangle, \langle (\vee\ \neg\ell\ \psi) : \beta \rangle \vdash \langle\ (\vee\ \varphi\ \psi) : \mathbf{min}\{\alpha, \beta\}\ \rangle$.
- MinD : $\langle \varphi : \alpha \rangle,\ \langle \psi : \beta \rangle \vdash \langle\ [\wedge \varphi\ \psi] : \mathbf{min}\{\alpha, \beta\}\ \rangle$.
- MaxN : $\langle \varphi : \alpha \rangle,\ \langle \varphi : \beta \rangle \vdash \langle\ \varphi : \mathbf{max}\{\alpha, \beta\}\ \rangle$.

Before formulating the completeness theorem in possibilistic logic, we need the next concept of α-cut; the α-cut (resp. strict α-cut) of Σ, denoted $\Sigma_{\geq\alpha}$ (resp. $\Sigma_{>\alpha}$), is the set of classical formulas in Σ having a necessity degree at least equal to α (resp. strictly greater than α), namely $\Sigma_{\geq\alpha} = \{\varphi \mid \langle \varphi : \beta \rangle \in \Sigma, \beta \geq \alpha\}$ (resp. $\Sigma_{>\alpha} = \{\varphi \mid \langle \varphi : \beta \rangle \in \Sigma, \beta > \alpha\}$).

Theorem 3.3. *The following soundness and completeness theorem holds:*

$$\Sigma \models_\pi \langle \varphi : \alpha \rangle \Leftrightarrow \Sigma \vdash_{\text{Res}} \langle \varphi : \alpha \rangle \quad \Longleftrightarrow \quad \Sigma^*_{\geq\alpha} \models \varphi \Leftrightarrow \Sigma^*_{\geq\alpha} \vdash \varphi$$

where $\models_\pi$ means any ω compatible with Σ is also compatible with $\langle \varphi : \alpha \rangle$, or formally, $\forall \omega, \pi_\Sigma(\omega) \leq \pi_{\{\langle p:\alpha \rangle\}}(\omega)$. $\vdash_{\text{Res}}$ relies on the repeated use of possibilistic resolution.

The last half of the above expression reduces to the soundness and completeness of propositional logic applied to each cut level of Σ, which is an ordinary propositional base.

[2] Nevertheless, as said previously, this translation can blow up exponentially the size of formulas and so can dramatically reduce the overall efficiency of the clausal reasoner.

3.4. Partial inconsistency

The inconsistency degree of a base Σ in terms of its α-cut can be equivalently defined as the largest weight α such that the α-cut of Σ is inconsistent:

$$\text{Inc}(\Sigma) = \max\{\alpha \mid \Sigma_{\geq\alpha} \text{ is inconsistent}\}.$$

$\text{Inc}(\Sigma) = 0$ entails Σ^* is consistent. In [29], the inconsistency degree of Σ is defined by the least possibility distribution π_Σ, concretely $\text{Inc}(\Sigma) = 1 - \sup_{\omega\in\Omega}\pi_\Sigma(\omega)$.

To check whether φ follows from Σ, one should add $\langle\neg\varphi : 1\rangle$ to Σ and then check whether $\Sigma \cup \{\langle\neg\varphi : 1\rangle\} \vdash \langle\bot : \alpha\rangle$. Equivalently the maximum α s.t. $\Sigma \models \langle\varphi : \alpha\rangle$ is given by the inconsistency degree of $\Sigma \cup \{\langle\neg\varphi : 1\rangle\}$, i.e. $\Sigma \models \langle\varphi : \alpha\rangle$ iff $\alpha = \text{Inc}(\Sigma \cup \{\langle\neg\varphi, 1\rangle\})$.

Proposition 3.4. *The next statements are proven in [29]:*

$$\Sigma \models \langle\varphi : \alpha\rangle \text{ iff } \Sigma \cup \{\langle\neg\varphi : 1\rangle\} \vdash \langle\bot : \alpha\rangle \text{ iff } \alpha = \text{Inc}(\Sigma \cup \{\langle\neg\varphi : 1\rangle\}) \text{ iff } \Sigma^*_{\geq\alpha} \vdash \varphi.$$

This result shows that any deduction problem in possibilistic logic can be viewed as computing an inconsistency degree.

A base $\Sigma = \{\langle\varphi_i : \alpha_i\rangle \mid i = 1, \dots, k\}$ is called Horn, clausal or NC if $\varphi_i, 1 \leq i \leq 1$, are Horn, clausal or NC, respectively. The novel class to be defined will be called Horn-NC. Computing the inconsistency degree of such classes has the following complexities:

- Co-NP-complete [57] for clausal bases and polynomial [57] for Horn bases.
- Co-NP-complete for NC bases. This claim stems from: (i) Theorem 3.3 applies to both clausal and NC bases; and (ii) checking whether an interpretation is a model of an NC propositional formula is polynomial as for clausal formulas.
- Polynomial for Horn-NC bases as proven in Section 7.2.

4. The possibilistic horn-NC class: $\overline{\mathcal{H}}_\Sigma$

Subsection 4.1 defines informally the Horn-NC formulas and Subsection 4.2 formally.

The formal proofs have been relegated to Subsection 7.1.

We will denote the class of possibilistic Horn clausal formulas by $\mathcal{H}_\Sigma$.

4.1. Informal definition of $\overline{\mathcal{H}}_\Sigma$

We start with the negative formulas that extend the negative literals in the clausal setting.

Definition 4.1. A possibilistic formula is negative if its propositional formula has uniquely negative literals. We will denote the set of negative possibilistic formulas by $\mathcal{N}^-_\Sigma$.

Example 4.2. $\langle(\vee_2\ [\wedge\ \neg P\ \neg R]\ [\wedge_2\ \neg S\ (\vee\ \neg P\ \neg Q)]) : \mathbf{0.5}\rangle \in \mathcal{N}^-_\Sigma$. □

Next we upgrade the Horn pattern "a Horn clause has (any number of negative literals and) at most one positive literal" to the NC context in the next way:

Definition 4.3. A propositional formula is Horn-NC if all the disjunctions have any number of negative disjuncts and at most one non-negative disjunct. A possibilistic formula is Horn-NC if its propositional formula is Horn-NC. We denote the class of possibilistic Horn-NC formulas by $\overline{\mathcal{H}}_\Sigma$. A Horn-NC possibilistic base is a subset of $\overline{\mathcal{H}}_\Sigma$.

Clearly $\mathcal{H}_\Sigma \subset \overline{\mathcal{H}}_\Sigma$.

By Definition 4.3, all sub-formulas of any Horn-NC are Horn-NC too. Yet, the converse does not hold: there are non-Horn-NC formulas whose all sub-formulas are Horn-NC.

Example 4.4. The propositional part of φ_1 below has only one non-negative disjunct and so φ_1 is Horn-NC, while φ_2 is not Horn-NC as its formula has two non-negative disjuncts.

$$\varphi_1 = \langle(\vee\ [\wedge\ \neg Q\ \neg S]\ [\wedge\ R\ P]) : \mathbf{0.7}\rangle \quad \varphi_2 = \langle(\vee\ [\wedge\ \neg Q\ S]\ [\wedge\ R\ \neg P]) : \mathbf{0.6}\rangle$$

Example 4.5. We now consider φ and φ' below. φ' results from φ by switching the leftmost $\neg P$ for P. All disjunctions of φ, i.e. $(\vee\ \neg P\ \ R)$, $(\vee\ P\ \neg S)$ and the whole formula, have one non-negative disjunct; so φ is Horn-NC. Yet, φ' is of the kind $(\vee\ P\ \ \phi)$, ϕ being non-negative, and so it has two non-negative disjuncts; thus φ' is not Horn-NC.

- $\varphi = \langle\, (\vee\ \ \neg P\ \ [\wedge\ \ (\vee\ \ \neg P\ \ R)\ \ [\wedge\ \ Q\ \ (\vee\ \ P\ \ \neg S)\,]\,])\, : \mathbf{0.6}\,\rangle$
- $\varphi' = \langle\, (\vee\ \ \ P\ \ [\wedge\ \ (\vee\ \ \neg P\ \ R)\ \ [\wedge\ \ Q\ \ (\vee\ \ P\ \ \neg S)\,]\,])\, : \mathbf{0.3}\,\rangle$ □

4.2. Formal definition of $\overline{\mathcal{H}}_\Sigma$

We first individually specify Horn-NC conjunctions and Horn-NC disjunctions, and subsequently, by merging both specifications, we *compactly* and formally specify $\overline{\mathcal{H}}_\Sigma$.

Just as conjunctions of propositional Horn clausal formulas are Horn, likewise conjunctions of propositional Horn-NC formulas are also Horn-NC.

Proposition 4.6. *Let $\alpha, \alpha_i \in (0, 1]$. Possibilistic Horn-NC formulas verify:*

$$\langle\, [\wedge\, \varphi_1\ \ldots\, \varphi_i \ldots\, \varphi_k] : \alpha\,\rangle \in \overline{\mathcal{H}}_\Sigma \ \text{ iff for } 1 \le i \le k,\ \langle \varphi_i : \alpha_i\,\rangle \in \overline{\mathcal{H}}_\Sigma.$$

The proof is straightforward and so is omitted.

One can verify that the propositional condition in Definition 4.3 can be equivalently reformulated inductively as: *"an NC is Horn-NC if all its disjunctions have any number of negative disjuncts and one disjunct is Horn-NC".* This leads to the next formalization:

Lemma 4.7. *Let $\alpha, \alpha_i, \alpha_j \in (0, 1]$. Possibilistic Horn-NC formulas hold the next property:*

$$\langle\, (\vee\, \varphi_1 \ldots \varphi_i \ldots \varphi_k) : \alpha\,\rangle \in \overline{\mathcal{H}}_\Sigma \ \text{ iff } \ \exists i \text{ s.t. } \langle \varphi_i : \alpha_i\,\rangle \in \overline{\mathcal{H}}_\Sigma \ \text{ and } \ \forall j \neq i,\ \langle \varphi_j : \alpha_j\,\rangle \in \mathcal{N}^-_\Sigma.$$

$\widehat{\mathcal{H}}_\Sigma$ is defined below using Lemmas 4.6 and 4.7.

We recall that the connective $\neg$ applies only at the literal level.

Definition 4.8. We define the set $\widehat{\mathcal{H}}_\Sigma$ as the smallest set such that the conditions below hold, where $k \ge 1$, $\alpha \in (0, 1]$ and $\mathcal{L}$ is the set of literals:

(1) $\langle \ell : \alpha\,\rangle \in \widehat{\mathcal{H}}_\Sigma$, where $\ell \in \mathcal{L}$.

(2) If $\forall i$, $\langle \varphi_i : \alpha_i \rangle \in \widehat{\mathcal{H}}_\Sigma$ then $\langle\, [\wedge\, \varphi_1\ \ldots\, \varphi_i \ldots \varphi_k] : \alpha\,\rangle \in \widehat{\mathcal{H}}_\Sigma$.

(3) If $\langle \varphi_i : \alpha_i \rangle \in \widehat{\mathcal{H}}_\Sigma$ and $\forall j \neq i$, $\langle \varphi_j : \alpha_j\,\rangle \in \mathcal{N}^-_\Sigma$ then $\langle\, (\vee\, \varphi_1 \ldots \varphi_i \ldots \varphi_k) : \alpha\,\rangle \in \widehat{\mathcal{H}}_\Sigma$.

Theorem 4.9. *We have that $\widehat{\mathcal{H}}_\Sigma = \overline{\mathcal{H}}_\Sigma$.*

Example 4.10. Viewed from Definition 4.8, below we analyze φ and φ' from Example 4.5, where $\alpha \in (0, 1]$:

- By (3), $\langle\, (\vee\ \ \neg P\ \ R) : \alpha\,\rangle \in \overline{\mathcal{H}}_\Sigma$.
- By (3), $\langle\, (\vee\ \ P\ \ \neg S) : \alpha\,\rangle \in \overline{\mathcal{H}}_\Sigma$.
- By (2), $\langle\, [\wedge\ \ Q\ \ (\vee\ P\ \ \neg S)\,] : \alpha\,\rangle \in \overline{\mathcal{H}}_\Sigma$.
- By (2), $\langle\, \phi = [\wedge\ \ (\vee\ \ \neg P\ \ R)\ \ [\wedge\ \ Q\ \ (\vee\ \ P\ \ \neg S)\,]\,] : \alpha\,\rangle \in \overline{\mathcal{H}}_\Sigma$.
- By (3), $\varphi = \langle\, (\vee\ \ \neg P\ \ \ \phi\,) : \mathbf{0.6}\rangle \in \overline{\mathcal{H}}_\Sigma$
- By (3), $\varphi' = \langle\, (\vee\ \ P\ \ \ \phi\,) : \mathbf{0.3}\rangle \notin \overline{\mathcal{H}}_\Sigma$. □

Example 4.11. We analyze $\langle \varphi : \mathbf{0.8} \rangle$ from Example 1.1. We can rewrite φ as follows:

- $\psi_1 = (\vee\ \ \neg P\ \ \neg Q\ \ \ R\,)$. $\qquad \psi_2 = (\vee\ \ \phi_1\ \ [\wedge\ \ \phi_2\ \ \neg P])$.
- $\psi_3 = (\vee\ \ \neg Q\ \ [\wedge\ \ \psi_1\ \ \psi_2\ \ Q\,])$. $\qquad \varphi = [\wedge\ \ P\ \ \ \psi_3\ \ \ \phi_3\,]$.

Below we analyze one-by-one such disjunctions and finally the proper φ, with $\alpha \in (0, 1]$:

- $\langle \psi_1 : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ because ψ_1 is Horn.
- $\langle \psi_2 : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ if at least one $\langle \phi_1 : \alpha \rangle$ or $\langle \phi_2 : \alpha \rangle$ belongs to $\mathcal{N}_\Sigma$.
- $\langle \psi_3 : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ if $\langle \psi_2 : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ (as $\langle \psi_1 : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$).
- $\langle \varphi : \mathbf{0.8} \rangle \in \overline{\mathcal{H}}_\Sigma$ if $\langle \psi_2 : \alpha \rangle, \langle \phi_3 : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ (as $\langle \psi_3 : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ if $\langle \psi_2 : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$).

Summarizing, we have that:

$\langle \varphi : \mathbf{0.8} \rangle \in \overline{\mathcal{H}}_\Sigma$ if $\langle \phi_3 : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ and if at least one of $\langle \phi_1 : \alpha \rangle$ or $\langle \phi_2 : \alpha \rangle$ is in $\mathcal{N}_\Sigma$. □

Example 4.12. One can check that the nested head in Example 1.2 is a propositional Horn-NC formula. Also, the formula of the whole nested rule, which is the disjunction of its head and the negation of its body, is a propositional Horn-NC formula.

The next three theorems state the relationships between $\overline{\mathcal{H}}_\Sigma$ and the standard Horn and NC possibilistic classes. The first one states that applying distributivity to a Horn-NC formula $\langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ leads to a Horn formula.

Theorem 4.13. *Denoting by* $\mathrm{cl}(\varphi)$ *the clausal form of* φ*, we have:*

$$\langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma \quad \textit{entails} \quad \langle \mathrm{cl}(\varphi) : \alpha \rangle \in \mathcal{H}_\Sigma .$$

Clearly, $\mathcal{H}_\Sigma \subset \overline{\mathcal{H}}_\Sigma$.

We now prove that both classes are semantically equivalent, namely they both have the same expressiveness and their syntactical difference is that possibilistic Horn-NC bases may need exponentially less symbols than their equivalent Horn clausal bases.

Theorem 4.14. *Let us consider the classes* $\overline{\mathcal{H}}_\Sigma$ *and* $\mathcal{H}_\Sigma$*. Each formula in a class is logically equivalent to some formula in the other class and formally:*

$$\langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma \quad \textit{iff} \quad \langle H : \alpha \rangle \in \mathcal{H}_\Sigma \quad \textit{where } \varphi \equiv H .$$

Proof. By Theorem 4.13, for every $\langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ there exists $\langle \mathrm{cl}(\varphi) : \alpha \rangle \in \mathcal{H}_\Sigma$ and obviously $\varphi \equiv \mathrm{cl}(\varphi)$. The converse follows from the fact that $\mathcal{H}_\Sigma \subset \overline{\mathcal{H}}_\Sigma$. ■

Corollary 4.15. *Every consistent* $\langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ *have only one minimal model.*

Proof. It follows from Theorem 4.14 and the well-known fact that consistent Horn clausal formulas have only one minimal model. ■

The next theorem makes it explicit how the classes Horn-NC and NC are related.

Theorem 4.16. $\overline{\mathcal{H}}_\Sigma$ *contains the next NC fragment: if applying distributivity to an NC formula* $\langle \varphi : \alpha \rangle$ *results in a Horn formula, then* $\langle \varphi : \alpha \rangle$ *is Horn-NC; formally:*

$$\langle \varphi : \alpha \rangle \in \mathcal{NC}_\Sigma \quad \textit{and} \quad \langle \mathrm{cl}(\varphi) : \alpha \rangle \in \mathcal{H}_\Sigma \quad \textit{implies} \quad \langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma .$$

where $\mathcal{NC}_\Sigma$ *is the set of possibilistic NC formulas and* $\mathrm{cl}(\varphi)$ *is the clausal form of* φ.

In view of the last three theorems, we can conclude that $\overline{\mathcal{H}}_\Sigma$ is non-clausal, analogous to the standard possibilistic Horn class $\mathcal{H}_\Sigma$.

5. Possibilistic NC unit-resolution $\mathcal{UR}_\Sigma$

Possibilistic clausal resolution was defined in the 1980s [30,31] but its non-clausal generalization has not been proposed yet. This section is a step forward towards its definition as we define possibilistic NC unit-resolution, denoted $\mathcal{UR}_\Sigma$. The main rule of $\mathcal{UR}_\Sigma$ is called UR_Σ, and while the other rules in $\mathcal{UR}_\Sigma$ are simple, UR_Σ is somewhat involved and so is presented in two steps:

– for quasi-clausal Horn-NC bases in Subsection 5.1; and
– for general Horn-NC bases in Subsection 5.2.

Afterwards, Subsection 5.3 describes $\mathcal{UR}_\Sigma$, which besides UR_Σ, comprises some propositional rules. Subsection 5.4 gives two further inferences rules, not needed for warranting completeness. Subsection 5.5 determines the inconsistency degree of a Horn-NC base. Finally, Subsection 5.6 explains the reasoning by contradiction in Horn-NC bases. The formal proofs are included in Subsection 7.2.

5.1. Quasi-clausal unit-resolution

We start with propositional and then switch to possibilistic logic. Assume the quasi-clausal formulas below; ℓ and $\neg\ell$[3] are complementary literals and the φ's and ϕ's are formulas:

$$[\wedge\ \varphi_1 \ldots \varphi_{l-1}\ \ell\ \varphi_{l+1} \ldots \varphi_{i-1}\ (\vee\ \phi_1 \ldots \phi_{j-1}\ \neg\ell\ \phi_{j+1} \ldots \phi_k)\ \varphi_{i+1} \ldots \varphi_n]$$

These formulas are quasi-clausal as if the φ's and ϕ's were clauses and literals, respectively, then they would be clausal. Obviously a quasi-clausal formula is equivalent to:

$$[\wedge\ \varphi_1 \ldots \varphi_{l-1}\ \ell\ \varphi_{l+1} \ldots \varphi_{i-1}\ (\vee\ \phi_1 \ldots \phi_j\ \phi_{j+1} \ldots \phi_k)\ \varphi_{i+1} \ldots \varphi_n]$$

Thus, for propositional formulas, one can derive the next simple inference rule:

$$\frac{\ell\ ,\ (\vee\ \phi_1 \ldots \phi_j\ \neg\ell\ \phi_{j+1} \ldots \phi_k)}{(\vee\ \phi_1 \ldots \phi_j\ \phi_{j+1} \ldots \phi_k)} \tag{1}$$

Switching to possibilistic logic, NC unit-resolution is applicable when Σ has a unit clause $\langle \ell : \alpha\rangle$ and a formula $\langle [\wedge\ \varphi_1 \ldots \varphi_i\ (\vee\ \phi_1 \ldots \phi_{j-1}\ \neg\ell\ \phi_{j+1} \ldots \phi_k)\ \varphi_{i+1} \ldots \varphi_n] : \beta\rangle$. Then, by using Min-Decomposability (Definition 3.2), one can easily derive:

$$\frac{\langle \ell\ :\ \alpha\rangle\ ,\ \langle (\vee\ \phi_1 \ldots \phi_j\ \neg\ell\ \phi_{j+1} \ldots \phi_k)\ :\ \beta\rangle}{\langle (\vee\ \phi_1 \ldots \phi_j\ \phi_{j+1} \ldots \phi_k)\ :\ \mathbf{min}\{\alpha, \beta\}\rangle} \tag{2}$$

Notice that for clausal formulas, *Rule (2) coincides with possibilistic clausal unit-resolution.* The soundness of (2) follows from the property Min-Decomposability. If $\mathcal{D}$ stands for $(\vee\ \phi_1 \ldots \phi_j\ \phi_{j+1} \ldots \phi_n)$, then (2) can be concisely rewritten as:

$$\frac{\langle \ell : \alpha\rangle\ ,\ \langle (\vee\ \neg\ell\ \mathcal{D} : \beta\rangle}{\langle \mathcal{D}\ :\ \mathbf{min}\{\alpha, \beta\}\rangle} \tag{3}$$

Notice that the previous rule amounts to substituting the formula referred to by the right conjunct in the numerator with the formula in the denominator, and in practice, to just eliminate $\neg\ell$ and update the necessity weight. Let us illustrate these notions.

Example 5.1. Let Σ be a base including $\langle P : \mathbf{0.8}\rangle$ and φ below, where ϕ is a formula:

$$\varphi = \langle [\wedge_4\ \phi\ (\vee\ \neg R\ \ \neg P\ \ S)\ (\vee_2\ \ S\ [\wedge\ \neg Q\ \ \neg P\,])\ \ R\,] : \mathbf{0.6}\rangle.$$

Taking $\langle P : \mathbf{0.8}\rangle$ and the left-most $\neg P$ in φ, we have $\mathcal{D} = (\vee\ \neg R\ \ S)$, and by applying Rule (3) to φ, the formula below is deduced and added to the base Σ.

$$\langle [\wedge_4\ \phi\ (\vee\ \neg R\ \ S)\ (\vee_2\ \ S\ [\wedge\ \neg Q\ \ \neg P\,])\ \ R\,] : \mathbf{0.6}\rangle. \quad \square$$

We now extend our analysis from formulas with pattern $\langle (\vee\ \neg\ell\ \ \mathcal{D}) : \beta\rangle$ to those with pattern $\langle (\vee\ \ \mathcal{C}(\neg\ell)\ \ \ \mathcal{D}) : \beta\rangle$, where $\mathcal{C}(\neg\ell)$ is the maximal sub-formula in Σ that becomes false when $\neg\ell$ is false, namely $\mathcal{C}(\neg\ell)$ is the maximal sub-formula in Σ equivalent to a conjunction of the kind $\neg\ell \wedge \psi$, i.e. $\mathcal{C}(\neg\ell) \equiv \neg\ell \wedge \psi$.

Example 5.2. If Σ contains $\langle \ell : \alpha\rangle$ and another formula of the kind:

$$\varphi = \langle (\vee_3\ \ \varphi_1\ \ \ [\wedge_3\ \ \phi_1\ [\wedge_2\ \ \neg\ell\ \ (\vee\ \ \phi_2\ \neg P\,)\,]\ \phi_3\,]\ \ \ \varphi_2\,) : \mathbf{0.7}\,\rangle$$

then $\mathcal{C}(\neg\ell) = [\wedge_3\ \ \phi_1\ \ [\wedge_2\ \neg\ell\ \ (\vee\ \ \phi_2\ \ \neg P\,)\,]\ \phi_3\,]$ because $\mathcal{C}(\neg\ell)$ verifies:

(*i*) $\mathcal{C}(\neg\ell) \equiv \neg\ell \wedge \psi = \neg\ell \wedge [\wedge_3\ \phi_1\ (\vee\ \phi_2\ \neg P\,)\ \phi_3]$ (namely, if $\neg\ell$ is false so is $\mathcal{C}(\neg\ell)$);

(*ii*) no sub-formula of φ including $\mathcal{C}(\neg\ell)$ verifies (*i*). $\square$

---

[3] For colors see the web version of the article.

Example 5.3. Let us consider φ below. If we take $\neg P$, then φ has a sub-formula with pattern $(\vee\ \mathcal{C}(\neg P)\ \mathcal{D})$, in which $\mathcal{C}(\neg P) = [\wedge\ \neg P\ \ (\vee\ S\ \neg R)\,]$ and $\mathcal{D} = (\vee\ \neg R\ S)$.

$$\varphi = \langle\, [\wedge_4\ \phi\ \ (\vee_3\ \neg R\ \ [\wedge_2\ \neg P\ \ (\vee\ S\ \neg R)\,]\ \ S\,)\ \ \phi_1\ \ R\,] : \mathbf{0.6} \,\rangle \quad \square$$

Regarding the inference rule, we have that when Σ has both a unitary clause $\langle \ell : \alpha \rangle$ and another formula $\langle \varphi : \beta \rangle$ such that φ has the pattern $(\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D})$, then the possibilistic NC unit-resolution rule is easily obtained by extending Rule (3) as follows:

$$\frac{\langle \ell\ :\ \alpha \rangle\ ,\ \langle\, (\vee\ \mathcal{C}(\neg\ell)\ \ \mathcal{D}\,) : \beta \,\rangle}{\langle\, \mathcal{D}\ :\ \mathbf{min}(\alpha, \beta)\,\rangle} \tag{4}$$

Proposition 5.4. *Rule (4) is sound:*

$$\langle \ell\ :\ \alpha \rangle\ ,\ \langle\, (\vee\ \mathcal{C}(\neg\ell)\ \ \mathcal{D}) : \beta \rangle\ \models\ \langle \mathcal{D} : \mathbf{min}\{\alpha, \beta\} \rangle.$$

Example 5.5. Rule (4) with $\varphi_1 = \langle P : \mathbf{0.3} \rangle$ and with φ from Example 5.3 derives:

$$\langle\, [\wedge_4\ \phi\ \ (\vee\ \neg R\ \ S\,)\ \ \phi_1\ \ R\,] : \mathbf{0.3} \,\rangle.$$

5.2. NC unit-resolution rule

Coming back to the almost-clausal pattern expressed previously and extending its literal $\neg\ell$ to $\mathcal{C}(\neg\ell)$, we now rewrite it compactly as indicated below, where Π and Π' denote a concatenation of formulas, namely $\Pi = \varphi_1 \ldots \varphi_{i-1}$ and $\Pi' = \varphi_{i+1} \ldots \varphi_n$:

$$\langle\, [\wedge\ \Pi\ \ (\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D}\,)\ \ \Pi'\,] : \beta \,\rangle$$

We now analyze when NC unit-resolution can be indeed applied to Horn-NC bases Σ. That is, Σ must have a unit-clause $\langle \ell\ :\ \alpha \rangle$ and a possibilistic Horn-NC formula, denoted $\langle \Pi : \beta \rangle$, with a syntactical pattern of the kind below,

$$\langle\, [\wedge\ \Pi_1\ (\vee_2 \ldots\ \ [\wedge_k\ \ \Pi_k\ \ (\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D}\,)\ \ \Pi'_k\,]\ \ \ldots\,)\ \ \Pi'_1\,] : \beta \,\rangle$$

where all the Π_j's and Π'_j's are concatenations of formulas, e.g. for the nesting level $j, 1 \le j \le k$, we have $\Pi_j = \varphi_{j_1} \ldots \varphi_{j_{i-1}}$ and $\Pi'_j = \varphi_{j_{i+1}} \ldots \varphi_{j_{n_j}}$. By following the same principle that led us to Rule (4) and taking into account that $N(\varphi_1 \wedge \varphi_2) = \min\{N(\varphi_1), N(\varphi_1)\}$, one obtains the possibilistic NC unit-resolution rule UR_Σ:

$$\frac{\langle \ell\ :\ \alpha \rangle\ ,\ \langle\, [\wedge_1\ \Pi_1 \ldots [\wedge_k\ \ \Pi_k\ \ (\vee\ \mathcal{C}(\neg\ell)\ \ \mathcal{D}\,)\ \ \Pi'_k\,] \ldots\ \Pi'_1\,] : \beta \,\rangle}{\langle\, [\wedge_1\ \Pi_1 \ldots [\wedge_k\ \ \Pi_k\ \ \mathcal{D}\ \ \Pi'_k\,] \ldots\ \Pi'_1\,] : \mathbf{min}\{\alpha, \beta\} \,\rangle}\ UR_\Sigma \tag{5}$$

Recapitulating, Rule (5) indicates that if the Horn-NC base Σ has two formulas such that one is a unit clause $\langle \ell\ :\ \alpha \rangle$ and the other $\langle \Pi : \beta \rangle$ has the pattern of the right conjunct in the numerator, then Π can be replaced with the formula in the denominator. In practice, applying (5) amounts to just removing $\mathcal{C}(\neg\ell)$ from Π and updating the necessity weight.

We now denote the right conjunct in the numerator of (5) by Π and denote that $(\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D}\,)$ is a sub-formula of Π by $\Pi \succ (\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D})$. Rule (5) above can be compacted, giving rise to a more concise formulation of UR_Σ:

$$\frac{\langle \ell : \alpha \rangle\ ,\ \langle\, \Pi \succ (\vee\ \ \mathcal{C}(\neg\ell)\ \ \mathcal{D}\,) : \beta \,\rangle}{\langle\, \Pi \succ \mathcal{D} : \mathbf{min}\{\alpha, \beta\} \,\rangle}\ UR_\Sigma$$

Proposition 5.6. *Te rule UR_Σ is sound:*

$$\langle \ell\ :\ \alpha \rangle\ ,\ \langle\, \Pi \succ (\vee\ \mathcal{C}(\neg\ell)\ \ \mathcal{D}) : \beta \rangle\ \models\ \langle \mathcal{D} : \mathbf{min}\{\alpha, \beta\} \rangle.$$

Corollary 5.7. UR_Σ coincides with clausal unit-resolution [29,33] *for clausal formulas.*

The corollary is easy to verify.

Examples 6.1 and 6.2, using two simple formulas, illustrate how UR_Σ works. Two more complete formulas are given in Examples 6.5 and 6.8 but they make use of other inferential mechanisms that are described in the remainder of this section.

Note. The clausal-like formulation of NC unit-resolution contrasts with the functional-like one of NC (full) resolution [64] handled until now in the literature (see also [5]). We believe that our version, as previously said, is more suitable to understand, implement and formally analyze.

5.3. NC unit-resolution calculus

$\mathcal{UR}_\Sigma$, besides UR_Σ, also includes: (a) propositional NC unit-resolution, or UR_P, which is UR_Σ adapted to propositional logic, and (b) rules to simplify propositional formulas.

Propositional NC Unit-Resolution. A major difference between computing the inconsistency degree of clausal and non-clausal bases is that the unity members of the former, i.e. clauses, are always consistent, while a non-clausal formula can itself be inconsistent. That is, Σ can contain a formula $\langle \Pi : \alpha \rangle$ where Π is inconsistent, and if so, $\langle \Pi : \alpha \rangle$ is equivalent to $\langle \bot : \alpha \rangle$, which brings to:

Proposition 5.8. *If* $\langle \Pi : \alpha \rangle \in \Sigma$ *and* Π *is inconsistent then* $\mathrm{Inc}(\Sigma) \geq \alpha$.

Proof. By definition $\mathrm{Inc}(\Sigma) = \max\{\beta \mid \Sigma_{\geq\beta}$ is inconsistent$\}$. If Π is inconsistent, then trivially $\langle \bot : \alpha \rangle \in \Sigma_{\geq\alpha}$, and thus, $\Sigma^*_{\geq\alpha}$ is inconsistent. So $\mathrm{Inc}(\Sigma) \geq \alpha$. ■

Hence, first of all, the propositional formula Π of each $\langle \Pi : \alpha \rangle \in \Sigma$ must be checked for consistency. If Π is inconsistent, then, by definition, $\mathrm{Inc}(\Sigma)$ is the maximum of α and the inconsistency degree of the strict α-cut of Σ. Thus, one can remove from Σ all formulas $\langle \Pi : \beta \rangle$ such that $\beta \leq \alpha$ and search whether $\mathrm{Inc}(\Sigma_{>\alpha}) > 0$.

The propositional rule UR_P, which tests the consistency of Π when $\langle \Pi : \alpha \rangle \in \Sigma$, is easily derived from UR_Σ by considering that the conjunction of a unit clause ℓ and of a sub-formula $\mathcal{C}(\neg\ell)$ happens inside Π. Thus UR_P is formalized as follows:

$$\frac{\langle\, \ell \wedge \Pi \succ (\vee\ \mathcal{C}(\neg\ell)\ \ \mathcal{D}) : \alpha \,\rangle}{\langle\, \Pi \succ \mathcal{D} : \alpha \rangle} UR_P \tag{6}$$

Lemma 5.9. *A propositional Horn-NC formula* φ *is inconsistent iff* $\mathcal{UR}_P = \{UR_P, \bot\vee, \bot\wedge, \odot\phi, \odot\odot\} \subset \mathcal{UR}_\Sigma$ *applied to* $\langle \varphi : \alpha \rangle$ *derives* $\langle \bot : \alpha \rangle$.

Proposition 5.10. *Testing the propositional consistency of* φ *such that* $\langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ *with inferences* $\{UR_P, \bot\vee, \bot\wedge, \odot\phi, \odot\odot\} \subset \mathcal{UR}_\Sigma$ *is polynomial.*

Examples. A complete example through which we show how UR_P proceeds is Example 6.3 and Example 6.4 illustrates the effects of applying Proposition 5.8.

Simplification Rules. Each application of UR_Σ and UR_P demands the subsequent application of trivial logical simplifications of propositional formulas. For instance, the formulas $(\vee\ \varphi\ (\vee\ P\ (\vee\ \ \neg R\ \phi)))$ and $(\vee\ P\ [\wedge\ (\vee)\ \varphi])$ can be obviously substituted by their equivalent $(\vee\ \varphi\ P\ \neg R\ \phi)$ and P, respectively. Assuming that $\langle \Pi : \alpha \rangle \in \Sigma$, the first two rules below simplify formulas by (upwards) propagating $(\vee)$:

- $\dfrac{\langle\ \Pi \succ (\vee\ \ \phi_1 \dots \phi_{i-1}\ (\vee)\ \phi_{i+1} \dots \phi_k\ ) : \alpha \rangle}{\langle\ \Pi \succ (\vee\ \ \phi_1 \dots \phi_{i-1}\ \phi_{i+1} \dots \phi_k\ ) : \alpha\ \rangle} \bot\vee$

- $\dfrac{\langle\ \Pi \succ [\wedge\ \ \varphi_1 \dots \varphi_{i-1}\ (\vee)\ \varphi_{i+1} \dots \varphi_k\ ] : \alpha \rangle}{\langle\ \Pi \succ\ (\vee) : \alpha\ \rangle} \bot\wedge$

The next two rules remove redundant connectives. The first one removes $\odot \in \{\wedge, \vee\}$ if it is applied to a single formula, e.g. $(\vee\ \phi_1)$. The second one removes $\odot$ if it is inside another equal $\odot$, i.e. applies to sub-formulas $\|\odot_1\ \ \varphi_1 \dots \varphi_{i-1} \|\ \odot_2\ \phi_1 \dots \phi_n \|\ \varphi_{i+1} \dots \varphi_k \|$, where $\odot_1 = \odot_2$ and where $\|\ \odot \dots \|$ stands for both $(\vee \dots)$ and $[\wedge \dots]$. So formally, the rules are:

- $\dfrac{\langle\ \Pi \succ \|\odot_1\ \ \varphi_1 \dots \varphi_{i-1} \|\ \odot_2\ \ \phi_1 \|\ \varphi_{i+1} \dots \varphi_k \| : \alpha \rangle}{\langle\ \Pi \succ \|\odot_1\ \ \varphi_1 \dots \varphi_{i-1}\ \phi_1\ \varphi_{i+1} \dots \varphi_k \| : \alpha \rangle} \odot\phi$

- $\dfrac{\langle\ \Pi \succ \|\odot_1\ \ \varphi_1 \dots \varphi_{i-1} \|\ \odot_2\ \ \phi_1 \dots \phi_n \|\ \varphi_{i+1} \dots \varphi_k \| : \alpha\ \rangle, \odot_1 = \odot_2}{\langle\ \Pi \succ \|\odot_1\ \ \varphi_1 \dots \varphi_{i-1}\ \phi_1 \dots \phi_n\ \varphi_{i+1} \dots \varphi_k \|\ \ : \alpha\ \rangle} \odot\odot$

The Calculus $\mathcal{UR}_\Sigma$. The calculus $\mathcal{UR}_\Sigma$ is composed of all the above inference rules, which are recalled in the next definition:

Definition 5.11. We define $\mathcal{UR}_\Sigma$ as the calculus formed by UR_Σ, UR_P and the simplification rules, namely $\mathcal{UR}_\Sigma = \{UR_\Sigma, UR_P, \bot\vee, \bot\wedge, \odot\phi, \odot\odot\}$ and thus where:

- $\dfrac{\langle \ell : \alpha \rangle\, ,\, \langle\, \Pi \succ (\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D}\,) : \beta\, \rangle}{\langle\, \Pi \succ \mathcal{D} : \mathbf{min}\{\alpha, \beta\}\, \rangle}\, UR_\Sigma$
- $\dfrac{\langle\, \ell \wedge \Pi \succ (\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D}\,) : \alpha\, \rangle}{\langle\, \Pi \succ \mathcal{D} : \alpha \rangle} UR_P$
- $\dfrac{\langle\, \Pi \succ\ (\vee\ \phi_1 \ldots \phi_{i-1}\, (\vee)\, \phi_{i+1} \ldots \phi_k\,) : \alpha \rangle}{\langle\, \Pi \succ (\vee\ \phi_1 \ldots \phi_{i-1}\, \phi_{i+1} \ldots \phi_k\,) : \alpha\, \rangle} \bot\vee$
- $\dfrac{\langle\, \Pi \succ [\wedge\ \varphi_1 \ldots \varphi_{i-1}\, (\vee)\, \varphi_{i+1} \ldots \varphi_k\,] : \alpha \rangle}{\langle\, \Pi \succ\ (\vee) : \alpha\, \rangle} \bot\wedge$
- $\dfrac{\langle\, \Pi \succ \|\odot_1\ \varphi_1 \ldots \varphi_{i-1}\, \|\odot_2\ \phi_1\|\, \varphi_{i+1} \ldots \varphi_k\, \| : \alpha \rangle}{\langle\, \Pi \succ \|\odot_1\ \varphi_1 \ldots \varphi_{i-1}\, \phi_1\, \varphi_{i+1} \ldots \varphi_k\, \| : \alpha \rangle} \odot\phi$
- $\dfrac{\langle\, \Pi \succ \|\odot_1\ \varphi_1 \ldots \varphi_{i-1}\, \|\odot_2\ \phi_1 \ldots \phi_n\, \|\, \varphi_{i+1} \ldots \varphi_k\, \| : \alpha\, \rangle, \odot_1 = \odot_2}{\langle\, \Pi \succ \|\odot_1\ \varphi_1 \ldots \varphi_{i-1}\, \phi_1 \ldots \phi_n\, \varphi_{i+1} \ldots \varphi_k\, \|\ : \alpha\, \rangle} \odot\odot$

Remark. Having established possibilistic NC unit-resolution $\mathcal{UR}_\Sigma$, the procedure NC unit-propagation for possibilistic NC formulas can be designed, and on top of it, the possibilistic NC DPLL scheme can be defined (see related and future work section).

Possibilistic Rules. We recall the possibilistic rules in Definition 3.2. We will pay special attention to the deduction of conjunctions $\langle [\wedge\ \ \varphi_1 \ldots \varphi_i \ldots \varphi_k] : \alpha \rangle$. In this case, we can deduce that the necessity weight of each individual φ_i is α. The last rule is *MaxN*:

$$\frac{\langle [\wedge\ \ \varphi_1 \ldots \varphi_k] : \alpha \rangle}{\{\langle \varphi_1 : \alpha \rangle, \ldots, \langle \varphi_k : \alpha \rangle\}}\ \underline{\text{MinD}} \qquad\qquad \frac{\langle \varphi : \alpha \rangle, \langle \varphi : \beta \rangle}{\langle \varphi : \max\{\alpha, \beta\} \rangle}\ \underline{\text{MaxN}}$$

Example. Example 6.5 illustrates how $\mathcal{UR}_\Sigma$ searches for just one empty clause $\langle \bot : \alpha \rangle$.

Lemma 5.12. *Let $\Sigma \in \overline{\mathcal{H}}_\Sigma$. Σ is inconsistent iff applying $\mathcal{UR}_\Sigma$, MinD and MaxN to Σ the formula $\langle (\vee) : \alpha \rangle$ is derived, and if $\langle (\vee) : \alpha \rangle$ is derived then $Inc(\Sigma) \geq \alpha$.*

Proposition 5.13. *Calculating the inconsistency degree of any $\Sigma \in \overline{\mathcal{H}}_\Sigma$ is polynomial.*

Proof. It follows from Proposition 5.10 and the fact that dichotomic search, which is outlined in Subsection 5.5, requires at most $\log n$ calls to the propositional solver. ■

5.4. Further inferences rules

We now present two further inferences not required to ensure completeness but, since they allow shorter proofs, their appropriate management can yield significant speed-ups.

Propositional NC Local-Unit-Resolution. UR_P could also apply to propositional sub-formulas and can be used in the general framework of non-Horn-NC bases. The UR_P local application means that applying UR_P to sub-formulas φ of any formula Π, where $\langle \Pi : \alpha \rangle \in \Sigma$, such that φ has the UR_P numerator pattern, should be authorized. Namely, applying UR_P to sub-formulas with pattern $\varphi = \ell \wedge \Pi \succ (\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D}\,)$ should be permitted and φ could be substituted with $\ell \wedge \Pi \succ \mathcal{D}$. Hence, the formal specification of the Propositional NC Local-Unit-Resolution rule, *LUR*, for any NC $\langle \varphi : \alpha \rangle$ is:

$$\frac{\langle\, \Pi \succ (\,\ell\ \wedge\ \varphi \succ (\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D}\,)) : \alpha\, \rangle}{\langle\, \Pi \succ (\,\ell \wedge \varphi \succ \mathcal{D}\,) : \alpha\, \rangle}\ \underline{LUR}$$

This inference rule should be read: if $\langle \Pi : \alpha \rangle \in \Sigma$ and Π has a **sub-formula** with a literal ℓ conjunctively linked to a sub-formula φ having pattern $(\vee\ \mathcal{C}(\neg\ell)\ \ \mathcal{D})$, then its component $\mathcal{C}(\neg\ell)$ can be eliminated.

Example. Example 6.6 illustrates the functioning of LUR.

Remark. The introduction of this new rule *LUR* applicable to certain sub-formulas habilitates new sequences of inferences, and so, shorter proofs are now available.

Proposition 5.14. *If applying LUR to Π results in Π', then $\langle \Pi : \alpha \rangle$ and $\langle \Pi' : \alpha \rangle$ are logically equivalent.*

Proof. The soundness of *LUR* follows from that of UR_P which is proven in Lemma 5.9. ∎

Possibilistic NC Hyper-Unit-Resolution. The rule of possibilistic NC unit-resolution UR_Σ can be extended in order to obtain Possibilistic NC Hyper-Unit-Resolution (*HUR*), which is formalized as follows. Assume that the possibilistic base has a unit-clause $\langle \ell : \alpha \rangle$ and two sub-formulas $(\vee\ \mathcal{C}(\neg\ell^1)\ \ \mathcal{D}^1)$ and $(\vee\ \mathcal{C}(\neg\ell^2)\ \ \mathcal{D}^2)$, where $\neg\ell^i$ denotes a specific occurrence of $\neg\ell$. The simultaneous application of NC unit-resolution with two sub-formulas is formally expressed as follows:

$$\frac{\langle \ell : \alpha \rangle \quad , \quad \langle \Pi^1 \succ (\vee\ \mathcal{C}(\neg\ell^1)\ \ \mathcal{D}^1) : \beta^1 \rangle \quad , \quad \langle \Pi^2 \succ (\vee\ \mathcal{C}(\neg\ell^2)\ \ \mathcal{D}^2) : \beta^2 \rangle}{\langle \Pi^1 \succ \mathcal{D}^1 : \mathbf{min}\{\alpha, \beta^1\} \rangle \quad , \quad \langle \Pi^2 \succ \mathcal{D}^2 : \mathbf{min}\{\alpha, \beta^2\} \rangle}$$

If the sub-formula $\langle \Pi^i \succ (\vee\ \mathcal{C}(\neg\ell^i)\ \ \mathcal{D}^i) : \beta^i \rangle$ is denoted by $\langle \Pi, \mathcal{CD}(\neg\ell), \beta \rangle^i$ and the sub-formula $\langle \Pi^i \succ \mathcal{D}^i : \beta^i \rangle$ by $\langle \Pi, \mathcal{D}, \beta' \rangle^i$, where $\beta' = min\{\alpha, \beta^i\}$ for $i, 1 \le i \le k$, then *HUR* for k sub-formulas is formally expressed:

$$\frac{\langle \ell : \alpha \rangle \ , \ \langle \Pi, \mathcal{CD}(\neg\ell), \beta \rangle^1 \ , \ldots , \ \langle \Pi, \mathcal{CD}(\neg\ell), \beta \rangle^i \ , \ldots , \ \langle \Pi, \mathcal{CD}(\neg\ell), \beta \rangle^k}{\langle \Pi, \mathcal{D}, \beta' \rangle^1 \ , \ldots , \ \langle \Pi, \mathcal{D}, \beta' \rangle^i \ , \ldots , \ \langle \Pi, \mathcal{D}, \beta' \rangle^k} \ \underline{HUR}$$

Since $\neg\ell^i, 1 \le i \le k$, are literal occurrences that are pairwise different, so are the sub-formulas $\mathcal{CD}^i$, and so $\mathcal{D}^i$, in the numerator and denominator of *HUR*, respectively. However, the formulas Π^i are not necessarily different.

Example. The last concrete question and in general the working of the rule *HUR* is illustrated in Example 6.7.

Note. An NC hyper unit-resolution rule, more general than *HUR*, can be devised to include simultaneously $k \ge 2$ unit-clauses so that for each unit-clause $\langle \ell : \alpha \rangle$, one considers $n \ge 2$ sub-formulas $\langle \Pi^i \succ (\vee\ \mathcal{C}(\neg\ell^i)\ \ \mathcal{D}^i) : \beta^i \rangle$. In other words, one can consider applying simultaneously $k \ge 2$ *HUR* rules.

5.5. Finding the inconsistency degree

$\mathcal{UR}_\Sigma$ can determine **just one** subset of contradictory formulas, along with its inconsistency degree, whereas a given possibilistic base Σ can typically contain many contradictory subsets, each of them inducing the deduction of an empty formula $\langle \bot : \alpha \rangle$ with a specific weight α. Thus by definition of $\text{Inc}(\Sigma)$ and by Proposition 3.4, we have that:

$$\text{Inc}(\Sigma) = \max\{\alpha : \Sigma_{\ge\alpha} \text{is inconsistent}\} = \max\{\alpha \mid \Sigma \vdash \langle \bot : \alpha \rangle\}. \tag{7}$$

Below, we give two alternative methods to determine the inconsistency degree of Horn-NC bases. The first one is based on Dichotomic Search and calls to a propositional SAT (DS-SAT) solver, while the second one Finds the Inconsistency degree through calls to logical calculi $\mathcal{UR}_\Sigma$ (FI-UR). We demonstrate that both methods are polynomial for possibilistic Horn-NC bases and that neither of them outperforms the other for all such bases, to put it another way, their performance depends on the specific features of the Horn-NC base to be solved. Thus an efficient methodology for practical applications is likely to be derived from an appropriate amalgamation of both methods.

Dichotomic Search (DS-SAT). Just as for possibilistic clausal formulas [57], we can rank the weights α occurring in an input Horn-NC base Σ and then, with dichotomic search, check whether $\text{Inc}(\Sigma) > \alpha$ for some of such weights α. For that purpose, the strict α-cut $\Sigma_{>\alpha}$ of Σ is obtained and its propositional part $\Sigma^*_{>\alpha}$ is checked for consistency through the invocation of a solver for propositional Horn-NC formulas. Thus, if $\Sigma^*_{>\alpha}$ is found consistent (resp. inconsistent), then the next weight $\beta < \alpha$ (resp. $\beta > \alpha$) in the dichotomic search is selected, and subsequently, whether $\Sigma^*_{>\beta}$ is consistent is verified.

Dichotomic search invokes the propositional solver to a maximum of $\log m$ times, m being the number of different weights in Σ. Since propositional Horn-NC formulas are tested for consistency in polynomial time (see Proposition 5.10, Section 7.2), DS-SAT also needs just polynomial time to calculate the inconsistency degree of Horn-NC bases.

Along these lines, the following worst-case complexity of determining the inconsistency degree of *standard* possibilistic Horn bases can be easily established and which, however, had not been done up to now (to the best of our knowledge):

Proposition 5.15. *The worst-case complexity of DS-SAT to determine the inconsistency degree of standard Horn possibilistic bases* Σ *is* $O(n \times \log m)$*, where n is the size of* Σ^* *and m is the number of distinct weights in* Σ*.*

The proof follows immediately from the facts that dichotomic search needs only $\log m$ calls to a Horn-SAT solver and that testing the satisfiability of propositional Horn formulas was proven to be a strictly-linear complexity problem [25,55,37].

Practical Issues. In the case of Horn-NC bases, DS-SAT can skip easy and short proofs and, in some cases, become unsuitably redundant. Below, we first give counterexamples for DS-SAT on which DS-SAT is substantially outperformed by the second method FI-UR to be described, but subsequently, we will take the opposite direction and provide counterexamples for FI-UR on which DS-SAT notably improves FI-UR.

Example 5.16. Let us consider the next standard possibilistic Horn base:

$$\Sigma = \{\langle(\vee\ P_1\ \neg Q_1) : \mathbf{.125}\rangle, \langle(\vee\ Q_1 \neg Q_2) : \mathbf{.25}\rangle, \langle(\vee\ \neg Q_1\ Q_2) : \mathbf{.375}\rangle, \langle(\vee\ \neg Q_1\ \neg Q_2) : \mathbf{.5}\rangle,$$
$$\langle(\vee\ \neg P_1\ \neg P_2) : \mathbf{.625}\rangle, \langle(\vee\ P_1\ \neg P_2) : \mathbf{.625}\rangle, \langle(\vee\ \neg P_1\ Q_1) : \mathbf{.75}\rangle, \langle(\vee\ \neg P_1\ Q_2) : \mathbf{.875}\rangle,$$
$$\langle S_1 : \mathbf{1.}\rangle, \langle S_2 : \mathbf{1.}\rangle, \langle(\vee\ \neg S_1\ \neg S_2) : \mathbf{1.}\rangle\}$$

When applying DS-SAT to Σ, the following claims arise and are easily verifiable:

- Σ^* is Horn and so is Horn-NC.
- $\Sigma_{\geq 1.} = \{\langle S_1 : \mathbf{1.}\rangle, \langle S_2 : \mathbf{1.}\rangle, \langle(\vee\ \neg S_1\ \neg S_2) : \mathbf{1.}\rangle\} \subset \Sigma$ is inconsistent.
- $\forall \alpha > 0$: $\Sigma^*_{>\alpha}$ includes the inconsistent subset $\Sigma^*_{\geq 1.}$.
- The strict α-cuts checked for inconsistency are $\alpha \in \{.\mathbf{0}, .\mathbf{5}, .\mathbf{75}, .\mathbf{875}, \mathbf{1}.\}$.
- All five performed calls end for the same cause: $\Sigma^*_{\geq 1.}$ is inconsistent.

Therefore, the behavior of DS-SAT is heavily redundant for this example. □

The concrete Horn possibilistic base specified in Example 5.16 is parameterized and then generalized by the pattern of possibilistic Horn bases given next:

Example 5.17. Consider the next pattern of Horn bases (the $C_{i,j}$'s are Horn clauses):

$$\Sigma = \{\langle C_{1,1} : \alpha_1\rangle, \ldots, \langle C_{1,n_1} : \alpha_1\rangle, \ldots, \langle C_{m-1,1} : \alpha_{m-1}\rangle,$$
$$\ldots, \langle C_{m-1,n_{m-1}} : \alpha_{m-1}\rangle, \langle C_{m,1} : 1.\rangle, \langle C_{m,n_m} : 1.\rangle\}$$

in which we will assume that we have:

- $\Sigma^*_{\geq 1} = \{C_{m,1}, \ldots, C_{m,n_m}\}$ is inconsistent.
- For $1 \leq i \leq m-1$, $1 \leq j \leq n_i$, $C_{i,j}$, is not unitary.
- $\Sigma_{\geq 1}$ and $\Sigma/\Sigma_{\geq 1}$ do not share any literal.
- The weights α_i verify the following condition:

$$\alpha_i = 1/2^m + 2/2^m + \ldots + 2^i/2^m, \quad \text{for } 1 \leq i \leq m-1.$$

Proposition 5.18. *The worst-case complexity of DS-SAT for possibilistic Horn bases* Σ *having the pattern of Example 5.17 is* $O(n \times \log m)$*, where* $n = size(\Sigma^*_{\geq 1})$*.*

The proof is obtained following the reasoning sketched in Example 5.16.

Below, we describe the second method.

Finding Inconsistency via $\mathcal{UR}_\Sigma$ ***(FI-UR).*** We next tackle with algorithm FI-UR whose complexity is proven to be $O(n)$ for possibilistic bases having the pattern of Example 5.17, and so that it improves that of DS-SAT by a factor $\log m$. FI-UR relies on calls to logical calculi $\mathcal{UR}_\Sigma$, does not require any propositional solver and its principle is as follows.

Let us assume that FI-UR has deduced the first empty formula $\langle\bot : \alpha_1\rangle$, indicating that there is an inconsistent subset with degree α_1, and so that $\text{Inc}(\Sigma) \geq \alpha_1$. Then, FI-UR recursively verifies whether the strict α_1-cut $\Sigma_{>\alpha_1}$ is inconsistent, and again using $\mathcal{UR}_\Sigma$, attempts to deduce $\langle\bot : \alpha_2\rangle$. If so, obviously $\text{Inc}(\Sigma) \geq \alpha_2 > \alpha_1$ and the process goes on with the strict α_2-cut $\Sigma_{>\alpha_2}$. These operations are recursively performed until consistency is attained, and at that moment, FI-UR stops the search and returns the weight α_k of the last deduced formula $\langle \bot : \alpha_k \rangle$. The algorithm FI-UR specified below materializes the principle we have just described and must be called with its input value: **Inc** = 0.

FI-UR(Σ, Inc)

(1) *Apply* $\mathcal{UR}_\Sigma$ *to* Σ *adding resolvents to* Σ.

(2) *If* $\langle \bot : \alpha \rangle \in \Sigma$ *then:* $\Sigma_\alpha := \alpha\text{-}cut(\Sigma)$, $\mathbf{Inc} := \alpha$, $\mathbf{FI\text{-}UR(}\Sigma_\alpha, \mathbf{Inc)}$.

(3) *Else Return* **Inc**.

Lemma 5.19. *For any* $\Sigma \in \overline{\mathcal{H}}_\Sigma$, *we have: Inc*$(\Sigma)$ = *FI-UR*$(\Sigma, 0)$.

Lemma 5.20. *For any* $\Sigma \in \overline{\mathcal{H}}_\Sigma$, *FI-UR*$(\Sigma, 0)$ *ends in polynomial time in size*(Σ).

Example 5.21. Let us consider the next possibilistic Horn-NC base:

$$\Sigma = \{\langle(\vee\ P_1) : \mathbf{.5}\rangle, \langle\{\wedge\ \neg Q_1\ Q_2\} : \mathbf{.6}\rangle, \langle(\vee\ \{\wedge \neg P_1\ \neg P_2\}\ \{\wedge\ Q_1\ Q_2\}) : \mathbf{.4}\rangle,$$
$$\langle(\vee\ \neg Q_1) : \mathbf{.5}\rangle, \langle(\vee\ Q_1\ \neg Q_2) : \mathbf{.6}\rangle, \langle(\vee\ \neg Q_1\ Q_2) : \mathbf{.5}\rangle, \langle(\vee\ Q_2) : \mathbf{.5}\rangle,$$

Below, we specify step-by-step the operations executed by FI-UR running on Σ:

- In step (1), $\mathcal{UR}_\Sigma$ picks up $\langle(\vee\ P_1) : \mathbf{.5}\rangle$ and $\langle(\vee\ \{\wedge \neg P_1\ \neg P_2\}\ \{\wedge\ Q_1\ Q_2\}) : \mathbf{.4}\rangle$.
- UR_Σ deduces $\langle(\vee\ \{\wedge\ Q_1\ Q_2\}) : \mathbf{.4}\rangle$ and adds it to Σ.
- $\langle(\vee\ \{\wedge \neg P_1\ \neg P_2\}\ \{\wedge\ Q_1\ Q_2\}) : \mathbf{.4}\rangle$ is now subsumed and so canceled.
- After simplifications, $\langle\{\wedge\ Q_1\ Q_2\} : \mathbf{.4}\rangle$ is added to Σ.
- Rule $\odot\phi$ with $\langle\{\wedge\ Q_1\ Q_2\} : \mathbf{.4}\rangle$ adds $\langle Q_1 : \mathbf{.4}\rangle$ and $\langle Q_2 : \mathbf{.4}\rangle$.
- Rule $\odot\phi$ with $\langle\{\wedge\ \neg Q_1\ Q_2\} : \mathbf{.6}\rangle$ adds $\langle \neg Q_1 : \mathbf{.6}\rangle$ and $\langle Q_2 : \mathbf{.6}\rangle$.
- UR_Σ with $\langle \neg Q_1 : \mathbf{.6}\rangle$ and $\langle Q_1 : \mathbf{.4}\rangle$ deduces $\langle \bot : \mathbf{.4}\rangle$.
- In step (2), the obtained strict **.4**-cut is:

$$\Sigma_{>.4} = \{\langle(\vee\ \neg Q_1) : \mathbf{.5}\rangle, \langle(\vee\ Q_1\ \neg Q_2) : \mathbf{.6}\rangle, \langle(\vee\ \neg Q_1\ Q_2) : \mathbf{.5}\rangle, \langle(\vee\ Q_2) : \mathbf{.5}\rangle\}$$

- FI-UR($\Sigma_{>.4}$, **.4**) is recursively called.
- In step (1), the reiterative application of UR_Σ leads to $\langle \bot : \mathbf{.5}\rangle$.
- In step (2), the obtained strict **.5**-cut is: $\Sigma_{>.5} = \{\langle(\vee\ Q_1\ \neg Q_2) : \mathbf{.6}\rangle\}$.
- FI-UR($\Sigma_{>.5}$, **.5**) is recursively called.
- In step (1), $\mathcal{UR}_\Sigma$ cannot deduce any $\langle \bot : \alpha \rangle$.
- Step (2) is skipped.
- In step (3), FI-UR returns **Inc= .5**. □

Example 6.8, which is the continuation of Example 6.5, also shows the process followed by FI-UR that interleaves inference rules of $\mathcal{UR}_\Sigma$ with α-cuts, in order to determine the inconsistency degree of a possibilistic Horn-NC base.

Algorithm FI-UR guides its deduction search *through unit-clauses*, which allows to it to become quicker than DS-SAT for certain Horn-NC bases. We next check this claim by employing the possibilistic bases in Examples 5.16 and 5.17:

Example 5.16. FI-UR, running on Σ in Example 5.16, works as follows:

– In step (1), $\mathcal{UR}_\Sigma$ is applied to Σ.

– Since $\mathcal{UR}_\Sigma$ requires unit clauses, only $\langle(\vee\ \neg S_2) : \mathbf{1.}\rangle$ and $\langle \bot : \mathbf{1.}\rangle$ are deduced.

– In step (2), we have: $\Sigma_{>1} = \emptyset$ and $\mathbf{Inc} := 1$

– FI-UR($\emptyset$, **1.**) is recursively called.

– In step (1), no inference is performed.

– Step (2) is skipped.

– In step (3), FI-UR returns **Inc** $= 1$. □

Consequently, FI-UR, which executes just once NC unit-resolution and exclusively on formulas in $\Sigma_{\geq 1}$, outperforms DS-SAT in the solving of Example 5.16. This improvement is more formally and accurately generalized by applying FI-UR to Example 5.17:

Proposition 5.22. *The worst-case complexity of FI-UR to determine the inconsistency degree of Horn bases Σ with the pattern of Example 5.17 is $O(n)$, where $n = size(\Sigma^*_{\geq 1})$.*

Proof. (Sketch) The proof follows from the following facts:

– $\Sigma^*_{\geq 1.}$ is the unique subset with unit-clauses, by the assumptions in Example 5.17.

– Resolvents subsume their non-unitary "father" clause because:

– The resolvent inherits the weight of its non-unitary "father".

– Hence, FI-UR keeps constant the number of formulas in $\Sigma_{\geq 1}$ and Σ.

– $\mathcal{UR}_\Sigma$ recovers classical unit-resolution if applied to Horn bases.

– The same datastructure of linear Horn-SAT solvers [25,55,37] can be used. □

Even though FI-UR is more efficient than DS-SAT on the previous bases, it is not the case for all Horn-NC bases: for some of them, the behavior of both methods is the inverse one. In fact, $\mathcal{UR}_\Sigma$ can add a quadratic number of formulas to Σ in some detrimental cases, and so, be heavily defeated by DS-SAT. We give a counterexample for FI-UR next:

Example 5.23. Let us study the next possibilistic Horn base:

$$\Sigma = \{\langle(\vee \neg P_1 \ \ldots \ \neg P_m) : \alpha\rangle, \quad \langle(\vee \ P_1) : \beta_1\rangle, \ldots, \langle(\vee \ P_m) : \beta_m\rangle\}$$

where we suppose: $\beta_1 < \ldots < \beta_i < \ldots < \beta_m < \alpha$.

(1) In a first stage, $\mathcal{UR}_\Sigma$ can deduce m formulas of the kind:

$$C_i = \langle(\vee \neg P_1 \ldots \neg P_{i-1} \neg P_{i+1} \ldots \neg P_m) : \beta_i\rangle, \text{ for } 1 \leq i \leq m,$$

whose parent $\langle(\vee \neg P_1 \ldots \neg P_i \ldots \ \neg P_m) : \alpha\rangle$ is not subsumed because $\beta_i < \alpha$.

(2) In a second stage, for each C_i issued from the first stage and the unit formulas $\langle(\vee \ P_j) : \beta_j\rangle, \ i \neq j$, belonging to Σ, $\mathcal{UR}_\Sigma$ can deduce formulas of type:

$$C_{i,j} = \langle(\vee \neg P_1 \ldots \neg P_{i-1} \neg P_{i+1} \ldots \neg P_{j-1} \neg P_{j+1} \ldots \neg P_m) : \beta_{i,j}\rangle,$$

where $\beta_{i,j} = min\{\beta_i, \beta_j\}$. In such a case, we have that:

– if $\beta_i > \beta_j$ then $\beta_{i,j} = \beta_j$ and $C_{i,j}$ does not subsume C_i; however

– if $\beta_i < \beta_j$ then $\beta_{i,j} = \beta_i$ and $C_{i,j}$ subsumes C_i;

In the latter case, C_i could be removed if a subsumption rule was added to $\mathcal{UR}_\Sigma$.

(3) Therefore, after having applied unit-resolution k times, one obtains formulas:

$$C_{i,j} = \langle(\vee \neg P'_1 \ldots\ldots \neg P'_k) : \beta'_{k+1,\ldots,m}\rangle, \text{ where}$$

$$\{P'_1, \ldots P'_k, P'_{k+1} \ldots, P'_m\} = \{P_1, \ldots, P_m\}, \quad \beta'_{k+1,\ldots,m} = \min\{\beta'_{k+1}, \ldots, \beta'_m\}$$

and where β'_{k+j} comes from a formula $\langle(\vee \ P'_{k+j}) : \beta'_{k+j}\rangle \in \Sigma$.

It is not hard to check (we omit the details for space saving purposes) that if the unitary formulas are selected in decreasing order w.r.t. their weights, that is, first $\langle(\vee \ P_m) : \beta_m\rangle$, then $\langle(\vee \ P_{m-1}) : \beta_{m-1}\rangle$, and so on, the number of generated clauses is:

$$m^2 < \sum_{k=1}^{m} k \times (m-k) < m^3$$

whereas if they are chosen in increasing order, then such a number is bounded by:

$$\sum_{k=1}^{m} k = m \times (m-1)/2$$

and the latter quantity of clauses belongs to $O(m^2)$. □

Proposition 5.24. *The worst-case complexity of FI-UR to determine the inconsistency degree of possibilistic bases* Σ *from Example 5.23, is* $O(m^2)$*, where* m *is the maximum number of literals in a specific clause in* Σ^**.*

On the contrary, for bases specified in Example 5.23, DS-SAT warrants an $O(n \times \log m)$ complexity by Proposition 5.18, and so it ameliorates algorithm FI-UR. Besides, please remark that Proposition 5.24 entails that FI-UR is at least quadratic for standard Horn bases, so it is outperformed by DS-SAT as well.

Altogether, high practical efficiency is probably achievable by a suitable combination of both kinds of searching principles. We believe that one of such suitable integrations could rely on using DS-SAT to guide the global search and applying FI-UR to certain local sub-formulas encountered in the searching process. Nonetheless, this aspect demands further investigation.

5.6. Reasoning by contradiction

In classical logic, to know whether a question φ follows from a knowledge base Σ, one reasons by contradiction and checks whether $\Sigma \wedge \neg\varphi$ is inconsistent. In possibilistic logic [29,57], one adds $\langle \neg\varphi : \mathbf{1}. \rangle$ to Σ and searches $\text{Inc}(\Sigma \cup \{\langle \neg\varphi : \mathbf{1}. \rangle\})$.

With clausal bases, if φ is a literal conjunction then $\neg\varphi$ is a clause, and if φ is a clause then $\langle \neg\varphi : \mathbf{1}. \rangle$ is equivalent to a set of clauses $\langle \ell : \mathbf{1}. \rangle$, where ℓ is a literal. Hence, in both cases, in order to determine $\text{Inc}(\Sigma \cup \{\langle \neg\varphi : \mathbf{1}. \rangle\})$, we can use the same inference mechanisms which allow the determination of the inconsistency of clausal bases.

With Horn-NC bases, that advantage is lost because if the question φ is Horn-NC then $\neg\varphi$ is not Horn-NC (it is dual Horn-NC). Then, there exist two possibilities briefly discussed below to determine $\text{Inc}(\Sigma \cup \{\langle \neg\varphi : \mathbf{1}. \rangle\})$, being both options polynomial-time solvable and making use of only a solver for propositional Horn-NC formulas.

The first one consists of language restriction for questions φ. Concretely, allowing only positive formulas for φ would entail that $\neg\varphi$ would be negative and so Horn-NC as well. Note that some sort of limitation also exists in the clausal setting: one cannot submit as a question a clausal formula, and yet, clausal formulas form the language of the clausal bases. So, one cannot use for questions φ the full language used for bases.

The second possibility allows for handling the same Horn-NC language for bases and for questions φ. It takes advantage that consistent Horn-NC bases have only one minimal model (Corollary 4.15) and its first step consists in determining the inconsistency degree $\text{Inc}(\Sigma) = \alpha$, which means that $\Sigma^*_{>\alpha}$ is consistent and also that $\text{Inc}(\Sigma \cup \{\langle \neg\varphi : \mathbf{1}. \rangle\}) \geq \alpha$.

Subsequently, by using non-clausal unit-resolution one can obtain the minimal model of $\Sigma^*_{>\alpha}$, just as it can be done in the clausal setting with clausal unit-resolution and Horn clausal formulas, and then evaluate φ within it.

According to the value of φ within the minimal model of $\Sigma^*_{>\alpha}$ we have:

(1) True entails $\Sigma^*_{>\alpha} \models \varphi$; so $\Sigma^*_{>\alpha} \wedge \neg\varphi$ is inconsistent.

(2) False entails $\Sigma^*_{>\alpha} \models \neg\varphi$ and thus $\Sigma^*_{>\alpha} \not\models \varphi$ and $\Sigma^*_{>\alpha} \wedge \neg\varphi$ is consistent.

(3) Undefined entails $\Sigma^*_{>\alpha}$ and φ share models but $\Sigma^*_{>\alpha} \not\models \varphi$ and $\Sigma^*_{>\alpha} \wedge \neg\varphi$ is consistent.

Case (1): if the minimal model of $\Sigma^*_{>\alpha}$ evaluates φ to true, then we have that both $\Sigma^*_{>\alpha}$ is consistent and $\Sigma^*_{>\alpha} \wedge \neg\varphi$ is inconsistent. Therefore, $\text{Inc}(\Sigma \cup \{\langle \neg\varphi : \mathbf{1}. \rangle\}) > \alpha$ is the smallest weight β occurring in $\Sigma^*_{>\alpha}$.

Case (3): in case the minimal model of $\Sigma^*_{>\alpha}$ evaluates φ to undefined, then $\Sigma^*_{>\alpha} \wedge \neg\varphi$ is consistent and then one can check that: $\text{Inc}(\Sigma \cup \{\langle \neg\varphi : \mathbf{1}. \rangle\}) = \text{Inc}(\Sigma) = \alpha$.

Case (2): if the minimal model of $\Sigma^*_{>\alpha}$ evaluates φ to false, then it is still possible that for a certain $\beta > \alpha$, we could have $\Sigma^*_{>\beta} \not\models \neg\varphi$ and even that $\Sigma^*_{>\beta} \models \varphi$. Thus, the process is continued searching for a bigger β-cut such that the minimal model of $\Sigma^*_{>\beta}$ evaluates φ to true or undefined. In the former case and by Case (1) above, $\text{Inc}(\Sigma \cup \{\langle \neg\varphi : \mathbf{1}. \rangle\}) = \beta$, and in the latter case and by Case (3) above, $\text{Inc}(\Sigma \cup \{\langle \neg\varphi : \mathbf{1}. \rangle\}) = \alpha$.

Example 5.25. Assume that for $\text{Inc}(\Sigma) = \mathbf{0.5}$ and that $\Sigma_{>\mathbf{0.5}}$ and φ are as follows:

$$\Sigma_{>\mathbf{0.5}} = \{\, \langle a, \mathbf{0.6} \rangle,\ \langle b, \mathbf{0.7} \rangle,\ \langle (\vee\ \neg a\ d), \mathbf{0.8} \rangle \,\} \qquad \varphi = [\wedge\ \neg a\ \ (\vee\ b\ \neg d)\,]$$

One can obtain with propositional NC unit-resolution that the minimal model of $\Sigma^*_{>\mathbf{0.5}}$ is $\{a, b, d\}$ and that φ is evaluated to false within it. Hence, as mentioned above, the search continues with a value $\beta > 0.5$. Then the minimal model of $\Sigma^*_{>\mathbf{0.6}}$ is $\{b\}$ in which φ is true. Thus, by applying Case (1) above, we have that $\mathrm{Inc}(\Sigma \cup \{\langle \neg\varphi : \mathbf{1}\rangle\}) = \mathbf{0.7}$.

Assume that we maintain the same base $\Sigma_{>\mathbf{0.5}}$ but that now $\varphi = [\wedge\ \neg a\ \neg b\ \ (\vee\ b\ \neg d)]$. We have: (i) the minimal model $\{a, b, d\}$ of $\Sigma_{>\mathbf{0.5}}$ evaluates φ to false; (ii) the minimal model $\{b\}$ of $\Sigma^*_{>\mathbf{0.6}}$ evaluates φ to false; and (iii) the minimal model $\emptyset$ of $\Sigma^*_{>\mathbf{0.7}} = \{\langle(\vee\ \neg a\ d), \mathbf{0.8}\rangle\}$ evaluates φ to undefined. Hence, we have to apply Case (3) above, which leads to $\mathrm{Inc}(\Sigma \cup \{\langle \neg\varphi : \mathbf{1}\rangle\}) = \mathbf{0.5}$. □

The complexity of determining $\mathrm{Inc}(\Sigma \cup \{\langle \neg\varphi : \mathbf{1}\rangle\})$ is clearly polynomial because:

(1) evaluating a non-clausal formula ϕ in a specific interpretation takes only linear time;

(2) the propositional Horn-NC solver ends in polynomial time (see Section 7.2); and

(3) the number of dichotomic steps is at most logarithmic.

6. Illustrative examples

The notions defined or discussed above are illustrated by the following examples:

- Example 6.1: a simple inconsistent Horn-NC base.
- Example 6.2: a simple consistent Horn-NC base.
- Example 6.3: an inconsistent propositional Horn-NC formula.
- Example 6.4: a base with an inconsistent propositional formula.
- Example 6.5: a complete Horn-NC base.
- Example 6.6: NC Local Unit-Resolution.
- Example 6.7: NC Hyper Unit-Resolution.
- Example 6.8: algorithm **Find**.

We highlight Examples 6.5 and 6.8 as a rather complete Horn-NC base solved in two phases: the first one in Example 6.5 and the second one in Example 6.8.

Example 6.1. Let us assume the next possibilistic Horn-NC base:

$$\Sigma_0 = \{\,\langle P : \mathbf{0.8}\rangle,\ \langle \varphi = (\vee_2\ [\wedge\ \neg P\ \ \neg Q\,]\ \ Q\,) : \mathbf{0.6}\rangle,\ \langle (\vee\ \neg P\ \neg Q\,) : \mathbf{0.7}\rangle\}$$

- UR_Σ with $\langle P : \mathbf{0.8}\rangle$ and φ gives rise to the next matchings:

$$\Pi = \varphi = (\vee\ \ \mathcal{C}(\neg P)\ \mathcal{D}\,) \qquad \mathcal{C}(\neg P) = [\wedge\ \ \neg P\ \ \neg Q\,] \qquad \mathcal{D} = Q$$

- Hence, UR_Σ adds: $\Sigma_1 \leftarrow \Sigma_0\ \cup\ \langle(\vee\ \ Q\,) : \mathbf{0.6}\rangle$
- Applying simplifications to the last formula: $\Sigma_2 \leftarrow \Sigma_1\ \cup\ \langle Q : \mathbf{0.6}\rangle$
- UR_Σ with $\langle Q : \mathbf{0.6}\rangle$ and with the last formula in Σ_0 gives:

$$\Pi = (\vee\ \neg P\ \neg Q\,) = (\vee\ \ \mathcal{C}(\neg Q\,)\ \mathcal{D}\,) \qquad \mathcal{C}(\neg Q\,) = \neg Q \qquad \mathcal{D} = \neg P$$

- Hence, UR_Σ adds: $\Sigma_3 \leftarrow \Sigma_2\ \cup\ \langle(\vee\ \neg P\,) : \mathbf{0.6}\rangle$
- Resolving $\langle P : \mathbf{0.8}\rangle$ in Σ_0 with the last added formula: $\Sigma_4 \leftarrow \Sigma_3\ \cup \langle(\vee) : \mathbf{0.6}\rangle$
- Therefore $\mathcal{UR}_\Sigma$ obtains $\mathrm{Inc}(\Sigma) = \mathbf{0.6}$ □

Example 6.2. Let us assume that Σ_0 is the next possibilistic Horn-NC base:

$$\Sigma_0 = \{\,\langle Q : \mathbf{0.8}\rangle,\ \langle \varphi = (\vee_2\ \neg Q\ \ [\wedge_2\ R\ \ (\vee_2\ \neg Q\ \ [\wedge\ S\ \neg P\,]\,)\,]\,) : \mathbf{0.6}\rangle,\ \langle (\vee\ \neg P\ \neg Q\,) : \mathbf{0.7}\rangle\}$$

- UR_Σ with $\langle Q : \mathbf{0.8}\rangle$ and with the rightest $\neg Q$ in φ gives the next matchings:
  - $\Pi = \varphi$;
  - $(\vee\ \mathcal{C}(\neg Q\,)\ \mathcal{D}\,) = (\vee_2\ \neg Q\ \ [\wedge\ S\ \neg P\,]\,)$;
  - $\mathcal{C}(\neg Q\,) = \neg Q\,$;
  - $\mathcal{D} = [\wedge\ S\ \neg P\,]$.

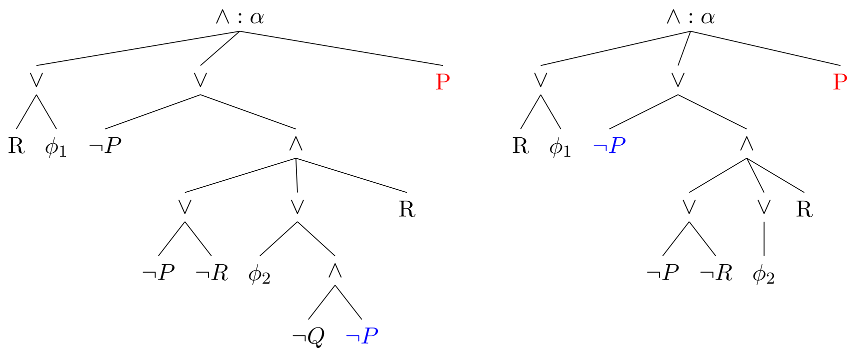

Fig. 1. Formulas φ (left) and φ' (right).

- Hence, UR_Σ adds: $\Sigma_1 \leftarrow \Sigma_0 \cup \langle (\vee_2\ \neg Q\ [\wedge_2\ R\ (\vee_1\ [\wedge\ S\ \neg P])]) : \mathbf{0.6} \rangle$
- After simplifications: $\Sigma_2 \leftarrow \Sigma_1 \cup \langle (\vee_2\ \neg Q\ [\wedge\ R\ S\ \neg P]) : \mathbf{0.6} \rangle$
- Using again $\langle Q : \mathbf{0.8} \rangle$ and the last formula in Σ_2, we have the matchings:
  - $\Pi = (\vee_2\ \neg Q\ [\wedge\ R\ S\ \neg P]) = (\vee\ \mathcal{C}(\neg Q)\ \mathcal{D})$
  - $\mathcal{C}(\neg Q) = \neg Q$
  - $\mathcal{D} = [\wedge\ R\ S\ \neg P]$
- Hence, UR_Σ adds: $\Sigma_3 \leftarrow \Sigma_2 \cup \langle (\vee_1\ [\wedge\ R\ S\ \neg P]) : \mathbf{0.6} \rangle$
- After simplifications: $\Sigma_4 \leftarrow \Sigma_3 \cup \langle [\wedge\ R\ S\ \neg P] : \mathbf{0.6} \rangle$
- Applying *MinD*: $\Sigma_5 \leftarrow \Sigma_4 \cup \{\langle R : \mathbf{0.6} \rangle, \langle S : \mathbf{0.6} \rangle, \langle \neg P : \mathbf{0.6} \rangle\}$.
- Using the first and last formulas in Σ_0: $\Sigma_6 \leftarrow \Sigma_5 \cup \langle (\vee\ \neg P) : \mathbf{0.7} \rangle$
- Applying *MaxN* with $\langle \neg P : \mathbf{0.6} \rangle$ and $\langle (\vee\ \neg P) : \mathbf{0.7} \rangle$ the former is eliminated.
- Since no more resolvents apply, Σ_6 is consistent, and so $\mathrm{Inc}(\Sigma) = 0$. □

Next, we give a rather elaborated propositional formula and show how the propositional NC unit-resolution, or UR_P, together with the simplification rules, detect its inconsistency.

Example 6.3. Let us assume that the input Σ has a possibilistic Horn-NC $\langle \varphi : \alpha \rangle$ given below, where ϕ_1 and ϕ_2 are assumed to be Horn-NC formulas.

$$\langle \varphi = [\wedge_3\ (\vee\ R\ \phi_1)\ (\vee_2\ \neg P\ [\wedge_3\ (\vee\ \neg P\ \neg R)\ \ (\vee_2\ \phi_2\ [\wedge\ \neg Q\ \neg P])\ R])\ P] : \alpha \rangle$$

The tree associated with φ is depicted in **Fig. 1**, on the left. Thus, before computing the inconsistency degree of Σ, one needs to check whether its propositional formulas are inconsistent. We show below how UR_P checks the inconsistency of φ. UR_P with P and the right-most $\neg P$ yields the next matchings in the UR_P numerator:

- $\Pi = (\vee_2\ \neg P\ \ [\wedge_3\ (\vee\ \neg P\ \neg R)\ \ (\vee_2\ \phi_2\ [\wedge\ \neg Q\ \neg P])\ \ R])$
- $(\vee\ \mathcal{C}(\neg P)\ \mathcal{D}) = (\vee_2\ \phi_2\ [\wedge\ \neg Q\ \neg P])$
- $\mathcal{C}(\neg P) = [\wedge\ \neg Q\ \neg P]$
- $\mathcal{D} = \phi_2$

Applying UR_P to φ yields:

$$\varphi' = \langle [\wedge_3\ (\vee\ R\ \phi_1)\ (\vee_2\ \neg P\ [\wedge_3\ (\vee\ \neg P\ \neg R)\ \ (\vee\ \phi_2)\ R])\ P] : \alpha \rangle$$

The resulting tree is the right one in **Fig. 1**. Assume that we proceed now with a second NC unit-resolution step by picking the same P and the left-most $\neg P$ (colored blue in **Fig. 1**, on the right). Then, the right conjunct of the numerator of UR_P is as follows:

- $\Pi = (\vee_2\ \neg P\ \ [\wedge_3\ (\vee\ \neg P\ \neg R)\ (\vee\ \phi_2)\ \ R])$

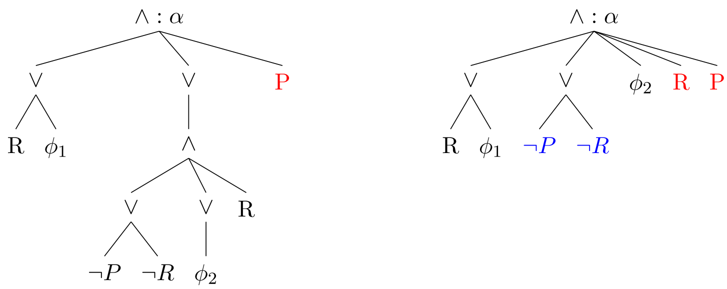

Fig. 2. Example 6.3 continued.

- $(\vee\ \mathcal{C}(\neg P)\ \mathcal{D}) = \Pi$
- $\mathcal{C}(\neg P) = \neg P$
- $\mathcal{D} = [\wedge_3\ \ (\vee\ \neg P\ \ \neg R\,)\ \ (\vee\ \phi_2\,)\ \ R\,]$

By applying UR_P to φ', the obtained formula is depicted in **Fig. 2**, on the left.

After three simplifications, one gets the formula associated with the right tree in **Fig. 2.** Finally, two applications of UR_P to the two pairs R and $\neg R$, and P and $\neg P$, lead the calculus to derive $\langle(\vee) : \alpha\rangle$. □

In the next example, we illustrate the effects of Proposition 5.8.

Example 6.4. Let φ be the formula from Example 6.3 and Σ_1 be Σ from Example 6.1 and let us analyze the base $\Sigma = \Sigma_1 \cup \{\langle \varphi : \mathbf{0.6}\rangle\}$. Then, firstly the propositional rules of $\mathcal{UR}_\Sigma$ are applied to each propositional Horn-NC in Σ, and in particular, to $\langle \varphi : \mathbf{0.6}\rangle$, which, according to Example 6.3, yields $\langle(\vee) : \mathbf{0.6}\rangle$. Then Σ_1 is reduced to $\Sigma_1 = \{\langle P : \mathbf{0.8}\rangle,\ \langle\,(\vee\ \neg P\ \neg Q\,) : \mathbf{0.7}\rangle\}$. Since Σ_1 is consistent, one can conclude that $\text{Inc}(\Sigma) = \mathbf{0.6}$.

We next give a complete formula and illustrate how $\mathcal{UR}_\Sigma$ determines just one inconsistent subset $\Sigma' \subseteq \Sigma$ and its degree $\text{Inc}(\Sigma')$. By now, we are not concerned with finding $\text{Inc}(\Sigma)$, but just in finding one inconsistent subset. Later, in Example 6.8, we will illustrate the process performed by **FI-UR** to obtain $\text{Inc}(\Sigma)$.

Example 6.5. Let us assume that Σ_0 is the next possibilistic Horn-NC base:

$$\Sigma_0 = \{\,\langle P : \mathbf{0.8}\rangle,\ \langle \varphi 1 : \mathbf{0.6}\rangle,\ \langle \varphi_2 : \mathbf{0.5}\rangle,\ \langle\,[\wedge\ \neg P\ \neg Q\,] : \mathbf{0.7}\rangle\}$$

wherein the propositional formulas φ_1 and φ_2, both individually consistent, are as follows:

- $\varphi_1 = (\vee\ \ [\wedge\ \ \neg P\ \ \neg Q\,]\ \ [\wedge\ \ Q\ \ \ P])$
- $\varphi_2 = (\vee\ \ \neg Q\ \ [\wedge\ \ R\ \ (\vee\ \ \neg Q\ \ [\wedge\ \ S\ \ \neg P])\,]\,)$

The input base Σ is inconsistent and below, we step-by-step provide the inferences carried out by the calculus $\mathcal{UR}_\Sigma$ to derive one empty formula $\langle\bot : \alpha\rangle$.

- We apply UR_Σ with $\langle P : \mathbf{0.8}\rangle$ and $\langle \varphi_1 : \mathbf{0.6}\rangle$ and the next matchings:
  - $\Pi = \varphi_1 = \ (\vee\ \ \mathcal{C}(\neg P)\ \ \mathcal{D})$
  - $\mathcal{C}(\neg P) = [\wedge\ \neg P\ \neg Q\,]$
  - $\mathcal{D} = [\wedge\ \ Q\ \ P]$
- Hence, UR_Σ adds: $\Sigma_1 \leftarrow \Sigma_0\ \cup\ \langle\,(\vee\ \ [\wedge\ \ Q\ \ P]) : \mathbf{0.6}\rangle$
- Simplifying the last formula: $\Sigma_2 \leftarrow \Sigma_1\ \cup\ \langle\,[\wedge\ \ Q\ \ P] : \mathbf{0.6}\rangle$
- Applying *MinD* to the last formula: $\Sigma_3 \leftarrow \Sigma_2\ \cup\ \langle\ Q\ \ : \mathbf{0.6}\rangle \cup \langle P : \mathbf{0.6}\rangle$
- Since $\langle P : \mathbf{0.8}\rangle, \langle P : \mathbf{0.6}\rangle \in \Sigma$, by *MaxN*: $\Sigma_4 \leftarrow \Sigma_3 / \langle P : \mathbf{0.6}\rangle$
- Applying UR_Σ with $\langle\ Q\ \ : \mathbf{0.6}\rangle$ and the rightest $\neg Q$ of $\langle \varphi_2 : \mathbf{0.5}\rangle$:

$$\Pi = \varphi_2\ \ (\vee\ \mathcal{C}(\neg Q\,)\ \mathcal{D}) = (\vee_2\ \neg Q\ \ [\wedge\ S\ \neg P])\ \ \mathcal{C}(\neg Q\,) = \neg Q\ \ \ \mathcal{D} = [\wedge\ S\ \neg P]$$

- Thus UR_Σ adds: $\Sigma_5 \leftarrow \Sigma_4 \,\cup\, \langle (\vee_2 \;\neg Q\; [\wedge_2\; R\; (\vee_2\; [\wedge\; S\; \neg P])\,])\,{:}\,\mathbf{0.5}\rangle$
  - We denote the last added formula by $\langle \varphi_3 : \mathbf{0.5}\rangle$.
- Applying UR_Σ with again $\langle\, Q : \mathbf{0.6}\rangle$ and $\langle \varphi_3 : \mathbf{0.5}\rangle$:
  - $\Pi = \varphi_3 = (\vee\; \mathcal{C}(\neg Q)\; \mathcal{D}$
  - $\mathcal{C}(\neg Q) = \neg Q$
  - $\mathcal{D} = [\wedge_2\; R\; (\vee_1\; [\wedge\; S\; \neg P])]$
- Hence UR_Σ adds: $\Sigma_6 \leftarrow \Sigma_5 \,\cup\, \langle (\vee_1\; [\wedge_2\; R\; (\vee_1\; [\wedge\; S\; \neg P])\,])\,{:}\,\mathbf{0.5}\rangle$
- Simplifying the last formula: $\Sigma_7 \leftarrow \Sigma_6 \,\cup\, \langle [\wedge\; R\; S\; \neg P] : \mathbf{0.5}\rangle$
- Using the rule *InvMinD:* $\Sigma_8 \leftarrow \Sigma_7 \,\cup\, \{\langle R : \mathbf{0.5}\rangle, \langle S : \mathbf{0.5}\rangle, \langle \neg P : \mathbf{0.5}\rangle\}$
- From $\langle \neg P : \mathbf{0.5}\rangle$ and $\langle P : \mathbf{0.8}\rangle$ in the initial Σ_0: $\Sigma_9 \leftarrow \Sigma_8 \cup \langle (\vee) : \mathbf{0.5}\rangle$.
- So the (first) inconsistency degree found is **0.5.** □

Example 6.6. Consider again φ from Example 6.3: One can check that its sub-formula

$$\phi = (\vee_2\; \neg P\; [\wedge_3\; (\vee\; \neg P\; \neg R)\; (\vee_2\; \phi_2\; [\wedge\; \neg Q\; \neg P])\; R])$$

has the pattern of the *LUR* numerator regarding $\neg R$ and R. Thus *LUR* can be applied and so ϕ be replaced, after simplifications, with $(\vee_2\; \neg P\; [\wedge_3\; \neg P\; (\vee_2\; \phi_2\; [\wedge\; \neg Q\; \neg P])\; R])$ in φ. In this specific example, only one literal is removed, but in a general case, big sub-formulas may be eliminated. □

Example 6.7. Let us reconsider also the formula in previous Example 6.3 (recall that the i superscript in literal ℓ^i denotes an specific literal occurrence of ℓ):

$$\langle\, [\wedge_3\; (\vee\; R\, \phi_1)\; (\vee_2\; \neg P^1\; [\wedge_3\; (\vee\; \neg P^2\; \neg R)\; (\vee_2\; \phi_2\; [\wedge\; \neg Q\; \neg P^3])\; R])\; P\,] : \alpha\,\rangle$$

One can apply NC Hyper Unit-Resolution with P and the three literals $\neg P^i$. The formula Π in the numerator of *HUR* is the same for the three literals, so it is denoted by $\Pi^{1,2,3}$, but the formulas $(\vee\; \mathcal{C}(\neg P^i)\; \mathcal{D}^i)$ are different and are given below:

- $\Pi^{1,2,3} = (\vee_2\; \neg P\; [\wedge_3\; (\vee\; \neg P\; \neg R)\; (\vee_2\; \phi_2\; [\wedge\; \neg Q\; \neg P])\; R])$
- $(\vee\; \mathcal{C}(\neg P^1)\; \mathcal{D}^1) = \Pi^{1,2,3}$
- $(\vee\; \mathcal{C}(\neg P^2)\; \mathcal{D}^2) = (\vee\; \neg P^2\; \neg R)$
- $(\vee\; \mathcal{C}(\neg P^3)\; \mathcal{D}^3) = (\vee_2\; \phi_2\; [\wedge\; \neg Q\; \neg P^3])$

By applying NC Hyper Unit-Resolution, one gets:

$$\langle\, [\wedge_3\; (\vee\; R\, \phi_1)\; (\vee_1\; [\wedge_3\; (\vee\; \neg R)\; (\vee\; \phi_2)\; R])\; P\,] : \alpha\,\rangle$$

After simplifying:

$$\langle\, [\wedge_5\; (\vee\; R\, \phi_1)\; \neg R\; \phi_2\; R\; P\,] : \alpha\,\rangle$$

Clearly, a simple NC unit-resolution deduces $\langle (\vee) : \alpha\rangle$. Altogether, in this particular example, the rule *HUR* accelerates considerably the proof of inconsistency.

Example 6.8. Let us continue with Example 6.5. Since $\langle (\vee) : \mathbf{0.5}\rangle$ was found, for checking whether $\mathrm{Inc}(\Sigma) > \mathbf{0.5}$, all possibilistic formulas whose necessity weight is not strictly bigger than **0.5** are useless, that is, one can obtain the strict **0.5**-cut of Σ. Thus, the new base is $\Sigma_{>\mathbf{0.5}} = \Sigma' \cup \Sigma''$, where Σ' and Σ'' are the strict **0.5**-cut of the initial formulas and of the deduced formulas, respectively, and which are given below:

$$\Sigma' = \{\langle P : \mathbf{0.8}\rangle,\; \langle \varphi_1 : \mathbf{0.6}\rangle,\; \langle [\wedge\; \neg P\; \neg Q] : \mathbf{0.7}\rangle\}$$

$$\Sigma'' = \{\langle (\vee\; [\wedge\; Q\; P]) : \mathbf{0.6}\rangle,\; \langle [\wedge\; Q\; P] : \mathbf{0.6}\rangle,\; \langle\, Q\; : \mathbf{0.6}\rangle, \}$$

One can check that, since $\langle P : \mathbf{0.8}\rangle$ and $\langle\, Q\; : \mathbf{0.6}\rangle$ belong to $\Sigma_{>\mathbf{0.5}} = \Sigma' \cup \Sigma''$, then the only non-subsumed formulas are $\{\langle P : \mathbf{0.8}\rangle,\; \langle\, Q : \mathbf{0.6}\rangle,\; \langle [\wedge\; \neg P\; \neg Q] : \mathbf{0.7}\rangle\}$, which form the new base Σ_0. Now, **FI-UR** newly launches the process to compute the inconsistency of the new Σ_0 and with $\mathbf{Inc} = \mathbf{0.5}$ and follows the next steps:

- Using $\langle\, Q \;\; : \mathbf{0.6}\rangle$ and right-most formula in Σ_0 yields: $\langle\, (\vee) : \mathbf{0.6}\rangle$.
- The new base is $\Sigma_0 = \{\langle P : \mathbf{0.8}\rangle, \langle\, [\wedge\ \neg P\ \neg Q\,] : \mathbf{0.7}\rangle\}$ and the new **Inc** is **0.6**.
- $\mathcal{UR}_\Sigma$ is relaunched and finds $\langle\, (\vee) : \mathbf{0.7}\rangle$.
- The new Σ_0 is $\{\,\langle P : \mathbf{0.8}\rangle\,\}$ and the new **Inc** is **0.7**.
- $\mathcal{UR}_\Sigma$ finds Σ_0 is consistent and hence **FI-UR** returns **Inc = 0.7.**

7. Formal proofs

7.1. Proofs of Section 4

Lemma 4.7. *Let $\alpha, \alpha_i, \alpha_j \in (0, 1]$. Possibilistic Horn-NC formulas hold the next property:*

$$\langle\, (\vee\ \varphi_1 \ldots \varphi_i \ldots \varphi_k) : \alpha\,\rangle \in \overline{\mathcal{H}}_\Sigma \ \text{ iff } \ \exists i \text{ s.t. } \langle \varphi_i : \alpha_i\,\rangle \in \overline{\mathcal{H}}_\Sigma \ \text{ and } \ \forall j \neq i, \langle \varphi_j : \alpha_j\,\rangle \in \mathcal{N}^-_\Sigma .$$

Proof. If: Formulas $\forall j \neq i, \varphi_j$ have no positive literals, thus the non-negative disjunctions of $\varphi = (\vee\ \varphi_1 \ldots \varphi_i \ldots \varphi_k)$ are only those of φ_i. Since by hypothesis $\langle \varphi_i : \alpha_i\rangle \in \overline{\mathcal{H}}_\Sigma$ then φ obviously verifies Definition 4.3 and hence, $\langle\, \varphi : \alpha\,\rangle \in \overline{\mathcal{H}}_\Sigma$.

Only-If: It is easy to prove by contradiction that $\langle\, \varphi : \alpha\,\rangle \notin \overline{\mathcal{H}}_\Sigma$ if any of the two conditions of the lemma are unsatisfied, i.e. (i) $\exists i, \varphi_i \notin \overline{\mathcal{H}}_\Sigma$ or (ii) $\exists i, j, i \neq j, \langle \varphi_i : \alpha_i\,\rangle, \langle \varphi_j : \alpha_j\,\rangle \notin \mathcal{N}^-_\Sigma$. ■

Theorem 4.9. *We have that $\widehat{\mathcal{H}}_\Sigma = \overline{\mathcal{H}}_\Sigma$.*

Proof. We prove first $\widehat{\mathcal{H}}_\Sigma \subseteq \overline{\mathcal{H}}_\Sigma$ and then $\widehat{\mathcal{H}}_\Sigma \supseteq \overline{\mathcal{H}}_\Sigma$.

- $\widehat{\mathcal{H}}_\Sigma \subseteq \overline{\mathcal{H}}_\Sigma$ is easily proven by structural induction as outlined below:

(1) $\mathcal{L} \subset \overline{\mathcal{H}}_\Sigma$ trivially holds.

(2) The non-nested $\widehat{\mathcal{H}}_\Sigma$ conjunctions are literal conjunctions, which trivially verify Definition 4.3 and so are in $\overline{\mathcal{H}}_\Sigma$. Assume that $\widehat{\mathcal{H}}_\Sigma \subseteq \overline{\mathcal{H}}_\Sigma$ holds until a given inductive step and that $\langle \varphi_i : \alpha_i\,\rangle \in \widehat{\mathcal{H}}_\Sigma$, $\langle \varphi_i : \alpha_i\,\rangle \in \overline{\mathcal{H}}_\Sigma$, $1 \le i \le k$. In the next recursion, any $\langle [\wedge\ \varphi_1 \ldots \varphi_i \ldots \varphi_k] : \alpha\rangle$ may be added to $\widehat{\mathcal{H}}_\Sigma$. On the other hand, by Lemma 4.6, $\langle [\wedge\ \varphi_1 \ldots \varphi_i \ldots \varphi_k] : \alpha\rangle \in \overline{\mathcal{H}}_\Sigma$. Therefore $\widehat{\mathcal{H}}_\Sigma \subseteq \overline{\mathcal{H}}_\Sigma$ holds.

(3) The non-nested disjunctions in $\widehat{\mathcal{H}}_\Sigma$ are exclusively formed by literals, and so they are clearly Horn clauses. The latter are in $\overline{\mathcal{H}}_\Sigma$ because they have at least one non-negative disjunct (literal) and so fulfill Definition 4.3. Then assuming that for a given recursive level $\widehat{\mathcal{H}}_\Sigma \subseteq \overline{\mathcal{H}}_\Sigma$ holds, in the next recursion, only disjunctions in (3) are added to $\widehat{\mathcal{H}}_\Sigma$. But the condition of (3) and that of Lemma 4.7 coincide. Thus $\widehat{\mathcal{H}} \subseteq \overline{\mathcal{H}}_\Sigma$ holds.

- $\widehat{\mathcal{H}}_\Sigma \supseteq \overline{\mathcal{H}}_\Sigma$. Given that the structures to define $\mathcal{NC}$ and $\widehat{\mathcal{H}}_\Sigma$ in Definition 2.1 and Definition 4.8, respectively, are equal, the potential inclusion of each NC formula $\langle \varphi : \alpha\rangle$ in $\widehat{\mathcal{H}}_\Sigma$ is systematically considered. Further, the statement: if $\langle \varphi : \alpha\rangle \in \overline{\mathcal{H}}_\Sigma$ then $\langle \varphi : \alpha\rangle \in \widehat{\mathcal{H}}_\Sigma$, is proven by structural induction on the depth of formulas, by applying a reasoning similar to that of the $\widehat{\mathcal{H}}_\Sigma \subseteq \overline{\mathcal{H}}_\Sigma$ case and by also using Lemmas 4.6 and 4.7. ■

Definition 7.1. For every $\varphi \in \mathcal{NC}$, we define $cl(\varphi)$ as the unique clausal formula that results from applying $\vee/\wedge$ distributivity to φ until a clausal formula, viz. $cl(\varphi)$, is obtained.

Theorem 7.2. *Let $\langle \varphi = (\vee\ \varphi_1 \ldots \varphi_i \ldots \varphi_k\,) : \alpha\rangle \in \overline{\mathcal{H}}_\Sigma$, $\mathcal{H}$ be the class of propositional Horn clausal formulas and $\mathcal{N}^-$ be the class of propositional negative NC formulas. Then:*

$$cl(\,(\vee\ \varphi_1 \ldots \varphi_i \ldots \varphi_k\,)\,) \in \mathcal{H} \quad \text{iff} \quad (1)\ \exists i,\ s.t.\ cl(\varphi_i) \in \mathcal{H} \text{ and } (2)\ \forall j \neq i, \varphi_j \in \mathcal{N}^- .$$

Proof. *If-then.* By refutation: let $cl(\,(\vee\ \varphi_1 \ldots \varphi_i \ldots \varphi_k)\,) \in \mathcal{H}$ and prove that if (1) or (2) are violated, then $cl(\varphi) \notin \mathcal{H}$.

- (1) $\nexists i$, s.t. $cl(\varphi_i) \in \mathcal{H}$
  - If we take the case $k = 1$, then $\varphi = \varphi_1$.
  - But $cl(\varphi_1) \notin \mathcal{H}$ implies $cl(\varphi) \notin \mathcal{H}$.
- (2) $\exists j \neq i, \varphi_j \notin \mathcal{N}^-$.
  - So by assumption $\varphi_i, \varphi_j \notin \mathcal{N}^-$ with $j \neq i$.
  - We take $k = 2$, $\varphi_1 = P$ and $\varphi_2 = Q$.

– So, $\varphi = (\vee\ \varphi_1\ \varphi_2) = (\vee\ P\ Q)$, and hence $cl(\varphi) \notin \mathcal{H}$.

Only-If. Without loss of generality, we consider that $(\vee\ \varphi_1 \ldots \varphi_i \ldots \varphi_{k-1}) = \varphi^- \in \mathcal{N}^-$ and $\langle \varphi_k : \alpha_k \rangle \in \overline{\mathcal{H}}_\Sigma$, and prove by structural induction on the formulas:

$$cl(\varphi) = cl(\,(\vee\ \ \varphi_1 \ldots \varphi_i \ldots \varphi_{k-1}\,\varphi_k)\,) = cl(\,(\vee\ \varphi^-\ \ \varphi_k)\,) \in \mathcal{H}.$$

Clearly the statement is verified for non-nested formulas, i.e. Horn clauses. Assume that it is verified for φ_k, namely if $\langle \varphi_k : \alpha_k \rangle \in \overline{\mathcal{H}}_\Sigma$ then $cl(\varphi_k) \in \mathcal{H}$.

– To obtain $cl(\varphi)$, one must obtain first $cl(\varphi^-)$ and $cl(\varphi_k)$, and so

$$(i)\ \ cl(\varphi) = cl(\,(\vee\ \ \varphi^-\ \varphi_k)\,) = cl(\,(\vee\ \ cl(\varphi^-)\ \ cl(\varphi_k)\,)\,).$$

– By definition of $\varphi^- \in \mathcal{N}^-$,

$$(ii)\ \ cl(\varphi^-) = [\wedge\ D_1^- \ldots D_{m-1}^-\, D_m^-];\ \text{ the } D_i^-\text{'s being negative clauses.}$$

– By induction hypothesis,

$$(iii)\ \ cl(\varphi_k) = \mathrm{H} = [\wedge\, h_1\ \ \ldots h_{n-1}\, h_n];\ \text{ the } h_i\text{'s being Horn clauses.}$$

– By (i), (ii) and (iii),

$$cl(\varphi) = cl(\,(\vee\ \ [\wedge\ D_1^-\ \ \ldots\ D_{m-1}^-\, D_m^-]\ \ [\wedge\, h_1\ \ \ldots h_{n-1}\, h_n\,]\,)\,).$$

– Applying $\vee/\wedge$ distributivity to $cl(\varphi)$ and noting $C_i = (\vee\ D_1^-\ \ h_i\,)$,

$$cl(\varphi) = cl(\,[\wedge\ \ [\wedge\, C_1 \ldots C_i \ldots C_n]\ \ (\vee\ \ [\wedge\ D_2^- \ldots D_{m-1}^-\, D_m^-\,]\ \ \mathrm{H}\,)\,]\,).$$

– The $C_i = (\vee\ D_1^-\ \ h_i\,)$'s are Horn clauses, and so:

$$[\wedge\, C_1 \ldots C_i \ldots C_n] = \mathrm{H}_1 \in \mathcal{H}.$$

$$cl(\varphi) = cl(\,[\wedge\ \ \mathrm{H}_1\ \ (\vee\ \ [\wedge\ D_2^- \ldots D_{m-1}^-\, D_m^-\,]\ \ \mathrm{H}\,)\,]\,).$$

– For $j < m$ we have,

$$cl(\varphi) = cl(\,[\wedge\, \mathrm{H}_1\ \ \ldots \mathrm{H}_{j-1}\mathrm{H}_j\ \ (\vee\ \ [\wedge\ D_{j+1}^- \ldots D_{m-1}^-\, D_m^-]\ \ \mathrm{H}\,)\,]\,).$$

– For $j = m$, $cl(\varphi) = [\wedge\ \mathrm{H}_1\ \ \ldots \mathrm{H}_{m-1}\,\mathrm{H}_m\ \ \mathrm{H}\,] = \mathrm{H}' \in \mathcal{H}$.

– Hence $cl(\varphi) \in \mathcal{H}$. ■

Theorem 4.13. *Denoting by* $\mathrm{cl}(\varphi)$ *the clausal form of* φ*, we have:*

$$\langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma \quad \textit{entails} \quad \langle \mathrm{cl}(\varphi) : \alpha \rangle \in \mathcal{H}_\Sigma.$$

Proof. We consider Definition 4.8 of $\overline{\mathcal{H}}_\Sigma$. The proof is done by structural induction on the depth $r(\varphi)$ of any $\langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ and defined below, where ℓ is a literal:

$$r(\varphi) = \begin{cases} 0 & \varphi = \ell. \\ 1 + max\,\{r(\varphi_1), \ldots, r(\varphi_{k-1}),\, r(\varphi_k)\} & \varphi = [\odot\, \varphi_1\ \ \ldots \varphi_{k-1}\, \varphi_k]. \end{cases}$$

- *Base Case:* $r(\varphi) = 0$.
  - Clearly, $r(\varphi) = 0$ entails $\varphi = \ell \in \mathcal{H}$.
  - So $cl(\varphi) = \varphi \in \mathcal{H}$.
- *Induction hypothesis:* $\forall \varphi,\ r(\varphi) \le n,\ \langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ entails $cl(\varphi) \in \mathcal{H}$.
- *Induction proof:* $r(\varphi) = n + 1$.

  By Definition 4.8, lines (2) ad (3) below arise:

  (2) $\varphi = [\wedge\ \varphi_1 \ldots \varphi_i \ldots \varphi_k]$, where $k \ge 1$.

– By definition of $r(\varphi)$,

$$r(\varphi) = n + 1 \text{ entails } 1 \le i \le k,\ r(\varphi_i) \le n.$$

– By induction hypothesis,

$$\langle \varphi_i : \alpha_i \rangle \in \overline{\mathcal{H}}_\Sigma \text{ and } r(\varphi_i) \le n \text{ entail } cl(\varphi_i) \in \mathcal{H}.$$

– It is obvious that,

$$cl(\varphi) = [\wedge\ cl(\varphi_1)\ \dots\ cl(\varphi_i) \dots cl(\varphi_k)\,].$$

– Therefore,

$$cl(\varphi) = [\wedge\ \mathrm{H}_1 \dots \mathrm{H}_i \dots \mathrm{H}_k] = \mathrm{H} \in \mathcal{H}.$$

(3) $\varphi = (\vee\ \varphi_1 \dots \varphi_i \dots \varphi_{k-1}\ \varphi_k)$, where by Definition 4.8:

$$k \ge 1,\ 0 \le i \le k-1,\ \varphi_i \in \mathcal{N}^- \text{ and } \langle \varphi_k : \alpha_k \rangle \in \overline{\mathcal{H}}_\Sigma.$$

– By definition of $r(\varphi)$,

$$r(\varphi) = n + 1 \text{ entails } r(\varphi_k) \le n.$$

– By induction hypothesis,

$$r(\varphi_k) \le n \text{ and } \langle \varphi_k : \alpha_k \rangle \in \overline{\mathcal{H}}_\Sigma \text{ entail } cl(\varphi_k) \in \mathcal{H}.$$

– By Theorem 7.2, *only-if*,

$$0 \le i \le k-1,\ \varphi_i \in \mathcal{N}^- \text{ and } cl(\varphi_k) \in \mathcal{H} \text{ entail:}$$

$$cl(\,(\vee\ \varphi_1 \dots \varphi_i \dots \varphi_{k-1}\ \varphi_k)\,) \in \mathcal{H}. \quad \blacksquare$$

Theorem 4.16. $\overline{\mathcal{H}}_\Sigma$ *contains the NC fragment:*

$$\langle \varphi : \alpha \rangle \in \mathcal{NC}_\Sigma \quad \textit{and} \quad \langle \mathrm{cl}(\varphi) : \alpha \rangle \in \mathcal{H}_\Sigma \quad \textit{then} \quad \langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma.$$

where $\mathcal{NC}_\Sigma$ *is the set of possibilistic NC formulas and* $\mathrm{cl}(\varphi)$ *is the clausal form of* φ.

Proof. It is done by structural induction on the depth $d(\varphi)$ of φ defined as

$$d(\varphi) = \begin{cases} 0 & \ell \in \mathcal{L}. \\ 1 + max\,\{d(\varphi_1), \dots, d(\varphi_i), \dots, d(\varphi_k)\} & \varphi = [\odot\ \varphi_1 \dots \varphi_i \dots \varphi_k]. \end{cases}$$

- *Base case:* $d(\varphi) = 0$.
  - $d(\varphi) = 0$ entails $\ell \in \mathcal{L}$.
  - By Definition 4.8, $\langle \ell : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$.
- *Inductive hypothesis:* $\forall \varphi \in \mathcal{NC},\ d(\varphi) \le n,\ cl(\varphi) \in \mathcal{H}$ entail $\langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$.
- *Induction proof:* $d(\varphi) = n + 1$.

  By Definition 2.1 of $\mathcal{NC}$, cases (i) and (ii) below arise.

  (i) $\varphi = cl(\,[\wedge\ \varphi_1 \dots \varphi_i \dots \varphi_k]\,) \in \mathcal{H}$ and $k \ge 1$.

  – Hence, $1 \le i \le k$, $cl(\varphi_i) \in \mathcal{H}$.

  – By definition of $d(\varphi)$,

  $$d(\varphi) = n + 1 \text{ entails } 1 \le i \le k,\ d(\varphi_i) \le n.$$

  – By induction hypothesis,

  $$1 \le i \le k,\ d(\varphi_i) \le n,\ cl(\varphi_i) \in \mathcal{H} \text{ entail } \langle \varphi_i : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma.$$

  – By Definition 4.8, line (2),

  $$1 \le i \le k,\ \langle \varphi_i : \alpha_i \rangle \in \overline{\mathcal{H}}_\Sigma \text{ entails } \langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma.$$

(ii) $cl(\varphi) = cl((\vee\ \varphi_1 \dots \varphi_i \dots \varphi_{k-1}\ \varphi_k)) \in \mathcal{H}$ and $k \geq 1$.

– By Theorem 7.2, *if-then,*

$$0 \leq i \leq k-1,\ \varphi_i \in \mathcal{N}^- \text{ and } cl(\varphi_k) \in \mathcal{H}.$$

– By definition of $d(\varphi)$,

$$d(\varphi) = n+1 \text{ entails } d(\varphi_k) \leq n.$$

– By induction hypothesis,

$$d(\varphi_k) \leq n \text{ and } cl(\varphi_k) \in \mathcal{H} \text{ entail } \langle \varphi_k : \alpha_k \rangle \in \overline{\mathcal{H}}_\Sigma.$$

– By Definition 4.8, line (3),

$$0 \leq i \leq k-1,\ \langle \varphi_i : \alpha_k \rangle \in \mathcal{N}_\Sigma^- \text{ and } \langle \varphi_k : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma \text{ entail:}$$

$$\langle (\vee\ \varphi_1 \dots \varphi_i \dots \varphi_{k-1}\ \varphi_k) : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma. \quad \blacksquare$$

7.2. Proofs of Section 5

Proposition 5.4. *Rule (4) is sound:*

$$\langle \ell : \alpha \rangle, \langle (\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D}) : \beta \rangle \models \langle \mathcal{D} : \mathbf{min}\{\alpha, \beta\} \rangle.$$

Proof. Denoting $\langle \ell : \alpha \rangle, \langle (\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D}) : \beta \rangle$ by $\mathcal{F}$, we have:

- By *MinD*, $\mathcal{F} \models \langle (\vee\ \ \ell \wedge \mathcal{C}(\neg\ell)\ \ \ell \wedge \mathcal{D} : \mathbf{min}\{\alpha, \beta\} \rangle$.
- Since $\ell \wedge \mathcal{C}(\neg\ell) \equiv \bot$ then: $\mathcal{F} \models \langle \ell \wedge \mathcal{D} : \mathbf{min}\{\alpha, \beta\} \rangle$.
- By *MinD*, $\mathcal{F} \models \langle \mathcal{D} : \mathbf{min}\{\alpha, \beta\} \rangle$. $\blacksquare$

Proposition 5.6. *The rule UR_Σ is sound:*

$$\langle \ell : \alpha \rangle, \langle \Pi \succ (\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D}) : \beta \rangle \models \langle \mathcal{D} : \mathbf{min}\{\alpha, \beta\} \rangle.$$

Proof. Denoting $\langle \ell : \alpha \rangle, \langle \Pi \succ (\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D}) : \beta \rangle$ by $\mathcal{F}$, we have:

– By *MinD*, $\mathcal{F} \models \langle \ell \wedge \Pi \succ (\vee\ \mathcal{C}(\neg\ell)\ \mathcal{D}) : \mathbf{min}\{\alpha, \beta\} \rangle$.

– Then, $\mathcal{F} \models \langle \ell \wedge \Pi \succ (\vee\ \ell \wedge \mathcal{C}(\neg\ell)\ \ \ell \wedge \mathcal{D}) : \mathbf{min}\{\alpha, \beta\} \rangle$.

– Since $\ell \wedge \mathcal{C}(\neg\ell) \models \bot$ then, $\mathcal{F} \models \langle \ell \wedge \Pi \succ \ell \wedge \mathcal{D} : \mathbf{min}\{\alpha, \beta\} \rangle$.

– Then, $\mathcal{F} \models \langle \ell \wedge \Pi \succ \mathcal{D} : \mathbf{min}\{\alpha, \beta\} \rangle$.

– By *MinD*, $\mathcal{F} \models \langle \Pi \succ \mathcal{D} : \mathbf{min}\{\alpha, \beta\} \rangle$. $\blacksquare$

Lemma 5.9. *A propositional Horn-NC formula φ is inconsistent iff $\mathcal{UR}_P = \{UR_P, \bot\vee, \bot\wedge, \odot\phi, \odot\odot\} \subset \mathcal{UR}_\Sigma$ applied to $\langle \varphi : \alpha \rangle$ derives $\langle \bot : \alpha \rangle$.*

Proof. We analyze below both directions of the lemma.

• $\Rightarrow$ Let us assume that φ is inconsistent. Then φ must have a sub-formula verifying the UR_P numerator; otherwise, all complementary pairs of literals ℓ and $\neg\ell$ are included in disjunctions. In this case, since all disjunctions of φ, by definition of Horn-NC formula, have at least one negative literal, φ would be satisfied by assigning to all propositions the value 0, which contradicts the initial hypothesis. Therefore, UR_P is applied to φ with two complementary literals ℓ and $\neg\ell$ and the resulting formula is simplified. The new formula is equivalent to φ and has at least one literal less than φ. Hence, by induction on the number of literals of φ, we obtain that $\mathcal{UR}_P$ ends only when $\langle (\vee) : \alpha \rangle$ is derived.

• $\Leftarrow$ Let us assume that $\mathcal{UR}_P$ has been iteratively applied until a formula φ' different from $\langle (\vee) : \alpha \rangle$ is obtained. Clearly, if the UR_P numerator is not applicable then there is not a conjunction of a literal ℓ with a disjunction including $\neg\ell$. Then we have, firstly, since $\mathcal{UR}_P$ is sound, that φ and φ' are equivalent. Secondly, if φ' has complementary literals, then they are integrated in disjunctions. Thus φ' is satisfied by assigning the value 0 to all its unassigned propositions, since, by definition of Horn-NC formula, all disjunctions have at least one negative disjunct. Therefore, since φ' is consistent so is φ, a contradiction. $\blacksquare$

Proposition 5.10. *Testing the propositional consistency of φ such that $\langle \varphi : \alpha \rangle \in \overline{\mathcal{H}}_\Sigma$ with the inferences $\{UR_P, \bot\vee, \bot\wedge, \odot\phi, \odot\odot\} \subset \mathcal{UR}_\Sigma$ is polynomial.*

Proof. Each application of rule UR_P deletes at least one literal, and so the maxim number required of such inference is bounded by the formula size. Each application of the rules $\{\bot\vee, \bot\wedge, \odot\phi, \odot\odot\}$ removes at least a connective, and hence, the maxim number required of such inferences is limited by the formula size. On the other hand, it is not difficult to find data structures to execute polynomially each rule. Hence, the lemma holds. ■

Lemma 5.12. *Let $\Sigma \in \overline{\mathcal{H}}_\Sigma$. $\mathcal{UR}_\Sigma$, MinD and MaxN derive $\langle (\vee) : \alpha \rangle$ iff Σ is inconsistent, and if $\langle (\vee) : \alpha \rangle$ is derived then Inc(Σ) $\geq \alpha$.*

Proof. The propositional component of possibilistic NC unit-resolution verifies Lemma 5.9, and hence, $\mathcal{UR}_\Sigma$ derives an empty formula $\langle (\vee) : \alpha \rangle$ iff the conjunction of the propositional formulas in the base Σ is inconsistent, namely if Σ^* is inconsistent. If $\mathcal{UR}_\Sigma$ derives $\langle (\vee) : \alpha \rangle$, then by Lemma 5.9, $\mathcal{UR}_\Sigma$ detects a subset $\Sigma_1 \subseteq \Sigma$ which is indeed inconsistent. Then by Proposition 5.6, the degree α found by $\mathcal{UR}_\Sigma$ corresponds to $\text{Inc}(\Sigma_1)$. Since obviously $\text{Inc}(\Sigma_1) \leq \text{Inc}(\Sigma)$, the lemma holds. ■

The remainder of this section is devoted to prove the correctness and polynomial complexity of the algorithm **FI-UR**, i.e. of Lemma 5.19.

Lemma 5.19. *If **FI-UR(**$\Sigma, 0$**)** returns α then* $\text{Inc}(\Sigma) = \alpha$.

Proof. We use Σ' and **Inc** to denote the variables of **Find** in a given recursion. Let us prove that the next hypothesis holds in every call to **FI-UR**:

$$\Sigma_{>\mathbf{Inc}} = \Sigma' \;\text{ and }\; \text{Inc}(\Sigma) = \max\{\text{Inc}(\Sigma'), \mathbf{Inc}\}$$

– We check that the initial call **FI-UR(**$\Sigma, 0$**)** verifies the hypothesis:
  - We have $\Sigma' = \Sigma$ and **Inc** = 0.
  - Thus $\Sigma_{>0} = \Sigma'$ and $\max\{\text{Inc}(\Sigma'), 0\} = \text{Inc}(\Sigma)$
– We prove that if the hypothesis holds for $k \geq 1$ then it holds for $k+1$.
– First of all, $\mathcal{UR}_\Sigma$ is applied to Σ'.
– By Lemma 5.12, if $\mathcal{UR}_\Sigma$ derives $\langle \bot : \alpha \rangle$ then $\text{Inc}(\Sigma') \geq \alpha$, else Σ' is consistent.
– *Case Σ' is consistent:* $\text{Inc}(\Sigma') = 0$.
  - By induction hypothesis $\text{Inc}(\Sigma) = \max\{\text{Inc}(\Sigma'), \mathbf{Inc}\} = \mathbf{Inc}$
  - So **FI-UR** correctly returns $\text{Inc}(\Sigma)$ and ends.
– *Case Σ' is inconsistent:* $\text{Inc}(\Sigma') = \alpha > 0$.
  - By induction hypothesis $\Sigma' = \Sigma_{>\mathbf{Inc}}$, and so $\alpha > \mathbf{Inc}$.
  - The next Σ', denoted Σ'', and **FI-UR**, denoted **Inc'**, are $\Sigma'' = \Sigma'_{>\alpha}$ and **Inc'**=α.
    We check in (i) and (ii) below that the hypothesis holds:
  (i) By induction hypothesis: $\Sigma' = \Sigma_{>\mathbf{Inc}}$
    – Since $\alpha > \mathbf{Inc}$ then trivially $\Sigma'_{>\alpha} = \Sigma_{>\alpha}$
    – Hence $\Sigma'' = \Sigma'_{>\alpha} = \Sigma_{>\alpha} = \Sigma_{>\mathbf{Inc'}}$.
  (ii) By Lemma 5.12, $\text{Inc}(\Sigma') \geq \alpha$
    – Since $\Sigma' = \Sigma_{>\mathbf{Inc}}$ and $\alpha > \mathbf{Inc}$ then $\text{Inc}(\Sigma) \geq \alpha$
    – Hence $\text{Inc}(\Sigma) = \max\{\text{Inc}(\Sigma_{>\alpha}), \alpha\} = \max\{\text{Inc}(\Sigma''), \mathbf{Inc'}\}$

Altogether, the hypothesis holds until **FI-UR** finds Σ' consistent and then correctly returns $\text{Inc}(\Sigma)$. Hence the lemma's statement holds. ■

Proposition 7.3. *If $\Sigma \in \overline{\mathcal{H}}_\Sigma$, then $\mathcal{UR}_\Sigma$ with input Σ performs at most $m \times k$ inferences, m and k being the number of connective occurrences and different weights in Σ, respect.*

Proof. On the one hand, each rule of $\mathcal{UR}_\Sigma$ adds a formula $\langle \varphi : \alpha \rangle$, where φ is a sub-formula of a propositional formula Π of $\langle \Pi : \beta \rangle \in \Sigma$. Hence, the current base always contains only sub-formulas from Σ. On the other hand, the weight α of added formulas $\langle \varphi : \alpha \rangle$ is the minimum of two weights in the current base. Hence, the weights of formulas in the current base always come from Σ. Thus, the maximum number of deduced formulas is $m \times k$. ■

Proposition 7.4. *If* $\Sigma \in \overline{\mathcal{H}}_\Sigma$*, then* ***FI-UR***$(\Sigma, 0)$ *performs at most* $n \times k$ *recursive calls to* $\mathcal{UR}_\Sigma$*, n and k being the number of propositions and different weights in* Σ*, respect.*

Proof. FI-UR stops when it detects that the current base is consistent. If it is inconsistent then $\langle (\vee) : \alpha \rangle$ is deduced, and **FI-UR** cancels the set $\{\langle \phi : \beta \rangle \mid \langle \phi : \beta \rangle \in \Sigma, \beta \leq \alpha\}$. This set trivially contains at least two unit clauses $\langle P : \beta \rangle$ and $\langle \neg P : \beta' \rangle$ with $\alpha = \min\{\beta, \beta'\}$ as $\langle (\vee) : \alpha \rangle$, was derived. On the other hand, each recursive call to **FI-UR**, the weight **Inc** increases and since $\mathcal{UR}_\Sigma$ requires unit-clauses, then Proposition 7.4 holds. ■

Lemma 5.20. *If* $\Sigma \in \overline{\mathcal{H}}_\Sigma$*, then computing Inc*$(\Sigma)$ *takes polynomial time.*

Proof. It follows from Propositions 7.3 and 7.4 and the fact that it is not hard to find a data structure so that each inference in $\mathcal{UR}_\Sigma$ can be polynomially performed. ■

Remark. A tight determination of the polynomial degree of the worst-case complexity of computing a Horn-NC base is planned for future work (see Section 8).

8. Related and future work

□ Polynomial NC Classes: Searching for propositional Horn (clausal) super-classes such as hidden-Horn, generalized Horn, Q-Horn, extended-Horn, etc. (see [44,52] for short reviews) has been a key issue for several decades. So it is arguable that, just as the tractable clausal fragment has helped to grow overall clausal efficiency, likewise widening the tractable NC fragment would grow overall NC efficiency. We will extrapolate such an argument to possibilistic logic and determine further polynomial Horn-NC super-classes.

□ Polynomial Algorithms. As we have seen in Section 7.2, the complexity of determining $\mathrm{Inc}(\Sigma)$ of Horn-NC bases is in (around) $O(n^4)$. This complexity, though polynomial, is not satisfactory for applications. However, not much care has been taken in the proofs of Section 7.2 because the goal was proving tractability. Thus, on the one hand, a fine-grained analysis of complexity is pending, and on the other hand, our research should be resumed towards notably decreasing the degree of the polynomial complexity.

□ Necessity and Possibility. Some ideas presented here can be extended to bases containing possibility and necessity measures [50,57,22]. Since that implies a need to enlarge the language, probably tractability would be lost. $\langle \varphi : \Pi(\varphi) \geq \alpha \rangle$ is equivalent to $\langle \neg\varphi : N(\neg\varphi) < 1 - \alpha \rangle$, but then the bases incorporating the latter formulas contain both: (1) constraints $N(\alpha) \geq \alpha$ and $N(\alpha) < \alpha$; and (2) Horn-NC formulas φ and formulas $\neg\varphi$ that are dual Horn-NC. The union of Horn-NC and of dual Horn-NC formulas is still a subset of NC, and thus, it is less hard than NC, yet solving together both formula classes is unlikely to be polynomial.

□ Logic Programming (LP). After the pioneer work in possibilistic LP [28], a number of further approaches e.g. [12,3,4,2] were published. However, all of them focus on the clausal form, whereas we will handle non-clausal logic programs (see Example 1.2 and Appendix). Formal arguments showing that the Horn-NC programs are the legitimate generalization of classical Horn programs are in [53]. See Appendix for the extension of Horn-NC programs from propositional to possibilistic logic.

□ Answer Set Programming (ASP). A succession of works on ASP in possibilistic logic has been carried out, started by [67] and continued with e.g. [70,71,20,21,8,9]. Although, most of them focus on the clausal form, possibilistic NC (nested) ASP has been formerly dealt with in [68,69]. The authors extend from classical to possibilistic logic, concepts as originally defined in [61], and so their aim is distinct from ours. Our first goal will be to study the scope of the class $\overline{\mathcal{H}}_\Sigma$ in computing reducts of possibilistic answer nested programs and to analyze the efficiency allowed by $\mathcal{UR}_\Sigma$. Answering queries in possibilistic ASP is an intractable problem [67]. The applicability of the presented proposal to both ASP with aggregates and ASP with preferences is discussed in Appendix.

□ NC Resolution. Computing arbitrary NC bases is a natural continuation of the presented work, and the latter also eases deduction based on resolution and DPLL. The formalization of the existing NC resolution [64] (see also [5]), which dates back to the 1980s, has some weaknesses due to its functional-like definition, such as not precisely identifying the available resolvents or requiring complex formal proofs [49]. On the other hand, resolution for possibility-necessity formulas is well-known [29,32,33], but its extension to general NC bases is an open question. We believe that our definition is fairly well oriented to define (full) NC Resolution and to generalize it to some uncertainty logics.

□ NC DPLL. Possibilistic DPLL was already studied [29,57] but possibilistic NC DPLL has received no attention. Our proposal is a step forward to specify it because NC DPLL relies on: (1) a suitable heuristic to choice the literal ℓ on which performing branching; and (2) applying unit-propagation to $\Sigma \wedge \ell$ and $\Sigma \wedge \neg\ell$. Since NC unit-propagation relies on NC unit-resolution, that is, on $\mathcal{UR}_\Sigma$, only addressing heuristic issues is pending.

□ Generalized Possibilistic Logic (GPL). In standard possibilistic logic only conjunctions are weighted, while in GPL [35, 36,34] disjunctions and negations are weighted too. For instance, GPL includes $(\vee\ \langle\varphi:\alpha\rangle\ \langle\psi:\beta\rangle)$ and also the next formula:

$$[\wedge\ \ \langle(\vee\ P\ \ Q):\mathrm{N}\ \mathbf{1}\rangle\ \langle\neg Q:\mathrm{N}\ \mathbf{.75}\rangle\ \neg\langle(\vee\ \neg P\ \ R):\mathrm{N}\ \mathbf{1}\rangle\quad(\vee\ \langle\neg R:\Pi\ \mathbf{.75}\rangle\ \langle\varphi:\Pi\ \mathbf{.25}\rangle\ )\ ]$$

GPL connectives can be internal or external (GPL is a two-tired logic) with their different semantics, and also, GPL formulas can be expressed in NC, e.g. the previous formula is NC. In [36] it is proven that satisfiability of GPL formulas is $\mathcal{NP}$-complete. Standard possibilistic Horn formulas are encapsulated in GPL and so can be Horn-NC formulas. We think that sub-classes of external Horn formulas can also be defined, as well as sub-classes that are both internally and externally Horn-like. Thus GPL embeds a variety of classes of Horn-like formulas which potentially could be lifted to NC.

□ Models/Inconsistent Subsets. In some frameworks, e.g., when the knowledge base is in an experimentation phase, the only data of $\mathrm{Inc}(\Sigma)$ may be of not much help. For example, if one expects the knowledge base to be consistent and the checker finds it is inconsistent, knowing the knowledge subset causing contradiction, called "witness", is required. Thus, we will envisage deductive calculi oriented to providing witnesses as a return data. A more complicated problem is the determination of whether a knowledge base has exactly one model, or one inconsistent subset (see [75], Chap. 17).

9. Conclusions

Deduction in classical logic and non-clausal form emerged within the pioneer fields of artificial intelligence. Nowadays, non-clausal deduction is present in many reasoning areas both in classical and non-classical logic, and is especially meaningful in answer set programming, which possesses a prominent problem-solving methodology.

In this paper, we have extrapolated the above non-clausal computing interest to possibilistic logic, the most extended approach to deal with knowledge impregnated with uncertainty and presenting partial inconsistencies.

Our first contribution has been lifting the possibilistic Horn class to the non-clausal level obtaining a new possibilistic class, which has been called Horn Non-Clausal, denoted by $\overline{\mathcal{H}}_\Sigma$ and shown to be non-clausal, analogous to the standard Horn class. Indeed, we have proven that $\overline{\mathcal{H}}_\Sigma$ subsumes syntactically the Horn class and that both classes are semantically equivalent. We have also proven that all possibilistic non-clausal bases whose clausal form is Horn belong to $\overline{\mathcal{H}}_\Sigma$.

Towards obtaining the inconsistency degree of Horn-NC bases, we have established the calculus "Possibilistic Non-Clausal Unit-Resolution", denoted $\mathcal{UR}_\Sigma$. We formally proved that $\mathcal{UR}_\Sigma$ correctly computes the inconsistency degree of any Horn-NC base. $\mathcal{UR}_\Sigma$ was nonexistent in the literature and extends the propositional logic calculus in [52].

After having specified $\overline{\mathcal{H}}_\Sigma$ and $\mathcal{UR}_\Sigma$, we have studied the computational problem of calculating the inconsistency degree of Horn-NC bases through two methods: (a) dichotomic search and calls to a propositional solver; and (b) interleaving calls to logical calculi $\mathcal{UR}_\Sigma$ with adequate α-cuts. Both methods have allowed us to determine that $\overline{\mathcal{H}}_\Sigma$ is polynomial. Altogether, $\overline{\mathcal{H}}_\Sigma$ is the first found class to be possibilistic, non-clausal and polynomial.

Our formulation of $\mathcal{UR}_\Sigma$ is unambiguously clausal-like since, when applied to clausal formulas, $\mathcal{UR}_\Sigma$ indeed coincides with clausal unit-resolution. This aspect is relevant in the sense that it lays the foundations towards redefining NC resolution in a clausal-like manner which could avoid the barriers caused by the existing functional-like definition (see related work). We believe that this clausal-like definition of NC resolution will allow it to be generalized to some other uncertainty logics.

Besides the application of the presented approach to polynomially determining the inconsistency degree of a meaningful sub-class of non-clausal possibilistic bases, we have informally discussed, via illustrative examples, that the Horn-NC formulas are the cornerstone of definite nested logic programing. To put it in another way, the Horn-NC programs are the rightful generalization of the Horn programs [53]. Furthermore and within this area, we have also outlined how Horn-NC programs can be helpful to handle aggregates and preferences in answer set nested logic programming.

Finally, we also attempted to show that effective NC reasoning for possibilistic logic is an open research field and, in view of our outcomes, rather promising. A symptom of such consideration is the potentiality of our method to be extended to different possibilistic logic contexts giving rise to a number of future research directions, which were briefly discussed and which include: computing possibilistic arbitrary NC bases; discovering additional tractable NC subclasses; extending generalist possibilistic logic to NC; combining necessity and possibility measures; considering partially ordered possibility measures; developing possibilistic NC answer set programming; defining possibilistic NC resolution; defining possibilistic NC DPLL; finding "witnesses", etc.

Some of the above listed future objectives can also be searched in the context of other non-classical logics such as Łukasiewicz logics, Gödel logic, product logic, etc.

10. Appendix

The content of this appendix is the following: (1) it gives specific examples of the equivalence between propositional definite programs (based on Horn clauses) and their equivalent definite non-clausal programs (based on Horn nested formulas) (see [53] for details and formal aspects); (2) it shows that the possibilistic Horn-NC formulas underpinning possibilistic definite non-clausal programs are more general than those employed in this article; and (3) it discusses briefly the connections of possibilistic Horn-NC bases with two sub-areas of ASP (which were not commented in related work).

□ Logic programming (LP). We surveyed some works in possibilistic LP in Related Work and now provide programs that illustrate the equivalence between traditional definite programs and definite non-clausal programs. A definite LP rule has a conjunction of propositions as body and a single literal as head, whereas a definite non-clausal rule has the syntax $H \leftarrow B$, where B is a propositional NC formula and H is Horn-NC. As expected, the examples indicate that Horn-NC programs, being more compact than Horn programs, need to evaluate substantially less literals and connectives. This allows for minimizing redundancies and so for important improvements in efficiency with respect to the interpreters, developed up to now, which run on their counterpart classical programs.

Example 10.1. The six definite rules below can be compacted into the definite non-clausal rule, where only one (negative) literal and just once is repeated. One can check that the head and the whole nested rule are both Horn-NC formulas:

$$b \leftarrow c, f, e \quad b \leftarrow c, d, e \quad a \leftarrow c, f \quad a \leftarrow c, d \quad h \leftarrow c, e \quad h \leftarrow c, b$$

$$[\wedge\ (\vee\ h\ [\wedge\ \neg e\ \neg b])\ \ (\vee\ [\wedge\ (\vee\ b\ \neg e)\ a]\ [\wedge\ \neg\ f\ \neg d])\ \longleftarrow c$$

Example 10.2. The nine classical definite rules below can be merged into the Horn-NC rule in the second line. It can be verified that the non-clausal head and the whole rule are Horn-NC formulas. Furthermore, this example serves to show that the saving of literals (and connectives) can be exponential because from 3^3 literals in LP, one requires only $3+3$ literals in the non-clausal rule:

$$c \leftarrow b \quad c \leftarrow d \quad c \leftarrow e \quad a \leftarrow b \quad a \leftarrow d \quad a \leftarrow e \quad f \leftarrow b \quad f \leftarrow d \quad f \leftarrow e$$

$$r = (\vee_2\ [\wedge\ \neg b\ \neg d\ \neg e]\ [\wedge\ c\ a\ f\,])\leftarrow \top \quad \square$$

However the situation becomes relatively more elaborated within possibilistic logic and to show it, we take a very simple base with only two rules $\alpha : h \leftarrow f, a$ and $\beta : h \leftarrow f, b$. In order to faithfully codify such rules in a possibilistic Horn-NC rule, one should resort to the rule:

$$(\vee\ h\ [\wedge\ \langle \neg a : \alpha\rangle\ \langle \neg b : \beta\rangle\,])\leftarrow f$$

This simple example exemplifies the fact that the head of possibilistic definite non-clausal rules contains formulas whose literal occurrences ℓ is associated with an individual weight, the one attached to the rule in which ℓ occurs. Hence, the head of a possibilistic definite non-clausal rule is a more general formula than those studied in this article, as we consider a set of formulas $\langle \varphi : \alpha\rangle$, where φ does not include inner weights. These more general possibilistic Horn-NC formulas will be coped with in future work.

□ Answer set programs (ASPs) with Aggregates. ASPs having aggregates as atoms, give rise to a powerful language that nowadays receives a vast attention, e.g. [41,16]. ASPs with aggregates are equivalent to nested ASPs [42,41]. Horn-NC rules have the form $H \leftarrow B$, where B is a positive NC and H is Horn-NC (the whole rule is also Horn-NC). So, Horn-NC programs cover the ASPs with aggregates whose reducts Π^X are Horn-NC programs. The example below is taken from [41]. In this case, the aggregate

$$r \leftarrow sum\langle\{p=1, q=1\}\rangle \neq 1 \qquad p \leftarrow r \qquad q \leftarrow r \qquad p \leftarrow q \qquad q \leftarrow p$$

and thus the whole rule are "almost" Horn-NC (the difficulty with aggregates is that their equivalent formula is sometimes hard to determine). Importantly, programs with aggregates can be decomposed into Horn-NC programs, just as classical disjunctive programs can be into Horn programs. For instance, if one had to cope with rule $H \vee \varphi \leftarrow sum\langle\{p=1, q=1\}\rangle \neq 1$, where H is Horn-NC, one could decompose it into: $\varphi \leftarrow sum\langle\{p=1, q=1\}\rangle \neq 1$ and $H \leftarrow sum\langle\{p=1, q=1\}\rangle \neq 1$. If φ was not Horn-NC, then the decomposition would be recursively applied to the former rule until only Horn-NC rules are derived. As with clausal rules, the number of derived Horn-NC rules is exponentially bounded.

□ ASP with Preferences. Preferences in ASP [13] are envisaged by allowing a new connective $\times$ in the rule heads: $a \times b \leftarrow B$ means "if B holds, option a is preferred to b". This approach was lifted to non-clausal ASP [19] whose disjunctive options can be nested or NC formulas φ. Horn-NC programs cannot represent any subclass of nested programs with preferences as the latter are intrinsically disjunctive, whereas the former are definite or deterministic. Yet, Horn-NC programs can be helpful as follows: nested programs with preferences can be decomposed into a set of Horn-NC programs (as aggregate programs above). For instance, the rule $\varphi \times \varphi' \leftarrow B$ can be split into $\varphi \leftarrow B$ and $\varphi' \leftarrow B$, and each of them can be recursively decomposed until only Horn-NC programs remain. Obviously, the Horn-NC programs inheriting $\varphi' \leftarrow B$ must inherit a penalty higher than that inherited by those programs inheriting $\varphi \leftarrow B$. After the decomposition, one can employ mechanisms based on NC unit-resolution to find the answer sets. An appropriate management of penalties by the Horn-NC decomposition process needs to be studied.

Declaration of competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

The author would like to express his gratitude to the anonymous reviewers for their help to significantly improve on previous versions of the paper. This work was supported by the Spanish project ISINC (PID2019-111544GB-C21) funded by MCIN/AEI/10.13039/501100011033.

References

[1] G. Aguilera, I. de Guzman, M. Ojeda, A reduction-based theorem prover for 3-valued logic, Mathw. Soft Comput. 4 (2) (1997) 99–127.
[2] T. Alsinet, C.I. Chesñevar, L. Godo, G.R. Simari, A logic programming framework for possibilistic argumentation: formalization and logical properties, Fuzzy Sets Syst. 159 (10) (2008) 1208–1228.
[3] T. Alsinet, L. Godo, A complete calculus for possibilistic logic programming with fuzzy propositional variables, in: UAI '00: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence, Stanford University, Stanford, California, USA, June 30 - July 3, 2000, 2000, pp. 1–10.
[4] T. Alsinet, L. Godo, S.A. Sandri, Two formalisms of extended possibilistic logic programming with context-dependent fuzzy unification: a comparative description, Electron. Notes Theor. Comput. Sci. 66 (5) (2002) 1–21.
[5] L. Bachmair, H. Ganzinger, Resolution theorem proving, in: Handbook of Automated Reasoning, 2001, pp. 19–99 (in 2 volumes).
[6] C. Baral, Knowledge Representation, Reasoning and Declarative Solving, Cambridge University Press, 2003.
[7] C.W. Barrett, J. Donham, Combining SAT methods with non-clausal decision heuristics, Electron. Notes Theor. Comput. Sci. 125 (3) (2005) 3–12.
[8] K. Bauters, S. Schockaert, M.D. Cock, D. Vermeir, Possible and necessary answer sets of possibilistic answer set programs, in: IEEE 24th International Conference on Tools with Artificial Intelligence, ICTAI 2012, Athens, Greece, November 7–9, 2012, 2012, pp. 836–843.
[9] K. Bauters, S. Schockaert, M.D. Cock, D. Vermeir, Semantics for possibilistic answer set programs: uncertain rules versus rules with uncertain conclusions, Int. J. Approx. Reason. 55 (2) (2014) 739–761.
[10] B. Beckert, R. Hähnle, G. Escalada-Imaz, Simplification of many-valued logic formulas using anti-links, J. Log. Comput. 8 (4) (1998) 569–587.
[11] M. Ben-Ari, Mathematical Logic for Computer Science, 3rd edition, Springer, 2012.
[12] S. Benferhat, D. Dubois, H. Prade, Possibilistic logic: from nonmonotonicity to logic programming, in: Symbolic and Quantitative Approaches to Reasoning and Uncertainty, European Conference, Proceedings, ECSQARU'93, Granada, Spain, November 8–10, 1993, 1993, pp. 17–24.
[13] G. Brewka, I. Niemelä, T. Syrjänen, Logic programs with ordered disjunction, Comput. Intell. 20 (2) (2004) 335–357.
[14] F. Bry, N. Eisinger, T. Eiter, T. Furche, G. Gottlob, C. Ley, B. Linse, R. Pichler, F. Wei, Foundations of rule-based query answering, in: Reasoning Web, Tutorial Lectures, Third International Summer School 2007, Dresden, Germany, September 3-7, 2007, 2007, pp. 1–153.
[15] U. Bubeck, H. Kleine Büning, Nested boolean functions as models for quantified boolean formulas, in: Theory and Applications of Satisfiability Testing - SAT 2013 - 16th International Conference, Proceedings, Helsinki, Finland, July 8–12, 2013, 2013, pp. 267–275.
[16] P. Cabalar, J. Fandinno, T. Schaub, S. Schellhorn, Gelfond-Zhang aggregates as propositional formulas, Artif. Intell. 274 (2019) 26–43.
[17] M. Cadoli, M. Schaerf, On the complexity of entailment in propositional multivalued logics, Ann. Math. Artif. Intell. 18 (1) (1996) 29–50.
[18] K. Claessen, N. Een, M. Sheeran, N. Sörenson, A. Voronov, K. Akesson, SAT-solving in practice, with a tutorial example from supervisory control, Discrete Event Dyn. Syst. 19 (2009) 495–524.
[19] R. Confalonieri, J.C. Nieves, Nested preferences in answer set programming, Fundam. Inform. 113 (1) (2011) 19–39.
[20] R. Confalonieri, J.C. Nieves, M. Osorio, J. Vázquez-Salceda, Dealing with explicit preferences and uncertainty in answer set programming, Ann. Math. Artif. Intell. 65 (2–3) (2012) 159–198.
[21] R. Confalonieri, H. Prade, Using possibilistic logic for modeling qualitative decision: answer set programming algorithms, Int. J. Approx. Reason. 55 (2) (2014) 711–738.
[22] O. Couchariere, M. Lesot, B. Bouchon-Meunier, Consistency checking for extended description logics, in: Proceedings of the 21st International Workshop on Description Logics (DL2008), Dresden, Germany, May 13–16, 2008, 2008, pp. 13–16.
[23] A. Darwiche, Decomposable negation normal form, J. ACM 48 (4) (2001) 608–647.
[24] P. Doherty, J. Kachniarz, A. Szalas, Meta-queries on deductive databases, Fundam. Inform. 40 (1) (1999) 7–30.
[25] W. Dowling, J. Gallier, Linear-time algorithms for testing the satisfiability of propositional Horn formulae, J. Log. Program. 3 (1984) 267–284.
[26] R. Drechsler, T. Junttila, I. Niemelä, Non-clausal SAT and ATPG, in: A. Biere, M. Heule, H. van Maaren, T. Walsh (Eds.), Handbook of Satisfiability: Chapter 21, IOS Press, 2009, pp. 655–693.
[27] V. D'Silva, D. Kroening, G. Weissenbacher, A survey of automated techniques for formal software verification, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 27 (7) (2008).
[28] D. Dubois, J. Lang, H. Prade, Towards possibilistic logic programming, in: Logic Programming, Proceedings of the Eigth International Conference, Paris, France, June 24–28, 1991, 1991, pp. 581–595.
[29] D. Dubois, J. Lang, H. Prade, Possibilistic logic, in: Handbook of Logic in Artificial Intelligence and Logic Programming, Oxford University Press, New York, 1994, pp. 419–513.
[30] D. Dubois, H. Prade, Necessity measures and the resolution principle, IEEE Trans. Syst. Man Cybern. 17 (3) (1987) 474–478.
[31] D. Dubois, H. Prade, Resolution principles in possibilistic logic, Int. J. Approx. Reason. 4 (1) (1990) 1–21.
[32] D. Dubois, H. Prade, Possibilistic logic: a retrospective and prospective view, Fuzzy Sets Syst. 144 (1) (2004) 3–23.
[33] D. Dubois, H. Prade, Possibilistic logic - an overview, in: J. Woods, D.M. Gabbay, J.H. Siekmann (Eds.), Handbook of the History of Logic. Vol 9, Computational Logic, North-Holland, 2014, pp. 283–342.
[34] D. Dubois, H. Prade, A crash course on generalized possibilistic logic, in: Scalable Uncertainty Management - 12th International Conference, Proceedings, SUM 2018, Milan, Italy, October 3–5, 2018, 2018, pp. 3–17.
[35] D. Dubois, H. Prade, S. Schockaert, Stable models in generalized possibilistic logic, in: Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10–14, 2012, 2012.

[36] D. Dubois, H. Prade, S. Schockaert, Generalized possibilistic logic: foundations and applications to qualitative reasoning about uncertainty, Artif. Intell. 252 (2017) 139–174.
[37] O. Dubois, P. André, Y. Boufkhad, Y. Carlie, Chap. SAT vs. UNSAT, in: Second DIMACS Implementation Challenge: Cliques, Coloring and Unsatisfiability, in: DIMACS Series in Discrete Mathematics and Theoretical Computer Sciences, vol. 26, American Mathematical Society, 1996, pp. 415–436.
[38] U. Egly, M. Seidl, S. Woltran, A solver for QBFs in nonprenex form, in: ECAI 2006, 17th European Conference on Artificial Intelligence, Including Prestigious Applications of Intelligent Systems (PAIS 2006), Proceedings, August 29 - September 1, 2006, Riva del Garda, Italy, 2006, pp. 477–481.
[39] U. Egly, M. Seidl, S. Woltran, A solver for QBFs in negation normal form, Constraints Int. J. 14 (1) (2009) 38–79.
[40] M. Färber, C. Kaliszyk, Certification of nonclausal connection tableaux proofs, in: Automated Reasoning with Analytic Tableaux and Related Methods - 28th International Conference, Proceedings, TABLEAUX 2019, London, UK, September 3–5, 2019, 2019, pp. 21–38.
[41] P. Ferraris, Logic programs with propositional connectives and aggregates, ACM Trans. Comput. Log. 12 (4) (2011) 25.
[42] P. Ferraris, V. Lifschitz, Weight constraints as nested expressions, Theory Pract. Log. Program. 5 (1–2) (2005) 45–74.
[43] G. Fiorino, A non-clausal tableau calculus for minsat, Inf. Process. Lett. 173 (2022) 106167.
[44] J. Franco, J. Martin, A history of satisfiability, in: A. Biere, M. Heule, H. van Maaren, T. Walsh (Eds.), Handbook of Satisfiability: Chapter 1, IOS Press, 2009, pp. 3–55.
[45] C.D. Giusto, M. Gabbrielli, M.C. Meo, On the expressive power of multiple heads in CHR, ACM Trans. Comput. Log. 13 (1) (2012) 6.
[46] G. Gogic, H.A. Kautz, C.H. Papadimitriou, B. Selman, The comparative linguistics of knowledge representation, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20–25, 1995, Morgan Kaufmann, 1995, pp. 862–869, 2 volumes.
[47] H. Habiballa, Resolution strategies for fuzzy description logic, in: New Dimensions in Fuzzy Logic and Related Technologies. Proceedings of the 5th EUSFLAT Conference, Vol. 2, Regular Sessions, Ostrava, Czech Republic, September 11–14, 2007, 2007, pp. 27–36.
[48] H. Habiballa, Fuzzy Logic: Algorithms, Techniques and Implementations, chapter Resolution Principle and Fuzzy Logic, InTec, 2012, pp. 55–74.
[49] R. Hähnle, N.V. Murray, E. Rosenthal, Linearity and regularity with negation normal form, Theor. Comput. Sci. 328 (3) (2004) 325–354.
[50] B. Hollunder, An alternative proof method for possibilistic logic and its application to terminological logics, Int. J. Approx. Reason. 12 (2) (1995) 85–109.
[51] A. Horn, On sentences which are of direct unions of algebras, J. Symb. Log. 16 (1) (1951) 14–21.
[52] G.E. Imaz, The Horn non-clausal class and its polynomiality, CoRR, http://arxiv.org/abs/2108.13744, 2021, 1–59.
[53] G.E. Imaz, Normal nested answer set programs: syntactics, semantics and logical calculi, November 2022, pp. 1–40, in preparation.
[54] G.E. Imaz, A first polynomial non-clausal class in many-valued logic, Fuzzy Sets Syst. (October 2022) 1–37, https://doi.org/10.1016/j.fss.2022.10.008, in press.
[55] A. Itai, J. Makowsky, Unification as a complexity measure for logic programming, J. Log. Program. 4 (1987) 105–177.
[56] A. Komuravelli, N. Bjørner, A. Gurfinkel, K.L. McMillan, Compositional verification of procedural programs using horn clauses over integers and arrays, in: R. Kaivola, T. Wahl (Eds.), Formal Methods in Computer-Aided Design, FMCAD 2015, Austin, Texas, USA, September 27-30, 2015, IEEE, 2015, pp. 89–96.
[57] J. Lang, Possibilistic logic: complexity and algorithms, in: D. Gabbay, Ph. Smets (Eds.), Handbook of Defeasible Reasoning and Uncertainty Management System, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001, pp. 179–220.
[58] S. Lehmke, A resolution-based axiomatization of 'bold' propositional fuzzy logic, in: Linz'96: Fuzzy Sets, Logics, and Artificial Intelligence, 1996, pp. 115–119.
[59] C.M. Li, F. Manyà, J.R. Soler, A tableau calculus for non-clausal maximum satisfiability, in: Automated Reasoning with Analytic Tableaux and Related Methods - 28th International Conference, Proceedings, TABLEAUX 2019, London, UK, September 3-5, 2019, 2019, pp. 58–73.
[60] V. Lifschitz, Answer Set Programming, Springer, 2019.
[61] V. Lifschitz, L.R. Tang, H. Turner, Nested expressions in logic programs, Ann. Math. Artif. Intell. 25 (3–4) (1999) 369–389.
[62] J. McKinsey, The decision problem for some classes of sentences without quantifiers, J. Symb. Log. 8 (1943) 61–76.
[63] R. Muhammad, P.J. Stuckey, A stochastic non-CNF SAT solver, in: Q. Yang, G.I. Webb (Eds.), PRICAI 2006: Trends in Artificial Intelligence, 9th Pacific Rim International Conference on Artificial Intelligence, Proceedings, Guilin, China, August 7–11, 2006, in: Lecture Notes in Computer Science, vol. 4099, Springer, 2006, pp. 120–129.
[64] N. Murray, Completely non-clausal theorem proving, Artif. Intell. 18 (1) (1982) 67–85.
[65] N. Murray, E. Rosenthal, Adapting classical inference techniques to multiple-valued logics using signed formulas, Fundam. Inform. 3 (21) (1994) 237–253.
[66] J. Navarro, A. Voronkov, Generation of hard non-clausal random satisfiability problems, in: The Twentieth National Conference on Artificial Intelligence, 2005, pp. 426–436.
[67] P. Nicolas, L. Garcia, I. Stéphan, C. Lefèvre, Possibilistic uncertainty handling for answer set programming, Ann. Math. Artif. Intell. 47 (1–2) (2006) 139–181.
[68] J.C. Nieves, H. Lindgren, Possibilistic nested logic programs, in: Technical Communications of the 28th International Conference on Logic Programming, ICLP 2012, September 4–8, 2012, Budapest, Hungary, 2012, pp. 267–276.
[69] J.C. Nieves, H. Lindgren, Possibilistic nested logic programs and strong equivalence, Int. J. Approx. Reason. 59 (2015) 1–19.
[70] J.C. Nieves, M. Osorio, U. Cortés, Semantics for possibilistic disjunctive programs, in: Logic Programming and Nonmonotonic Reasoning, 9th International Conference, Proceedings, LPNMR 2007, Tempe, AZ, USA, May 15–17, 2007, 2007, pp. 315–320.
[71] J.C. Nieves, M. Osorio, U. Cortés, Semantics for possibilistic disjunctive programs, Theory Pract. Log. Program. 13 (1) (2013) 33–70.
[72] B.E. Oliver, J. Otten, Equality preprocessing in connection calculi, in: Joint Proceedings of the 7th Workshop on Practical Aspects of Automated Reasoning (PAAR) and the 5th Satisfiability Checking and Symbolic Computation Workshop (SC-Square) Workshop, 2020 Co-Located with the 10th International Joint Conference on Automated Reasoning (IJCAR 2020), Paris, France, June-July, 2020 (Virtual), 2020, pp. 76–92.
[73] J. Otten, A non-clausal connection calculus, in: Automated Reasoning with Analytic Tableaux and Related Methods - 20th International Conference, Proceedings, TABLEAUX 2011, Bern, Switzerland, July 4–8, 2011, 2011, pp. 226–241.
[74] J. Otten, Non-clausal connection calculi for non-classical logics, in: Automated Reasoning with Analytic Tableaux and Related Methods - 26th International Conference, Proceedings, TABLEAUX 2017, Brasília, Brazil, September 25–28, 2017, 2017, pp. 209–227.
[75] C.H. Papadimitriou, Computational Complexity, Addison-Wesley, 1994.
[76] A. Platzer, Differential dynamic logic for hybrid systems, J. Autom. Reason. 41 (2) (2008) 143–189.
[77] E. Scala, P. Haslum, S. Thiébaux, M. Ramírez, Subgoaling techniques for satisficing and optimal numeric planning, J. Artif. Intell. Res. 68 (2020) 691–752.
[78] M. Seidl, A solver for quantified Boolean formulas in negation normal form, PhD thesis, Universität Wien, 2007.
[79] Z. Stachniak, Exploiting polarity in multiple-valued inference systems, in: 31st IEEE Int. Symp. on Multiple-Valued Logic, 2001, pp. 149–156.
[80] N. Tamura, A. Taga, S. Kitagawa, M. Banbara, Compiling finite linear CSP into SAT, Constraints Int. J. 14 (2) (2009) 254–272.
[81] C. Thiffault, F. Bacchus, T. Walsh, Solving non-clausal formulas with DPLL search, in: Tenth International Conference on Principles and Practice of Constraint Programming, 2004, pp. 663–678.
[82] H. Unno, T. Terauchi, Inferring simple solutions to recursion-free horn clauses via sampling, in: C. Baier, C. Tinelli (Eds.), Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, Proceedings, ETAPS 2015, London, UK, April 11-18, 2015, in: Lecture Notes in Computer Science, vol. 9035, Springer, 2015, pp. 149–163.

[83] P.F. Williams, A. Biere, E.M. Clarke, A. Gupta, Combining decision diagrams and SAT procedures for efficient symbolic model checking, in: Computer Aided Verification (CAV), in: LNCS, vol. 1855, 2000, pp. 124–138.
[84] Y. Xu, J. Liu, X. He, X. Zhong, S. Chen, Non-clausal multi-ary α-generalized resolution calculus for a finite lattice-valued logic, Int. J. Comput. Intell. Syst. 11 (1) (2018) 384–401.