Reasoning under inconsistency: the forgotten connective

Sébastien Konieczny
CRIL - Université d'Artois
62307 Lens, France
konieczny@cril.univ-artois.fr

Jérôme Lang
IRIT - Université Paul Sabatier
31062 Toulouse, France
lang@irit.fr

Pierre Marquis
CRIL - Université d'Artois
62307 Lens, France
marquis@cril.univ-artois.fr

Abstract

n any a o s o eason ng unde ncons s ency, s p c y assu ed a e o u ae o e be e base a e connec ed us ng a ea o conjunc on. en s cons s en , a be e base $B = \{\varphi_1, \ldots, \varphi_n\}$, e e φ_i a e p opos ona o u ae, s og ca y e u a en o e base $\{\varphi_1 \wedge \ldots \wedge \varphi_n\}$. o e e , en s no cons s en , bo bases yp ca y ead o d e en conc us ons. s us a es e ac a e co a used n base B as o be cons de ed as an add ona , genu ne connec e, and no as a s p e con unc on. n s o e de ne and n es ga e a p opos ona o a such a "co a connec e". e g e a se an cs and s o o gene a zes se e a app oac es o eason ng o ncons s en be e s.

1 Introduction

e e a e any d e en s ua ons a can ead o eason s se s o (u ua y ncons s en o u ae (see e.g. [oc and u e , 200] o a c ass ca on .

en dea ng an ncons s en be e base, a a ona agen us a n o us ng c ass ca en a en o e p o e be e base, s nce o an ncons s en be e base, e e y o u a s a og ca conse uence.

any app oac es a e been p oposed so a o add ess *the paraconsistency issue*, .e., e p ob e o des gn ng n e ence e a ons a do no necessa y a ze en e be e base s ncons s en , bu p ese e any e pec ed conse uences ne e e ess. A ong e , any app oac es o a o d a za on n p esence o a con ad c on cons s n *weakening the inconsistent belief base* by *inhibiting some formulae* n . o be o e p ec se, e ey dea o such app oac es s o se ec so e cons s en subse s o e be e base B, en o eason deduc e y o so e o e .

ea y enough, such app oac es e u e a be e base o be *a set* o o u ae, no a s ng e o u a (on cons s s n e o ng o u ae o e se . As a conse uence, a n e ncons s en se o o u ae canno be cons de ed as e u a en o e conjunc on o s e e en s (and s con as s a appens en cons s en se s a e cons de ed . o ns ance, e se o conse uences o $\{\varphi, \neg\varphi, \psi\}$ yp ca y d e s o e se o conse uences o $\{\varphi \wedge \neg\varphi \wedge \psi\}$.

and ng n a d e en ay conjunc ons o o u ae and se s o o u ae en ey a e c ass ca y ncons s en s an dea c unde es any pa acons s en og cs o decades. Such og cs a e ca ed *non-adjunctive logics*, c jus eans a e n e ence e c cons s s n cons de ng e conjunc e o u a $\phi \wedge \psi$ as de ab e o e se $\{\phi, \psi\}$ a s. o ns ance, as o s 's d scuss e og c [as o s , 969], one o e s pa acons s en og c po n ed ou so a , s non-adjunc e: n s og c, a p opos ona o u a φ s cons de ed as a conse uence o a se o o u ae $\{\varphi_1, \ldots, \varphi_n\}$ and on y $\diamond\varphi$ s a conse uence o $\{\diamond\varphi_1, \ldots, \diamond\varphi_n\}$ n ep s e c og c S5; s a oun s o say a φ s a conse uence o $\{\varphi_1, \ldots, \varphi_n\}$ n as o s 's og c and on y φ s a c ass ca conse uence o one o e φ_i's. s app oac as been u e ned by esc e and ano [esc e and ano , 970], o sugges o conjo n up o a a cons s ency, and en by any A esea c e s o n oduce conjo n ng po c es a can a e ad an age o add ona n o a on abou e e a e a n y o p eces o be e .

e e e ess, as a as e no and apa o e ecen o [eng n, 2004] d scussed n Sec on 8, cons de ng co a as a p n *connective* as ne e been o e s epped up o no , n pa cu a e anguages o p e ous pa acons s en og cs do no enab e se s o o u ae o be nes ed. e n ng and n es ga ng a og ca o a n c such a "co a connec e" s used s e a n pu pose o s pa pe , c s o gan zed as o o s. A e so e o a p e na es, e p esen bo e syn ac c a and e se an ca aspec s o e "co a og c" n Sec on 3. So e p ope es o e co espond ng n e ence e a on a e g en n Sec on 4. e gene a y o e p oposed o a s d scussed n Sec on 5. So e a s a ab y esu s and so e co p e y esu s a e g en n Sec on 6. An a e na e se an cs o e co a connec e based on ca d na y s b e y d scussed n Sec on 7. ue o space a ons, os p oo s a e on y s e c ed. Sec on 8 conc udes e pape .

2 Preliminaries

Le PS be a *finite* se o p opos ona a ab es: a, b, c, e c. $\mathcal{L}_{PS}$ s e p opos ona anguage bu o PS, e cons an s $\top$ (e u and $\bot$ (a su and e s anda d connec es $\{\wedge, \vee, \neg, \rightarrow\}$ n e usua ay. o u ae o $\mathcal{L}_{PS}$ a e no ed s a G ee e e s φ, ψ, e c. An n e p e a on (o o d M s a o a unc on o PS o $\{0, 1\}$, yp ca y ep esen ed as the set of all the variables of PS it satisfies. $\mathcal{M}_{PS}$ denotes the set of all interpretations built over PS. M is a model of a formula φ, denoted $M \models \varphi$, if and only if it makes φ true (in the usual truth-functional way). $Mod(\varphi)$ denotes the set of models of φ. $\models$ denotes classical entailment and $\equiv$ denotes equivalence between formulae of $\mathcal{L}_{PS}$.

Let $\subset$ denote strict inclusion. Let $B = \{\varphi_1, \ldots, \varphi_n\}$ be a finite set of formulae of $\mathcal{L}_{PS}$, and let $B' \subseteq B$, we note $\bigwedge B' = \bigwedge\{\varphi_i \mid \varphi_i \in B'\}$. B' is a *consistent* subset of B if and only if $\bigwedge B'$ is consistent. B' is a *maximal consistent* subset (or *maxcons*) of B if and only if B' is consistent and there is no consistent B'' such that $B' \subset B'' \subseteq B$. $MaxCons(B)$ denotes the set of the maximal consistent subsets of B.

3 The "comma logic"

Let us start with the definition of the language of our comma logic.

Definition 1 *The language $\mathcal{L}^C_{PS}$ of the propositional comma logic is defined inductively as follows:*

- $PS \cup \{\top, \bot\} \subseteq \mathcal{L}^C_{PS}$;
- *If $\varphi \in \mathcal{L}^C_{PS}$ and $\psi \in \mathcal{L}^C_{PS}$ then $(\neg\varphi)$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$ belong to $\mathcal{L}^C_{PS}$;*
- *If $n \geq 0$ and $\{\varphi_1, \ldots, \varphi_n\} \subseteq \mathcal{L}^C_{PS}$ then $\langle\varphi_1, \ldots, \varphi_n\rangle$ belongs to $\mathcal{L}^C_{PS}$.*

Parentheses can be omitted when no ambiguity is possible.

Clearly enough, $\mathcal{L}^C_{PS}$ is a superset of $\mathcal{L}_{PS}$: nonclassical formulae obtained by taking advantage of the comma connective can be formed. Here, we abuse words slightly: comma is actually considered as a set constructor, hence formally comma denotes a family of connectives, one per possible arity. Especially, comma is not always binary: one can apply the comma connective to no formula at all (the empty set), one formula (singletons), or more than two formulae. This forces us to introduce additional separators (the chevrons $\langle$ and $\rangle$ – which must not be confused with parentheses but enables to keep the notations not too heavy.

It proves also sometimes interesting to define the *depth* of a formula of $\mathcal{L}^C_{PS}$, and the languages $\{\mathcal{L}^{C[n]}_{PS}, n \in \mathbb{N}\}$ as:

Definition 2 *The* depth *of a formula φ from $\mathcal{L}^C_{PS}$ is defined inductively as follows:*

- *If $\varphi \in PS \cup \{\top, \bot\}$ then $depth(\varphi) = 0$;*
- $depth(\neg\varphi) = depth(\varphi)$;
- $depth(\varphi \wedge \psi) = depth(\varphi \vee \psi) = depth(\varphi \rightarrow \psi) = \max(depth(\varphi), depth(\psi))$;
- $depth(\langle\varphi_1, \ldots, \varphi_n\rangle) = \max_{i=1\ldots n} depth(\varphi_i) + 1$.[1]

For $n \in \mathbb{N}$, $\mathcal{L}^{C[n]}_{PS}$ is the set of all formulae of $\mathcal{L}^C_{PS}$ whose depth is less or equal to n.

One can easily check that $\mathcal{L}^C_{PS} = \bigcup_{n \in \mathbb{N}} \mathcal{L}^{C[n]}_{PS}$ and $\mathcal{L}^{C[0]}_{PS} = \mathcal{L}_{PS}$. Finally, for all $n \geq 1$, we define the language $\mathcal{L}^{C[s,n]}_{PS}$ (set of *simple formulae of depth n*) as the subset of $\mathcal{L}^{C[n]}_{PS}$ containing all formulae $\langle\varphi_1, \ldots, \varphi_k\rangle$, where $\varphi_1, \ldots, \varphi_k$ are formulae of $\mathcal{L}^{C[n-1]}_{PS}$, i.e., the set of formulae of depth n where comma is the main functor. Here are some examples of formulae of $\mathcal{L}^C_{PS}$, where $PS = \{a, b, c\}$:

- $\varphi_1 = a \wedge b \wedge (\neg b \vee c) \wedge \neg c$;
- $\varphi_2 = \langle a \wedge b, \neg b \vee c, \neg c\rangle$;
- $\varphi_3 = \langle a \wedge (b \vee \langle c, a \vee b, \neg c\rangle), \langle \neg b \vee c, \neg c\rangle \wedge \langle a, \top\rangle\rangle$;
- $\varphi_4 = \langle\ \rangle$.

We have $depth(\varphi_1) = 0$; $depth(\varphi_2) = 1$; $depth(\varphi_3) = 3$. Note also that $\varphi_2 \in \mathcal{L}^{C[s,1]}_{PS}$ (as φ_4) and that $\varphi_3 \in \mathcal{L}^{C[s,3]}_{PS}$.

Let us now present the semantics of our new language:

Definition 3 *The notion of* satisfaction *of a formula of $\mathcal{L}^C_{PS}$ by an interpretation M is given by the relation $\models_c$ on $\mathcal{M}_{PS} \times \mathcal{L}^C_{PS}$ defined inductively as:*

- $M \models_c \top$;
- $M \not\models_c \bot$;
- *If $a \in PS$, $M \models_c a$ if and only if $a \in M$;*
- *$M \models_c \neg\varphi$ if and only if $M \not\models_c \varphi$;*
- *$M \models_c \varphi \wedge \psi$ if and only if $M \models_c \varphi$ and $M \models_c \psi$;*
- *$M \models_c \varphi \vee \psi$ if and only if $M \models_c \varphi$ or $M \models_c \psi$;*
- *$M \models_c \varphi \rightarrow \psi$ if and only if $M \models_c \neg\varphi$ or $M \models_c \psi$;*
- *$M \models_c \langle\varphi_1, \ldots, \varphi_n\rangle$ if and only if there is no interpretation $M' \in \mathcal{M}_{PS}$ such that $\{i \mid M' \models_c \varphi_i\}$ strictly contains $\{i \mid M \models_c \varphi_i\}$.*

We now extend several standard notions (and the corresponding notations) from classical logic to comma logic. Let $Mod_C(\varphi) = \{M \in \mathcal{M}_{PS} \mid M \models_c \varphi\}$. A formula of $\mathcal{L}^C_{PS}$ is said to be *consistent* (or *satisfiable*) if and only if $Mod_C(\varphi) \neq \emptyset$, and *valid* if and only if $Mod_C(\varphi) = \mathcal{M}_{PS}$. If φ and ψ are two formulae of $\mathcal{L}^C_{PS}$, ψ is said to be a consequence of φ (denoted $\varphi \models_c \psi$) if and only if $Mod_C(\varphi) \subseteq Mod_C(\psi)$. When $\varphi = \top$, we simply write $\models_c \psi$ instead of $\top \models_c \psi$. Two formulae φ and ψ of $\mathcal{L}^C_{PS}$ are said to be *equivalent*, noted $\varphi \equiv_c \psi$, if and only if $\varphi \models_c \psi$ and $\psi \models_c \varphi$.

4 Some logical properties

It is easy to show that, in contrast to the other connectives in the logic, the comma connective is not truth-functional: in the general case, the truth value of $\langle\varphi_1, \ldots, \varphi_n\rangle$ from $\mathcal{L}^C_{PS}$ in interpretation M cannot be determined from the truth values of each φ_i ($i \in 1 \ldots n$) in M only. For instance, let $PS = \{a, b\}$, $M = \{a\}$, $\varphi_1 = \langle a, b\rangle$, $\varphi_2 = \langle a \wedge \neg b, b\rangle$: M satisfies the "first" element of φ_1 and φ_2, but does not satisfy the "second" one; while M is a model of φ_2, it is not a model of φ_1. This lack of truth-functionality explains why the comma connective is really genuine in the logic. (It cannot be defined by just combining connectives of classical logic.)

Actually, the discrepancy between classical logic and comma logic lies only in the comma connective, since:

Proposition 1 *Comma logic is a* conservative extension *of classical logic: for every pair of formulae φ, ψ in which the comma connective does not occur, then $\varphi \models_c \psi$ if and only if $\varphi \models \psi$.*

[1]By convention $depth(\langle\,\rangle) = 1$

Obviously, new valid formulae are obtained in comma logic, like $\langle\varphi,\neg\varphi\rangle$, because the language of propositional logic has been enriched, but in any case classical theorems are preserved.

Furthermore, the replacement theorem of classical logic can be extended to comma logic:

Proposition 2 *Let φ be a valid formula from $\mathcal{L}^C_{PS}$ and let $x \in PS$. Let ψ be any formula from $\mathcal{L}^C_{PS}$. The formula obtained by replacing in φ every occurrence of x by an occurrence of ψ is valid as well.*

Let us now make precise some properties of the inference relation $\models_c$. First, a direct consequence of its definition is that $\models_c$ is a *Tarskian consequence relation* (since it is defined as model containment; from this observation, a number of valid properties follows, especially the fact that $\models_c$ is reflexive and transitive). By the way, note that when in classical logic the basic entailment relation $\models$ on $\mathcal{L}_{PS}\times\mathcal{L}_{PS}$ can be easily extended to a relation on $2^{\mathcal{L}_{PS}}\times\mathcal{L}_{PS}$ by defining the models of a set of formulae as the interpretations satisfying every formula of these sets, these structures are not meaningful here; indeed, in comma logic, sets of formulae are interpreted in a nonstandard way. This renders the usual characterization of Tarskian consequence relations inadequate in our setting since $\models_c$ does satisfy neither strong reflexivity nor monotonicity:

- $\langle a,\neg a\rangle \not\models_c a$;
- $\langle a\rangle \models_c a$ but $\langle a,\neg a\rangle \not\models_c a$.

Here are some other valuable properties:

Proposition 3

1. *If $\varphi_1 \models_c \varphi_2$ and $\langle\varphi_1,\varphi_2\rangle \models_c \psi$, then $\varphi_1 \models_c \psi$.*
2. *If $\langle\varphi_1,\varphi_2\rangle \models_c \psi$, then $\varphi_1 \models_c (\varphi_2 \rightarrow \psi)$.*
3. *If $\varphi_1 \equiv_c \varphi_2$ and φ_1 is a subformula of ψ, then any formula obtained by replacing in ψ occurrences of φ_1 by φ_2 is equivalent to ψ.*
4. *$\varphi \models_c \psi_1 \wedge \psi_2$ if and only if $\varphi \models_c \psi_1$ and $\varphi \models_c \psi_2$.*
5. *If $\langle\varphi,\varphi_1\rangle \models_c \psi$ and $\langle\varphi,\varphi_2\rangle \models_c \psi$, then $\langle\varphi,(\varphi_1 \vee \varphi_2)\rangle \models_c \psi$.*

Item (1) is a form of the cut rule. Item (2) is a form of the deduction theorem. Item (3) is the substitution theorem for the "comma logic". In particular, it shows that $\forall k$ if $\forall i \in 1\ldots k$ $\varphi_i \equiv_c \varphi'_i$ then $\langle\varphi_1,\ldots,\varphi_k,\ldots,\varphi_n\rangle \equiv_c \langle\varphi'_1,\ldots,\varphi'_k,\ldots,\varphi_n\rangle$. Item (4) is a form of the "and rule". Item (5) is a form of proof by cases (or "or rule").

Other standard theorems of classical logic fail in the general case for full comma logic, i.e., when sets of formulae are considered. For instance, modus ponens fails ($\langle a,(a\rightarrow\neg a)\rangle \not\models_c \neg a$), and-elimination and or-introduction as well ($\langle\bot,\varphi\rangle \not\models_c \bot$ and $\langle\bot\rangle \not\models_c \bot\vee\varphi$); reductio ad absurdum fails ($\varphi \models_c \varphi$ while $\langle\varphi,\neg\varphi\rangle \not\models_c \bot$).

Let us now present additional properties which characterize how the comma connective interacts with the other connectives (and itself).

Proposition 4

6. *For any permutation σ of $\{1,\ldots,n\}$, $\langle\varphi_1,\ldots,\varphi_n\rangle$ is equivalent to $\langle\varphi_{\sigma(1)},\ldots,\varphi_{\sigma(n)}\rangle$.*
7. *$\langle\varphi_1,\ldots,\varphi_1,\varphi_2,\ldots,\varphi_n\rangle \equiv_c \langle\varphi_1,\ldots,\varphi_n\rangle$.*
8. *$\forall n \geq 0$ $\langle\varphi_1,\ldots,\varphi_n\rangle \not\models_c \bot$.*
9. *$\langle\varphi_1,\varphi_2,\ldots,\varphi_n\rangle \equiv_c \langle\varphi_2,\ldots,\varphi_n\rangle$ if φ_1 is inconsistent or ($\exists i \in 2\ldots n$, φ_i is consistent and $\forall i \in 2\ldots n$, $\varphi_i \models \varphi_1$).*
10. *$\varphi_1\wedge\ldots\wedge\varphi_n \models_c \langle\varphi_1,\ldots,\varphi_n\rangle$.*
11. *$\langle\varphi_1,\ldots,\varphi_n\rangle\wedge\langle\psi_1,\ldots,\psi_m\rangle \models_c \langle\varphi_1\wedge\psi_1,\ldots,\varphi_1\wedge\psi_m,\ldots,\varphi_n\wedge\psi_1,\ldots,\varphi_n\wedge\psi_m\rangle$.*
12. *$\langle\varphi_1,\ldots,\varphi_n\rangle \wedge \langle\psi_1,\ldots,\psi_m\rangle \models_c \langle\varphi_1,\ldots,\varphi_n,\psi_1,\ldots,\psi_m\rangle$.*
13. *If φ_i is consistent (for at least one $i \in 1\ldots n$), then $\langle\varphi_1,\ldots,\varphi_n\rangle \models_c \varphi_1\vee\ldots\vee\varphi_n$.*
14. *If φ is consistent,
then $\langle\varphi_1\vee\varphi,\ldots,\varphi_n\vee\varphi\rangle \equiv_c (\varphi_1\wedge\ldots\wedge\varphi_n)\vee\varphi$.*
15. *$\neg(\neg\varphi) \equiv_c \varphi$.*

Item (6) shows that the "comma connective" is symmetric in every argument. Item (7) shows that the "comma connective" is idempotent.

Item (8) shows that for every $n > 0$, every formula from $\mathcal{L}^{C[s,n]}_{PS}$ is consistent (obviously, this cannot be extended to the full language $\mathcal{L}^C_{PS}$). This is sufficient to ensure that $\models_c$ achieves a form of paraconsistency when it is applied to a set of formulae.

Item (9) gives a sufficient condition to simplify a set of formulae (note that in contrast to classical logic, it is not possible in general to remove a formula from a set as soon as it is entailed by another formula; for instance, while $a\wedge b \models_c a\vee b$, we do not have $\langle a\wedge b,\neg a,a\vee b\rangle$ equivalent to $\langle a\wedge b,\neg a\rangle$). Items (10), (11), (12) make precise the connection between the comma connective and conjunction. None of the converse relations holds in the general case (for (10), $\varphi_1\wedge\ldots\wedge\varphi_n$ should be consistent; the converse of (11) does not hold even in the restricted case $m = 1$ and $\langle\varphi_1,\ldots,\varphi_n\rangle\wedge\psi_1$ is consistent: $\langle a\wedge\neg b,\neg a\wedge\neg b,\neg a\wedge\neg b\rangle \not\models_c \langle a,\neg a\vee b,\neg a\rangle\wedge\neg b$; the converse of (12) fails since for instance $\langle a,\neg a\rangle \not\models_c \langle a\rangle \wedge \langle\neg a\rangle$). Note also that there is in general no deduction connections between a formula where nested commas occur and its "unfold" counterpart. Especially, in general, we have:

$$\langle\langle\varphi_1,\ldots,\varphi_n\rangle,\langle\psi_1,\ldots,\psi_m\rangle\rangle \not\models_c \langle\varphi_1,\ldots,\varphi_n,\psi_1,\ldots,\psi_m\rangle$$

For example, $\langle\langle a,b\rangle,\langle\neg a,c\rangle\rangle \not\models_c \langle a,\neg a,b,c\rangle$.

In general, we also have:

$$\langle\varphi_1,\ldots,\varphi_n,\psi_1,\ldots,\psi_m\rangle \not\models_c \langle\langle\varphi_1,\ldots,\varphi_n\rangle,\langle\psi_1,\ldots,\psi_m\rangle\rangle$$

For example, $\langle a,\neg a,b,c\rangle \not\models_c \langle\langle a,b,\neg a\rangle,\langle\neg a,c\rangle\rangle$.

Items (13) and (14) make precise the connection between the comma connective and disjunction. No other significant properties involving disjunction hold; for instance $\langle a,b,\neg b\rangle\vee\neg a \not\models_c \langle a\vee\neg a,b\vee\neg a,\neg b\vee\neg a\rangle$. Together with the previous properties concerning conjunction, it shows that we can consider the comma connective as a "weak" conjunction connective, since it is equivalent to it when the conjunction is consistent, but gives a consistent result even when it is not. Furthermore, when all the formulae are pairwise contradictory but one of them is consistent, comma is equivalent to the disjunction connective. Thus, if we specialize to the case where two formulae φ and ψ are considered, we get the easy corollary:

Corollary 1 $\langle\varphi,\psi\rangle \equiv_c \begin{cases} \varphi\wedge\psi & \textit{if consistent, else} \\ \varphi\vee\psi & \textit{if consistent, else} \\ \top & \textit{otherwise.} \end{cases}$

Item (5) finally states that negation is involutive for the full logic (in particular, $\neg(\neg\langle\varphi_1,\ldots,\varphi_n\rangle) \equiv_c \langle\varphi_1,\ldots,\varphi_n\rangle$). No other link between negation and the comma connective can be established in the general case (for instance, $\neg$ does not distribute over the "comma connective"; for instance, $\neg\langle\bot\rangle \equiv_c \bot$ while $\langle\neg\bot\rangle \equiv_c \top$).

5 Generality of the framework

Let us now show how several approaches for dealing with inconsistency can be encoded in this logic.

5.1 Skeptical inference

One of the most common method for reasoning from an inconsistent set of formulae consists in selecting the maximal (w.r.t. set inclusion) consistent subsets (also called *maxcons*) of formulae (see [Rescher and Manor, 1970] for example. Consequences w.r.t. skeptical inference (also called "universal" inference) are then defined as classical consequences of every maxcons:

Definition 4 $\Delta \vdash_\forall \varphi$ *iff* $\forall M \in MaxCons(\Delta)$, $M \models \varphi$.

Such an inference relation can be easily recovered in our comma logic:

Proposition 5 $\{\varphi_1,\ldots,\varphi_n\} \vdash_\forall \varphi$ *iff* $\langle\varphi_1,\ldots,\varphi_n\rangle \models_c \varphi$.

5.2 Credulous inference

From a set of maxcons, one can also consider credulous inference (also called "existential" inference), that allows to infer a formula if at least one maxcons classically entails it.

Definition 5 $\Delta \vdash_\exists \varphi$ *iff* $\exists M \in MaxCons(\Delta)$, $M \models \varphi$.

Again, such an inference relation can be easily recovered in our comma logic:

Proposition 6 $\{\varphi_1,\ldots,\varphi_n\} \vdash_\exists \varphi$ *iff* $\langle\varphi_1,\ldots,\varphi_n,\neg\varphi\rangle \wedge \varphi \not\models_c \bot$.

5.3 Supernormal defaults

Assumption-based theories and supernormal default logic [Poole, 1988; Brewka, 1989] can be described in the following way. Let $\langle B, B_d\rangle$ be a pair, where B and B_d are two finite sets of formulae. B is the set of facts (or hard constraints), that is supposed to be consistent. B_d is a set of defaults, i.e., a set of formulae that one wishes to add to facts if it does not result in an inconsistency. B_d is often not consistent with B.

An extension is a subset of formulae of $B \cup B_d$ that contains all the formulae of B and a maximum (w.r.t. set inclusion) of formulae of B_d. So, the set of all the extensions is defined as: $Extens(\langle B,B_d\rangle) = \{E \mid E \subseteq B\cup B_d$ and $B \subseteq E$ and $E \not\models \bot$ and $\forall E'$ s.t. $E \subset E' \subseteq B \cup B_d, E' \models \bot\}$.

Inference from such a supernormal default theory $\langle B, B_d\rangle$ is defined as:

Definition 6 $\langle B,B_d\rangle \vdash_D \varphi$ *iff* $\forall E \in Extens(\langle B,B_d\rangle)$, $E \models \varphi$.

This can be also easily expressed within the comma logic (note that since the set B of facts is consistent, it can be considered conjunctively):

Proposition 7 $\langle B,\{\varphi_1,\ldots,\varphi_n\}\rangle \vdash_D \varphi$ *if and only if* $\langle\varphi_1 \wedge \bigwedge B,\ldots,\varphi_n \wedge \bigwedge B\rangle \wedge \bigwedge B \models_c \varphi$.

Finally, note that since circumscription can be encoded as skeptical inference from supernormal default (see e.g., [Etherington, 1987]), it can also be encoded in the comma logic.

5.4 Belief revision

The basic revision operator proposed by Nebel [Nebel, 1991; 1998; Fagin *et al.*, 1983] is defined as follows:

Definition 7 *Let* $B\bot\varphi = \{M \mid M \subseteq B$ *and* $M \not\models \varphi$ *and* $\forall M'$ *s.t.* $M \subset M' \subseteq B, M' \models \varphi\}$. *Basic revision is then defined as:* $B \circ_N \varphi = (\bigvee_{M\in B\bot\neg\varphi} \bigwedge M) \wedge \varphi$.

In comma logic, this can be expressed as follows:

Proposition 8 $\{\varphi_1,\ldots,\varphi_n\} \circ_N \varphi \models \mu$ *if and only if* $\langle\varphi_1\wedge\varphi,\ldots,\varphi_n\wedge\varphi\rangle\wedge\varphi \models_c \mu$.

5.5 Belief merging

Suppose that one wants to merge several belief (or goal) bases with allowing some of those bases to be inconsistent. If one wants to take into account inconsistent belief bases in the merging process, one can first perform an intra-source merging in order to extract useful information from those bases, before making the proper inter-source merging.

Let $E = \{B_1,\ldots,B_n\}$ be a set of (possibly inconsistent) belief bases s.t. for all $i \in 1\ldots n$, we have $B_i = \{\varphi^i_1,\ldots,\varphi^i_{k_i}\}$. Then, we can define the merging of those bases as:

$$\triangle(E) = \langle\langle\varphi^1_1,\ldots,\varphi^1_{k_1}\rangle,\ldots,\langle\varphi^n_1,\ldots,\varphi^n_{k_n}\rangle\rangle$$

This operator is not definable as a two aggregation steps merging operator (as defined in [Konieczny *et al.*, 2002]). Conversely, considering an alternative semantics for the comma connective (see Section 7), we can capture some of the two aggregation steps merging operators in comma logic.

6 Computational aspects

Let us investigate both translatability issues and complexity issues for comma logic. First of all, one can prove that the expressiveness of the comma logic is exactly the same as the expressiveness of classical logic (every piece of information which can be encoded in one of the logics can also be encoded in the other one):

Proposition 9 *For any formula* φ *of* $\mathcal{L}^C_{PS}$, *there is a formula* ψ *of* $\mathcal{L}^{C[0]}_{PS} = \mathcal{L}_{PS}$ *that is equivalent to* φ.

A simple induction on the depth of the formulae shows that each formula φ of $\mathcal{L}^C_{PS}$ can be equivalently (for $\models_c$) rewritten into a classical formula $C(\varphi)$. Let φ be a formula of $\mathcal{L}^C_{PS}$ of depth k. A translation of φ into a classical formula can be obtained as follows. First, replace every subformula of φ

that is in $\mathcal{L}_{PS}^{C[s,1]}$ by an equivalent formula of $\mathcal{L}_{PS}^{C[0]}$. The substitution theorem for the "comma logic" makes this sound; one gets then a formula of depth $n-1$, so we can iterate the process until a formula of depth 0 is obtained. For example, let $\varphi = \langle a, b, \langle \neg b \vee c, \neg c\rangle\rangle$; we start by translating $\langle \neg b \vee c, \neg c\rangle$ in $\neg b \wedge \neg c$: after one iteration, we get the formula $\langle a, b, \neg b \wedge \neg c\rangle$, and a second iteration gives $(a \wedge b) \vee (a \wedge \neg b \wedge \neg c)$, that is equivalent to $a \wedge (b \vee \neg c)$. The problem with such a translation is that the size of $C(\varphi)$ is generally exponentially larger than the size of φ. Can we do better? Actually, the answer is negative:

Proposition 10 *Under the usual assumptions of complexity theory,[2] there is no polysize function that maps any formula from $\mathcal{L}_{PS}^{C}$ (and even from $\mathcal{L}_{PS}^{C[1]}$) to*

- *an equivalent classical formula.*
- *an equivalent formula from $\mathcal{L}_{PS}^{C}$ in which the "comma connective" is binary (i.e., whenever it is used, it may connect at most two formulae).*

The first result is a direct consequence of the non-compilability of circumscription [Cadoli *et al.*, 1997]. The second is a consequence of the first one, given that every formula φ from $\mathcal{L}_{PS}^{C}$ in which the comma connective is binary can be turned into a classical formula whose size is bounded by $|\varphi|$ (the proof is easy by structural induction given Corollary 1).

Let us now turn to the inference issue. From Proposition 3, it is easy to show that $\varphi \models_c \psi$ if and only if $\varphi \wedge \neg\psi$ is inconsistent. Hence, it is sufficient to focus on the satisfiability issue. The satisfiability problem in $\mathcal{L}_{PS}^{C}$ is, of course, at least as hard than for classical logic (so it is NP-hard). Can we say more than that? Let us begin with $\mathcal{L}_{PS}^{C[1]}$.

Proposition 11 *The satisfiability problem in $\mathcal{L}_{PS}^{C[1]}$ is Σ_2^p-complete.*

Membership to Σ_2^p comes from the fact that the corresponding model checking problem is in coNP: for any simple subformula $\psi = \langle\alpha_1,\ldots,\alpha_n\rangle$ of φ from $\mathcal{L}_{PS}^{C[s,1]}$, deciding whether $M \not\models_c \psi$ simply consists in guessing an interpretation M' and to check in polynomial time that the set of α_i satisfied by M' is a proper superset of the set of α_i satisfied by M. Actually, the truth values of all such ψ in M can be established in nondeterministic polynomial time (guess in parallel a set M', one for each simple subformula ψ of depth 1 which is not satisfied by M). Once this has been done, the truth value of φ in M can be computed in deterministic polynomial time. Hardness comes from Σ_2^p-hardness of credulous inference from maximal consistent subbases [Nebel, 1998; Cayrol and Lagasquie-Schiex, 1995].

Now, what about the satisfaction problem when commas are nested? An iterated use of the proof of the preceding result (membership part) easily leads to the following:

Proposition 12

- *The satisfiability problem in $\mathcal{L}_{PS}^{C[n]}$ is in Σ_{n+1}^p.*
- *The satisfiability problem in $\mathcal{L}_{PS}^{C}$ is in PSPACE.*

[2] Especially the fact that the polynomial hierarchy does not collapse.

We do not have hardness proofs but we conjecture them.[3] Thus, we do not know how to perform compilation for the nested occurrences in polynomial time (and we conjecture that it is not possible to do it; especially, introducing new symbols would not necessarily preserve equivalence to the original language; for instance, $\langle a, \langle\neg a, b\rangle\rangle \not\equiv_c \langle a, new\rangle \wedge (new \Leftrightarrow \langle\neg a, b\rangle)$).

Let us now give a few comments about those complexity results. First, the fact that satisfiability in $\mathcal{L}_{PS}^{C[1]}$ is at the second level of the polynomial hierarchy, and not higher, are good news. We know from Section 5 that universal and existential inference problems can be reduced, in polynomial time, to satisfaction or validity problems of formulae of $\mathcal{L}_{PS}^{C[1]}$. Given complexity results from [Cayrol *et al.*, 1998; Nebel, 1998], it follows that the satisfiability problem in $\mathcal{L}_{PS}^{C[1]}$ cannot be lower than the second level of the polynomial hierarchy. The fact that it is not higher shows that the gain of flexibility we get does not result in a complexity shift.

7 Variation on the "comma semantics"

Among others, another semantics for the comma connective is obtained by considering as "models" of $\langle\varphi_1,\ldots,\varphi_n\rangle$ the interpretations that maximize the *number* of satisfied formulae φ_i. Formally, this amounts to replacing in Definition 3, the item $M\models_c\langle\varphi_1,\ldots,\varphi_n\rangle$ by the following one:

- $M\models_c^C\langle\varphi_1,\ldots,\varphi_n\rangle$ if and only if there is no interpretation $M' \in \mathcal{M}_{PS}$ such that $|\{i \mid M'\models_c\varphi_i\}| > |\{i \mid M\models_c\varphi_i\}|$.

Properties reported in Section 4 are typically preserved by this modification, but idempotence: $\langle\varphi_1,\varphi_1,\varphi_2,\ldots,\varphi_n\rangle$ is generally not equivalent to $\langle\varphi_1,\varphi_2,\ldots,\varphi_n\rangle$. For example, $\langle a, a, \neg a\rangle$ is not equivalent to $\langle a, \neg a\rangle$ under the cardinality maximality semantics.

Contrariwise to the $\models_c$ case, we can show that:

Proposition 13 *For every fixed n, there is a polysize function that maps any formula of $\mathcal{L}_{PS}^{C[n]}$ into a query-equivalent formula of $\mathcal{L}_{PS}$ when we appeal to the $\models_c^C$ semantics.*

The proof is constructive and relies on a translation function C similar to the one given in Section 6: we first replace each occurrence of the formulae $\psi \doteq \langle\alpha_1,\ldots,\alpha_n\rangle$ of $\mathcal{L}_{PS}^{C[s,1]}$ in φ by the *cardinality formulae* $[= k_\psi] : \langle\alpha_1,\ldots,\alpha_n\rangle$, meaning that exactly k_ψ formulae among the n formulae $\alpha_1,\ldots,\alpha_n$ must be satisfied (see e.g. [Benhamou *et al.*, 1994]); such cardinality formulae can be translated in polynomial space as classical query-equivalent formulae. Such a translatability result is the main objective of compiling formulae from $\mathcal{L}_{PS}^{C[k]}$ into classical formulae so as to improve inference.

Proposition 14 *The satisfiability problem in $\mathcal{L}_{PS}^{C}$ for $\models_c^C$ is in Δ_2^p.*

[3] t t e co ne t n e oe ee n ypr e t co pe t y ncre e nq rpr n r p eno enon p pen en condt on re ne t ed ee e Et er nd qt o

Membership to Δ_2^p comes from the fact that one can associate to any simple subformula $\psi = \langle\alpha_1, \ldots, \alpha_n\rangle$ of $\varphi \in \mathcal{L}_{PS}^C$ the maximal number k_ψ of α_i which can be jointly satisfied. Simple subformulae of φ are processed by increasing depth: for any ψ of depth 1, the α_i are classical formulae, and using binary search, logarithmically many calls to an NP oracle are sufficient to compute the corresponding k_ψ. For any ψ of depth 2, again using binary search, logarithmically many calls to an NP oracle are sufficient to compute the corresponding k_ψ: the key observation is that the model checking issue for ψ can be done in polynomial time once the $k_{\psi'}$ for each simple subformula ψ' of ψ have been computed ($M \models_c^C \psi'$ iff and only if M satisfies $k_{\psi'}$ elements in ψ', and so on. Since φ does not contain more than $|\varphi|$ simple subformulae, linearly many calls to an NP oracle are sufficient to compute all the k_ψ; since model checking is in P once they have been computed, a last call to an NP oracle is sufficient to determine whether φ is satisfiable. We conjecture that the problem is Δ_2^p-complete but only have a proof of Θ_2^p-hardness at that time.

One has to note that the complexity gap (between the first and the second level of the polynomial hierarchy) highlighted by [Cayrol et al., 1998] when one goes from the cardinality criterion to the set-inclusion one, becomes here an amazing gap (from Δ_2^p to PSPACE) – under the assumption of completeness for PSPACE, of course.

8 Conclusion

In this paper, we have given a "logic" for paraconsistent reasoning where conciliation is considered as a genuine connective.

We have presented some properties of this logic, and we have shown that it allows to express, in a unified language, several problems of reasoning with inconsistent belief bases, nonmonotonic inference, belief revision and belief merging. We have also provided some translatability results and some complexity results. Finally, we have given an alternative semantics for the conciliation connective.

A distinction between a set of formulas and the conjunction of them is also made in many substructural logics, i.e. logics where weakening and the classical one because of the absence of some structural rules [Restall, 1999]. The "conciliation connective" in such logics is not necessarily commutative or idempotent. Unlike in our conciliation logic, it is binary and cannot be in the scope of every other connective under consideration in the language.

Another closely related work is [Mengin, 2004], which defined a more general framework than ours: it can be shown indeed that any formula of our language can be translated in a formula which is a consequence of the equivalence given in this Section 3.3 for formulas of depth one together with our Proposition 2. Therefore, our logical properties and complexity results carry on to the framework of [Mengin, 2004].

Acknowledgements

The authors would like to thank the anonymous reviewers for their comments. This work has been supported by the IUT de Lens, the [illegible] Consortium, the Région Nord/Pas-de-Calais and the European Community [illegible] program.

References

[Benhamou et al., 1994] B. Benhamou, L. Saïs, and P. Siegel. Two proof procedures for a cardinality based language in propositional calculus. In *STACS'94*, pages 71–82. USA, 1994.

[Bloch and Hunter, 2001] I. Bloch and A. Hunter, editors. *Fusion: General Concepts and Characteristics*, volume 16 (10) of *International Journal of Intelligent Systems*. 2001.

[Brewka, 1989] G. Brewka. Preferred subtheories: an extended logical framework for default reasoning. In *IJCAI'89*, pages 1043–1048, 1989.

[Cadoli et al., 1997] M. Cadoli, F.M. Donini, M. Schaerf, and R. Silvestri. On compact representations of propositional circumscription. *Theoretical Computer Science*, 182: 183–202, 1997.

[Cayrol and Lagasquie-Schiex, 1995] C. Cayrol and M.-C. Lagasquie-Schiex. Non-monotonic syntax-based entailment: a classification of consequence relations. In *ECSQARU'95*, pages 107–114, 1995.

[Cayrol et al., 1998] C. Cayrol, M.-C. Lagasquie-Schiex, and Th. Schiex. Nonmonotonic reasoning: from complexity to algorithms. *Annals of Mathematics and Artificial Intelligence*, 22(3-4):207–236, 1998.

[Eiter and Gottlob, 1993] Th. Eiter and G. Gottlob. The complexity of nested counterfactuals and iterated knowledge base revisions. In *IJCAI'93*, pages 526–531, 1993.

[Etherington, 1987] D. W. Etherington. Relating default logic and circumscription. In *IJCAI'87*, pages 489–494, 1987.

[Fagin et al., 1983] R. Fagin, J. D. Ullman, and M. Y. Vardi. On the semantics of updates in databases. In *PODS'83*, pages 352–365, 1983.

[Jaśkowski, 1969] S. Jaśkowski. Propositional calculus for contradictory deductive systems. *Studia Logica*, 24: 143–157, 1969.

[Konieczny et al., 2002] S. Konieczny, J. Lang, and P. Marquis. Distance-based merging: a general framework and some complexity results. In *KR'02*, pages 97–108, 2002.

[Mengin, 2004] J. Mengin. Logical connectives for nonmonotonicity: A choice function-based approach. In *JELIA'2004*, pages 452–461, 2004.

[Nebel, 1991] B. Nebel. Belief revision and default reasoning: Syntax-based approaches. In *KR'91*, pages 417–428, 1991.

[Nebel, 1998] B. Nebel. *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, chapter How hard is it to revise a knowledge base? Kluwer Academics, 1998.

[Poole, 1988] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.

[Rescher and Manor, 1970] N. Rescher and R. Manor. On inference from inconsistent premises. *Theory and Decision*, 1: 179–219, 1970.

[Restall, 1999] Greg Restall. *An Introduction to Substructural Logics*. Routledge, 1999.