Raisonnement en présence d'incohérence : le connecteur oublié

Reasoning under inconsistency: the forgotten connective

Sébastien Konieczny[1] Jérôme Lang[1] Pierre Marquis[2]

[1] Institut de Recherche en Informatique de Toulouse
Université Paul Sabatier
31062 Toulouse
{konieczny,lang}@irit.fr
[2] Centre de Recherche en Informatique de Lens
Université d'Artois
62300 Lens
marquis@cril.univ-artois.fr

Résumé

Dans la plupart des approches pour le raisonnement en présence d'incohérence, en particulier celles basées sur la sélection de sous-ensembles cohérents de formules, une hypothèse sous-jacente est que des formules différentes sont liées par un lien moins fort qu'une conjonction logique. Dans le cas où elle est cohérente, une base de croyances $B = \{\varphi_1, \ldots, \varphi_n\}$*, où les* φ_i *sont des formules propositionnelles, est équivalente à la base* $B' = \{\varphi_1 \wedge \ldots \wedge \varphi_n\}$*. En revanche, dans le cas où cette conjonction est incohérente, ces deux bases ont typiquement des ensembles de conséquences distincts. Cela illustre le fait que la virgule utilisée dans la base* B *est un connecteur particulier, différent de la conjonction. Nous proposons dans ce travail d'étudier une logique comportant un tel connecteur "virgule". Ce cadre permet de fournir une sémantique à ce connecteur et de généraliser un certain nombre d'approches permettant de raisonner en présence d'informations contradictoires.*

Mots Clef

Raisonnement en présence d'incohérence.

Abstract

In many frameworks for reasoning under inconsistency, in particular those based on the selection of consistent subsets of the given belief base, it is implicitely assumed that the formulas from the base are connected using a weak form of conjunction. When it is consistent, a belief base $B = \{\varphi_1, \ldots, \varphi_n\}$*, where the* φ_i *are propositional formulas, is logically equivalent to the base* $B' = \{\varphi_1 \wedge \ldots \wedge \varphi_n\}$*. However, when it is not consistent, both bases typically lead to different conclusions. This illustrates the fact that the comma used in base* B *has to be considered as an additional connective, and not as a simple conjunction. In this work we define and study a logic with such a "comma" connective. We give it a semantics and on this ground, we generalise several approaches for reasoning from inconsistent beliefs.*

Keywords

Inconsistency handling.

1 Introduction

Il y a plusieurs situations assez différentes où on peut être conduit à travailler avec des ensembles de formules incohérentes (voir par exemple [BH01] pour une classification fine de ces situations) ; par exemple :

- plusieurs experts ou capteurs qui fournissent des données conflictuelles ;
- plusieurs avis divergents concernant l'évaluation d'un objet ou d'un candidat ;
- plusieurs contextes ou plusieurs règles simultanément applicables dont les conclusions sont contradictoires ;
- les hypothèses de bon fonctionnement simultané de composants d'un système dont la conjonction est contradictoire ;
- plusieurs critères conduisant à des préférences incompatibles pour un agent donné ;
- plusieurs agents dont les préférences sont incompatibles.

Il faut bien noter que pour les quatre premiers items il s'agit de *raisonnement à partir de croyances (contradictoires)* tandis que pour les deux derniers items il s'agit de *décision à partir de préférences (conflictuelles)*.

Tout agent doté de capacités déductives suffisantes doit renoncer à exploiter sa base de croyances dès que celle-ci est incohérente : en effet, à partir d'une base de croyances incohérente, la relation de déduction classique permet d'inférer n'importe quelle formule du langage (principe connu sous le nom de *ex falso quodlibet sequitur*).

Quelle attitude raisonnable peut-on alors envisager face à une base de croyances incohérentes ? On peut élaborer une taxonomie de ces attitudes.

– *attitude active*. L'agent cherche à résoudre les incohérences, c'est-à-dire à les identifier (on pourrait presque dire "débugger la base de croyances"), en effectuant des actions de prise d'information (des tests) qui, idéalement, rendront *in fine* la base de croyances[1] cohérente. Voir par exemple [LM00a, LD00].
– *attitude passive*. L'agent n'aura pas de nouvelles informations lui permettant de l'aider à raisonner avec l'incohérence. Il doit se débrouiller tant bien que mal avec sa base de croyances, et rien d'autre.

Intéressons-nous à ce second type d'attitude : puisqu'il faut de toute façon s'arranger pour réduire l'ensemble des conséquences déduites des informations de la base (par rapport à ce que ferait la logique classique), il n'y a en définitive que deux solutions : soit on affaiblit les informations de la base de croyances de telle manière que leur conjonction devienne cohérente, soit on affaiblit la relation de conséquence logique[2].

Affaiblir la déduction conduit à la famille des *logiques paraconsistantes*. Affaiblir les informations conduit à plusieurs familles d'approches :

– *affaiblissement par inhibition.* C'est la méthode la plus ancienne. Elle consiste à calculer d'abord des sous-ensembles de la base de croyances B cohérents et maximaux pour un critère qui est à définir. Cette façon de procéder remonte à [RM70]. Le critère le plus naturel est celui de l'inclusion : une *sous-base maximale cohérente* est tout simplement une sous-base cohérente de B dont tout sur-ensemble strict dans B est incohérent. Mais bien d'autres critères sont pertinents, qui tiennent compte par exemple du *nombre* de formules satisfaites, d'une relation de priorité entre les formules ou d'une attribution de degrés d'importance à chaque formule. Par ailleurs, il ne suffit pas de se fixer un critère de sélection ; il faut aussi savoir ce que l'on fera des informations déduites à partir des différentes sous-bases sélectionnées. Pour cela, on introduit ce que Cayrol et Lagasquie [CLS94] appellent un *procédé d'inférence*. Le plus courant consiste à inférer une formule si et seulement si elle est déductible de *toute* sous-base préférée (inférence dite "universelle" ou "sceptique")[3].
– *affaiblissement par dilatation : fusion d'informations*. L'affaiblissement par inhibition consistait à affaiblir les informations de B en les transformant en défauts dont la vérité est plausible mais pas certaine : lorsqu'un défaut n'est pas satisfait, peu importe la façon dont il n'est pas satisfait (en d'autres termes, on ne fera pas de distinction entre un défaut "violé de peu" et un autre "violé de beaucoup"). Cette distinction est pourtant pertinente si l'on considère que l'ensemble des modèles peut être muni d'une structure de *pseudo-distance* : une information sera d'autant plus insatisfaite par un monde que la distance de ce monde au plus proche des modèles de cette information est grande. Ce principe est la base de la *fusion de croyances* à base de distances [Rev97, LS98, KP98, KP02, KLM02].
– *affaiblissement par oubli de variables.* l'idée sous-jacente est que l'incohérence d'une base de croyances peut être causée par un excès d'information sur certaines variables dont l'importance n'est peut-être pas cruciale. On peut alors se focaliser sur ces variables, simplifier les informations en oubliant tout ce qui concerne ces variables "responsables de l'incohérence" et parvenir ainsi à restaurer la cohérence [LM00b]. Oublier un ensemble de variables dans une formule permet ainsi de l'affaiblir tout en préservant un maximum de croyances sur les variables non oubliées. Comme pour l'inhibition de formules, le choix des sous-ensembles de variables à oublier peut être guidé par des priorités ou des pénalités sur les variables (traduisant leur degré de pertinence). On montre que l'affaiblissement par oubli de variables généralise à la fois les approches par inhibition et les approches par dilatation.

Dans ces trois familles d'approches (inhibition, dilatation, oubli de variables), on a en entrée du processus de raisonnement non pas une formule logique au sens classique du terme mais *un ensemble (fini) de formules* : en particulier, l'inférence est définie comme une relation *entre un ensemble (fini) de formules et une formule*. *A contrario*, pour les logiques paraconsistantes comme pour la logique classique, l'inférence peut être définie entre une formule et une autre.

Il faut ici éclaircir un point : la logique classique – et les logiques en général – considèrent *aussi* des ensembles de formules (et il existe d'ailleurs certains systèmes d'inférence, en particulier pour la logique classique, qui sont des systèmes de réécriture entre ensembles de formules – par exemple le calcul des séquents de Gentzen). Cependant ces ensembles de formules sont toujours interprétés conjonctivement. En particulier, les conséquences logiques d'un ensemble fini de formules Δ coïncident avec les conséquences logiques de la formule $\bigwedge \Delta$ qui consiste en la conjonction de toutes les formules de Δ.

Il n'en va pas de même pour les approches du traitement de l'incohérence par affaiblissement ! Ainsi, qu'il s'agisse d'approches par inhibition de formules, de fusion ou d'oubli de variables, l'ensemble $\{a, \neg a, b\}$ n'a typiquement pas le même comportement vis-à-vis de l'inférence que sa conjonction $\{a \wedge \neg a \wedge b\}$. Ainsi, l'approche par inférence "universelle" à partir des sous-ensembles cohérents maximaux pour l'inclusion permet d'inférer de $\{a, \neg a, b\}$ les conséquences logiques de b tandis que seules les tautolo-

[1] *Stricto sensu*, ce n'est pas la base de croyances initiale qui sera "rendue" cohérente, mais sa révision par les résultats des tests.

[2] On pourrait évidemment envisager de faire les deux à la fois.

[3] Des procédés plus faibles sont l'inférence "existentielle" (ou "crédule") qui consiste à inférer une formule si et seulement si elle est déductible d'au moins une sous-base préférée, et l'inférence "argumentative", qui consiste à inférer une formule si et seulement si elle est existentiellement inférable et sa négation ne l'est pas.

gies classiques peuvent en être inférées de $\{a \wedge \neg a \wedge b\}$.

Ce comportement a souvent été qualifié de “dépendance vis-à-vis de la syntaxe”, et les approches correspondantes (que ce soit pour l’inférence à partir de bases de croyances incohérentes, la révision des croyances ou le raisonnement par défaut), ont elles été qualifiées de “fondées sur la syntaxe” – en anglais “syntax-based”) [Neb91] ; elles ont parfois été critiquées pour cette même raison [Dal88, Win88] : le principe d’indépendance à la syntaxe énoncé dans [Dal88] stipule que “la révision d’un état de croyance doit être indépendant de la forme syntaxique de la base de croyances représentant l’état de croyance (...)”. Ce principe demande que le méta-théorème de substitution de la logique classique s’applique sans restriction. Ceci n’est pas toujours désirable, en particulier toutes les logiques paraconsistantes rejettent ce principe.

Selon nous, tant cette dénomination que le reproche fait aux approches qui en relèvent, reposent sur un malentendu. Pourquoi $\{\varphi, \psi\}$ devrait-elle être “équivalente” à $\{\varphi \wedge \psi\}$ pour l’inférence, sinon parce qu’on a fait l’hypothèse implicite que *les ensembles de formules doivent être identifiés à leur conjonction* ? La dénomination “dépendance vis-à-vis de la syntaxe” revient en effet à ne voir entre $\{\varphi, \psi\}$ et $\{\varphi \wedge \psi\}$, qu’une seule différence syntaxique (tout comme entre $\varphi \wedge \psi$ et $\psi \wedge \varphi$ en logique classique), ce qui signifie qu’elles doivent être tenues pour (sémantiquement) équivalentes. Mais, si l’on tient à cette équivalence, pourquoi définir une base de croyances comme un *ensemble* de formules plutôt que par une conjonction (rappelons qu’on est dans un cadre fini, ce qui rend la chose possible) ? De surcroît, il est surprenant de voir la “dépendance à la syntaxe” comme une source de maux : cette dépendance n’est-elle pas inhérente à tout formalisme logique ? Si $\psi \wedge \varphi$ n’est pas équivalent à $\psi \vee \varphi$ (en général), n’est-ce pas là aussi une affaire de syntaxe, entre autres ?

Allons un peu plus loin. Si les multiples approches fondées sur la sélection d’ensembles maximaux cohérents se sont montrées satisfaisantes, et si les relations d’inférence qu’elles permettent de définir diffèrent assurément de l’inférence classique, c’est justement parce qu’elles introduisent dans leur langage quelque chose qui n’existe pas en logique classique : les ensembles de formules avec une interprétation non purement conjonctive. Une fois que l’on a noté que $\{\varphi, \psi\}$ est une formule de ce langage, qui diffère de $\{\varphi \wedge \psi\}$, on doit justement s’attendre à ce que ces formules aient des comportements différents en ce qui concerne l’inférence.

Le langage des approches que nous venons de mentionner contient donc, en plus du langage propositionnel classique, des ensembles finis de formules, qui ne peuvent pas être réduits à de simples conjonctions. Cela revient à dire que le langage contient un connecteur supplémentaire : la virgule. La mal-nommée “dépendance vis-à-vis de la syntaxe” se reformule alors tout simplement par le fait que la virgule et la conjonction sont des connecteurs différents. La virgule n’est d’ailleurs pas définissable à partir des connecteurs “classiques”, comme on le verra plus loin – ce qui renforce son statut de connecteur à part entière.

Or, les nombreuses approches qui font appel à cette virgule ne lui ont pas reconnu ce statut de connecteur, et ont donc été hésitantes à franchir le pas de définir un langage dans lequel les objets manipulés (ensembles finis de formules) sont des formules au sens rigoureux du terme. C’est ce que nous nous proposons de faire ici.

Une fois reconnu à la virgule son statut de connecteur, il est alors très facile (sinon immédiat) de définir un langage bien plus général où la virgule s’imbrique librement avec elle-même et avec les autres connecteurs. Ce langage sera défini formellement au paragraphe 3.1 (après quelques préliminaires formels au paragraphe 2) ; nous lui associerons une sémantique (au paragraphe 3.2) inspirée directement des approches existantes sur le raisonnement à base de sous-ensembles maximaux cohérents : les modèles de l’ensemble $\{\varphi_1, \ldots, \varphi_n\}$ seront les interprétations (au sens classique) qui satisfont un sous-ensemble maximal cohérent (au sens de l’inclusion) de $\{\varphi_1, \ldots, \varphi_n\}$. On donnera au paragraphe 4 quelques tautologies de la logique ainsi définie, et on étudiera ses aspects calculatoires et en particulier sa complexité algorithmique au paragraphe 6.

Une telle extension ne constitue pas un simple exercice de style, où l’on se contenterait de définir un langage certes bien plus expressif que les langages existants mais manifestement dépourvu de tout intérêt. En effet, d’une part, la généralité et l’expressivité de ce nouveau langage permet de réexprimer d’une façon plus rigoureuse divers problèmes d’inférence à partir de bases de croyances incohérentes, de révision des croyances ou de raisonnement non monotone ; et d’autre part, il permet d’exprimer des informations qui ne peuvent pas l’être dans les langages qui ne reconnaissent pas à la virgule son statut de connecteur. Ces arguments sont développés au paragraphe 5.

Enfin, on remarquera que le choix que l’on aura fait au paragraphe 3.2 pour la sémantique des ensembles de formules n’est pas le seul possible : puisqu’il existe de nombreux critères pertinents pour la sélection de sous-ensembles maximaux cohérents, il existe également plusieurs interprétations pertinentes du connecteur virgule. Par souci de concision, les premiers paragraphes de l’article reposent sur le choix correspondant à la maximalité pour l’inclusion (qui est le plus fréquent et le plus naturel) ; d’autres interprétations pertinentes, faisant intervenir la cardinalité, des priorités ou des pondérations, seront plus brièvement évoquées au paragraphe 7, avant que de nouvelles perspectives ne soient ouvertes en conclusion.

2 Préliminaires

On considère un ensemble *fini* de symboles propositionnels PS. $\mathcal{L}_{PS}$ est le langage propositionnel construit à partir de PS, des constantes $\top$ (vrai) et $\bot$ (faux) et des connecteurs classiques $\{\wedge, \vee, \neg\}$ de la manière usuelle. Les formules de $\mathcal{L}_{PS}$ sont notées par des lettres grecques φ, ψ, etc.

Une interprétation est une fonction totale de PS dans

$\{0,1\}$, vue aussi comme l'ensemble de tous les symboles de PS qu'elle satisfait. $\mathcal{M}_{PS}$ dénote l'ensemble des interprétations pour PS. Une interprétation M est un modèle d'une formule ϕ, noté $M \models \phi$, si et seulement si elle la satisfait (i.e. la rend vraie). $Mod(\phi)$ dénote l'ensemble des modèles de la formule ϕ, i.e. $Mod(\phi) = \{M \in \mathcal{M}_{PS} \mid M \models \phi\}$.
$\subset$ dénote l'inclusion stricte. Soit $B = \{\varphi_1, \ldots, \varphi_n\}$ un ensemble fini de formules de $\mathcal{L}_{PS}$, et soit $B' \subseteq B$. On note $\bigwedge B' = \bigwedge\{\varphi_i \mid \varphi_i \in B'\}$. B' est un sous-ensemble *cohérent* de B si et seulement si $\bigwedge B'$ est cohérent. B' est un sous-ensemble *maximal cohérent* (ou *maxcons*) de B si et seulement si B' est cohérent et il n'existe pas de B'' cohérent tel que $B' \subset B'' \subseteq B$. On note $MaxCons(B)$ l'ensemble des sous-ensembles maximaux cohérents de B.
Les résultats de complexité que nous donnons dans cet article font mention de certaines classes de complexité de la hiérarchie polynomiale, ainsi que la classe PSPACE. Les résultats de complexité n'étant pas cruciaux dans cet article, nous renvoyons le lecteur à [Pap94] pour plus de détails sur ces définitions.

3 Un langage avec une virgule, et sa sémantique

3.1 Langage

Commençons par définir le langage de cette logique "à virgule".

Définition 1 *Le langage $\mathcal{L}_{PS}^{Virg}$ est défini inductivement comme suit :*
- $PS \cup \{\top, \bot\} \subseteq \mathcal{L}_{PS}^{Virg}$ *;*
- *si $\varphi \in \mathcal{L}_{PS}^{Virg}$ et $\psi \in \mathcal{L}_{PS}^{Virg}$ alors $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$ appartiennent à $\mathcal{L}_{PS}^{Virg}$;*
- *si $n \geq 0$ et $\{\varphi_1, \ldots, \varphi_n\} \subseteq \mathcal{L}_{PS}^{Virg}$ alors $\langle\varphi_1, \ldots, \varphi_n\rangle$ appartient à $\mathcal{L}_{PS}^{Virg}$.*

Clairement, le langage $\mathcal{L}_{PS}^{Virg}$ est une extension de $\mathcal{L}_{PS}$ dans laquelle on a donné à la virgule son statut de connecteur à part entière. Il faut bien noter l'importance des délimiteurs $\langle$ et $\rangle$ (qui ne doivent surtout pas être confondus avec des parenthèses !), qui est due au fait que, dans la sémantique que nous allons bientôt donner, le connecteur virgule n'est pas associatif : $\langle\varphi_1, \langle\varphi_2, \varphi_3\rangle\rangle$ ne sera en général pas équivalent à $\langle\langle\varphi_1, \varphi_2\rangle, \varphi_3\rangle$ ni à $\langle\varphi_1, \varphi_2, \varphi_3\rangle$; en conséquence, les formules du type $(\varphi_1, \ldots, \varphi_n)$ ne sont pas des formules de $\mathcal{L}_{PS}^{Virg}$: toute occurrence du connecteur virgule doit être située entre des délimiteurs $''\langle''$ et $''\rangle''$.
On définit la profondeur d'une formule de $\mathcal{L}_{PS}^{Virg}$, et les langages $\{\mathcal{L}_{PS}^{Virg[n]}, n \in \mathbb{N}\}$, comme suit :

Définition 2
- *si* $\varphi \in PS \cup \{\top, \bot\}$ *alors* $prof(\varphi) = 0$ *;*
- $prof(\neg\varphi) = prof(\varphi)$ *;*
- $prof(\varphi \wedge \psi) = prof(\varphi \vee \psi) = \max(prof(\varphi), prof(\psi))$ *;*
- $prof(\langle\varphi_1, \ldots, \varphi_n\rangle) = \max_{i=1\ldots n} prof(\varphi_i) + 1$.

Pour $n \in \mathbb{N}$, on définit le langage $\mathcal{L}_{PS}^{Virg[n]}$ comme l'ensemble de toutes les formules de $\mathcal{L}_{PS}^{Virg}$ de profondeur inférieure ou égale à n.

On vérifie aisément que cette définition inductive est bien fondée ; et qu'en outre, $\mathcal{L}_{PS}^{Virg} = \bigcup_{n\in\mathbb{N}} \mathcal{L}_{PS}^{Virg[n]}$ et $\mathcal{L}_{PS}^{Virg[0]} = \mathcal{L}_{PS}$.
Enfin, pour tout $n \geq 1$ on définit le langage $\mathcal{L}_{PS}^{Virg[s,n]}$ (ensemble des *formules simples de profondeur* n) comme le sous-ensemble de $\mathcal{L}_{PS}^{Virg[n]}$ composé des formules de la forme $\langle\varphi_1, \ldots, \varphi_k\rangle$, où $\varphi_1, \ldots, \varphi_k$ sont des formules de $\mathcal{L}_{PS}^{Virg[n-1]}$, c'est-à-dire l'ensemble des formules de profondeur n où la virgule est le connecteur principal.
Voici des exemples de formules de $\mathcal{L}_{PS}^{Virg}$, où $PS = \{a, b, c\}$:
- $\varphi_1 = a \wedge b \wedge (\neg b \vee c) \wedge \neg c$;
- $\varphi_2 = \langle a \wedge b, \neg b \vee c, \neg c\rangle$;
- $\varphi_3 = \neg c \wedge \langle a \wedge b, \neg b \vee c\rangle$;
- $\varphi_4 = \langle a, a \wedge b, \langle\neg b \vee c, \neg c\rangle\rangle$;
- $\varphi_5 = \langle a \wedge b \vee \langle c, a \vee b, \neg c\rangle, \langle\neg b \vee c, \neg c\rangle \wedge \langle\neg a, \langle c, a \wedge b\rangle\rangle\rangle \vee \langle b \wedge (\neg c \vee a), \langle b, c\rangle\rangle$;
- $\varphi_6 = \langle a, a \wedge b, \top, \bot\rangle$;
- $\varphi_7 = \langle a\rangle$;
- $\varphi_8 = \langle\ \rangle$.

On a $prof(\varphi_1) = 0$; $prof(\varphi_2) = prof(\varphi_3) = prof(\varphi_6) = prof(\varphi_7) = prof(\varphi_8) = 1$; $prof(\varphi_4) = 2$; $prof(\varphi_5) = 3$. Notons aussi que $\varphi_2 \in \mathcal{L}_{PS}^{Virg[s,1]}$ (de même que φ_6, φ_7 et φ_8) et que $\varphi_5 \in \mathcal{L}_{PS}^{Virg[s,3]}$.

3.2 Sémantique

Définition 3 *$\models_V$ (satisfaction d'une formule de $\mathcal{L}_{PS}^{Virg}$ par une interprétation) est la relation sur $\mathcal{M}_{PS} \times \mathcal{L}_{PS}^{Virg}$ définie inductivement par :*
- $M \models_V \top$ *;*
- $M \not\models_V \bot$ *;*
- *si $a \in PS$, $M \models_V a$ si et seulement si $a \in M$;*
- *$M \models_V \neg\varphi$ si et seulement si $M \not\models_V \varphi$;*
- *$M \models_V \varphi \wedge \psi$ si et seulement si $M \models_V \varphi$ et $M \models_V \psi$;*
- *$M \models_V \varphi \vee \psi$ si et seulement si $M \models_V \varphi$ ou $M \models_V \psi$;*
- *$M \models_V \langle\varphi_1, \ldots, \varphi_n\rangle$ si et seulement si il n'existe pas d'interprétation $M' \in \mathcal{M}_{PS}$ tel que $\{i \mid M' \models_V \varphi_i\}$ contienne strictement $\{i \mid M \models_V \varphi_i\}$.*

On introduit la notation suivante :

Définition 4 *Lorsque $\varphi = \langle\varphi_1, \ldots, \varphi_n\rangle$ et $M \in \mathcal{M}_{PS}$, on note $Filter(M, \varphi) = \{i \mid M \models_V \varphi_i\}$.*

La dernière condition de la définition 3 peut alors s'écrire : si $\varphi = \langle\varphi_1, \ldots, \varphi_n\rangle$ alors $M \models_V \varphi$ si et seulement si il n'existe pas de $M' \in \mathcal{M}_{PS}$ tel que $Filter(M', \varphi) \supset Filter(M, \varphi)$.
On note $Mod_V(\varphi) = \{M \in \mathcal{M}_{PS} \mid M \models_V \varphi\}$. Une formule de $\mathcal{L}_{PS}^{Virg}$ est *cohérente* (ou *satisfaisable*) si et seulement si $Mod_V(\varphi) \neq \emptyset$, et *valide* si et seulement si

$Mod_V(\varphi) = \mathcal{M}_{PS}$. Si φ et ψ sont deux formules de $\mathcal{L}_{PS}^{Virg}$, on dit que ψ est conséqence de φ (noté $\varphi \models_V \psi$) si et seulement si $Mod_V(\varphi) \subseteq Mod_V(\psi)$. Lorsque $\varphi = \top$, on note simplement $\models_V \psi$ au lieu de $\top \models_V \psi$. On dit que deux formules φ et ψ de $\mathcal{L}_{PS}^{Virg}$ sont *équivalentes*, noté $\varphi \equiv_V \psi$, si et seulement si on a $\varphi \models_V \psi$ et $\psi \models_V \varphi$.
On remarque que la restriction de $\models_V$ aux formules de profondeur 0 coïncide avec la relation de satisfaction classique $\models$.

4 Quelques propriétés

Nous allons dans ce paragraphe donner les principales propriétés logiques du connecteur virgule, et voir en particulier comment celui s'articule avec les autres connecteurs.
Il faut d'abord noter que, contrairement aux autres connecteur, le connecteur virgule n'est pas vérifonctionnel : la valeur de vérité de $\langle \varphi_1, \ldots, \varphi_n \rangle$ dans M ne peut être déterminée lorsqu'on connait seulement les valeurs de vérité de chaque φ_i ($i \in 1 \ldots n$) dans M.
Nous allons à présent donner quelques propriétés triviales, mais illustrant bien le comportement de cet opérateur. La définition 3 implique immédiatement les propriétés suivantes :

$$\text{Pour toute permutation } \sigma \text{ de } \{1, \ldots, n\}, \quad \langle \varphi_1, \ldots, \varphi_n \rangle \text{ est équivalente à } \langle \varphi_{\sigma(1)}, \ldots, \varphi_{\sigma(n)} \rangle \tag{1}$$

$$\forall k \text{ si } \forall i \in 1 \ldots k \ \varphi_i \equiv_V \varphi'_i, \text{ alors } \langle \varphi_1, \ldots, \varphi_k, \ldots, \varphi_n \rangle \equiv_V \langle \varphi'_1, \ldots, \varphi'_k, \ldots, \varphi_n \rangle \tag{2}$$

$$\langle \varphi_1, \ldots, \varphi_1, \varphi_2, \ldots, \varphi_n \rangle \equiv_V \langle \varphi_1, \ldots, \varphi_n \rangle \tag{3}$$

La propriété suivante exprime qu'une formule de $\mathcal{L}_{PS}^{Virg[s,n]}$ est toujours cohérente[4].

$$\forall n \geq 0 \ \langle \varphi_1, \ldots, \varphi_n \rangle \not\models_V \bot \tag{4}$$

Et plus précisément, lorsque $n = 1$:

$$\langle \varphi \rangle \text{ est équivalente à } \varphi \text{ si } \varphi \text{ est cohérente, et à } \top \text{sinon.} \tag{5}$$

On a en particulier $\langle\ \rangle \equiv_V \langle \bot \rangle \equiv_V \top$.
En outre, il n'existe pas en général de lien déductif entre une imbrication de virgules et leur version "dépliée" ; ainsi, en général, on a :

$$\langle \langle \varphi_1, \ldots, \varphi_n \rangle, \langle \psi_1, \ldots, \psi_m \rangle \rangle \not\models_V \langle \varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m \rangle \tag{6}$$

Par exemple, $\langle \langle a, b \rangle, \langle \neg a, c \rangle \rangle \not\models_V \langle a, \neg a, b, c \rangle$. Cette propriété est à mettre en rapport avec l'équation (14), puisque cette dernière est un cas particulier de la propriété ci-dessus où l'implication est vérifiée.

[4]Ce qui n'est évidemment pas le cas des formules de $\mathcal{L}_{PS}^{Virg}$.

En général, on a aussi :

$$\langle \varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m \rangle \not\models_V \langle \langle \varphi_1, \ldots, \varphi_n \rangle, \langle \psi_1, \ldots, \psi_m \rangle \rangle \tag{7}$$

Par exemple, $\langle a, \neg a, b, c \rangle \not\models_V \langle \langle a, b, \neg a \rangle, \langle \neg a, c \rangle \rangle$.
Indiquons à présent comment ce connecteur s'articule avec les autres. Un premier résultat montre que le connecteur virgule se situe en quelque sorte entre la conjonction et la disjonction.

$$\varphi_1 \wedge \ldots \wedge \varphi_n \models_V \langle \varphi_1, \ldots, \varphi_n \rangle \tag{8}$$

$$\text{Si } \varphi_1 \vee \ldots \vee \varphi_n \not\models_V \bot \text{ alors } \langle \varphi_1, \ldots, \varphi_n \rangle \models_V \varphi_1 \vee \ldots \vee \varphi_n \tag{9}$$

On peut donc considérer le connecteur virgule comme une conjonction "faible", dans le sens où celui-ci donne le même résultat que la conjonction lorsque celle-ci est cohérente, mais donne tout de même un résultat cohérent lorsque cette dernière est incohérente. De plus dans le cas où toutes les formules sont deux à deux contradictoires, la virgule donne le même résultat que la disjonction.
Il est également instructif de considérer ce que donne le connecteur virgule avec deux formules, pour expliquer ce comportement :

$$\langle \varphi, \psi \rangle \equiv_V \begin{cases} \varphi \wedge \psi & \text{si cohérent, sinon} \\ \varphi \vee \psi & \text{si cohérent, sinon} \\ \top & \end{cases} \tag{10}$$

Il n'est donc pas surprenant que le connecteur virgule s'articule très bien avec la conjonction.

$$\langle \varphi_1, \ldots, \varphi_n \rangle \wedge \varphi \models_V \langle \varphi_1 \wedge \varphi, \ldots, \varphi_n \wedge \varphi \rangle \tag{11}$$

En général, on a :

$$\langle \varphi_1 \wedge \varphi, \ldots, \varphi_n \wedge \varphi \rangle \not\models_V \langle \varphi_1, \ldots, \varphi_n \rangle \wedge \varphi \tag{12}$$

Par exemple, $\langle a \wedge \neg a, b \wedge \neg a \rangle \not\models_V \langle a, b \rangle \wedge \neg a$. De plus, on a le même résultat négatif en toute généralité lorsque $\langle \varphi_1, \ldots, \varphi_n \rangle \wedge \varphi$ est cohérent, comme le montre l'exemple $\langle a \wedge \neg b, \neg a \wedge \neg b, \neg a \wedge \neg b \rangle \not\models_V \langle a, \neg a \vee b, \neg a \rangle \wedge \neg b$.
On a aussi :

$$\langle \varphi_1, \ldots, \varphi_n \rangle \wedge \langle \psi_1, \ldots, \psi_m \rangle \models_V \langle \varphi_1 \wedge \psi_1, \ldots, \varphi_1 \wedge \psi_m, \ldots, \varphi_n \wedge \psi_1, \ldots, \varphi_n \wedge \psi_m \rangle \tag{13}$$

L'implication réciproque n'est pas vérifiée, puisque par exemple $\langle a \wedge \neg a \rangle \not\models_V \langle a \rangle \wedge \langle \neg a \rangle$.

$$\langle \varphi_1, \ldots, \varphi_n \rangle \wedge \langle \psi_1, \ldots, \psi_m \rangle \models_V \langle \varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m \rangle \tag{14}$$

Voyons à présent quelles sont les relations entre la virgule et la disjonction.

Si φ est cohérent, alors

$$\langle \varphi_1 \vee \varphi, \ldots, \varphi_n \vee \varphi \rangle \equiv_V (\varphi_1 \wedge \ldots \wedge \varphi_n) \vee \varphi \quad (15)$$

Ce qui donne donc :

$$\langle \varphi_1 \vee \varphi, \ldots, \varphi_n \vee \varphi \rangle \models_V \langle \varphi_1, \ldots, \varphi_n \rangle \vee \varphi \quad (16)$$

L'implication réciproque n'est pas vérifiée, puisque par exemple $\langle a, b, \neg b \rangle \vee \neg a \not\models_V \langle a \vee \neg a, b \vee \neg a, \neg b \vee \neg a \rangle$.

5 Expressivité

Pour illustrer l'utilité d'inclure un tel connecteur virgule dans le langage, nous allons montrer dans ce paragraphe un ensemble d'approches qui peuvent être capturées par ce moyen.

5.1 Inférence sceptique

Une des méthodes les plus communes pour le raisonnement à partir d' un ensemble de formules globalement incohérent et de sélectionner les sous-ensembles maximaux (pour l'inclusion ensembliste) cohérents (dits *maxcons*) de formules (voir [RM70] par exemple). L'inférence sceptique (ou inférence "universelle") est alors définie comme l'ensemble des inférences classiques communes à tous les maxcons, soit :

Définition 5

$$\Delta \vdash_\forall \varphi \textit{ ssi } \forall M \in MaxCons(\Delta), M \models \varphi$$

Cette inférence sceptique est directement capturée par le connecteur virgule :

Proposition 1

$$\{\varphi_1, \ldots, \varphi_n\} \vdash_\forall \varphi \textit{ ssi } \langle \varphi_1, \ldots, \varphi_n \rangle \models_V \varphi$$

5.2 Inférence crédule

A partir d'un ensemble de maxcons, on peut également définir l'inférence crédule (ou inférence "existentielle"), qui permet d'inférer une formule si au moins un maxcons l'infère classiquement.

Définition 6 *Soit un ensemble de formules* Δ,

$$\Delta \vdash_\exists \varphi \textit{ ssi } \exists M \in MaxCons(\Delta), M \models \varphi$$

Cette inférence peut également être traduite dans le cadre de la logique de la virgule :

Proposition 2

$$\{\varphi_1, \ldots, \varphi_n\} \vdash_\exists \varphi \textit{ ssi } \langle \varphi_1, \ldots, \varphi_n, \neg\varphi \rangle \wedge \varphi \not\models_V \bot$$

5.3 Défauts super-normaux

La logique des défauts super-normaux [Bre89], mais aussi les "approches syntaxiques" de [Neb91, Poo88] peuvent être expliquées de la façon suivante : soit un couple $\langle B, B_d \rangle$, où B et B_d sont deux ensembles finis de formules. B est l'ensemble des faits (ou contraintes dures), qui est toujours supposé cohérent. Et B_d est un ensemble de défauts, c'est-à-dire un ensemble de formules que l'on désire ajouter aux faits, si cela ne donne pas d'incohérence. B_d est souvent incohérent avec B.
Une extension est un sous-ensemble de formules de $B \cup B_d$ qui contient toutes les formules de B et un maximum de formules de B_d (pour l'inclusion ensembliste). L'ensemble de toutes les extensions est donc défini par
$Extens(\langle B, B_d \rangle) = \{E | E \subseteq B \cup B_d$ et $B \subseteq E$ et $E \not\models \bot$ et $\forall E'$ t.q. $E \subset E' \subseteq B \cup B_d, E' \models \bot\}$.
L'inférence à partir d'une telle structure $\langle B, B_d \rangle$ est définie par :

Définition 7

$$\langle B, B_d \rangle \vdash_D \varphi \textit{ ssi } \forall E \in Extens(\langle B, B_d \rangle), E \models \varphi$$

Cela se traduit également facilement dans la logique de la virgule (l'ensemble des faits est cohérent et peut donc être considéré conjonctivement, i.e. $B \equiv \bigwedge B$) :

Proposition 3

$$\langle B, \{\varphi_1, \ldots, \varphi_n\} \rangle \vdash_D \varphi \textit{ ssi } \langle \varphi_1 \wedge B, \ldots, \varphi_n \wedge B \rangle \models_V \varphi$$

Finalement, on peut noter que la circonscription peut être codée comme une inférence à partir de défauts super-normaux, elle est donc récupérée également comme cas particulier de cette logique à virgule.

5.4 Révision de croyances

L'opérateur de révision basique de Nebel [Neb91, Neb98, FUV83] se définit comme suit :

Définition 8 *Soit* $B \bot \varphi = \{M \mid M \subseteq B \textit{ et } M \not\models \varphi \textit{ et } \forall M' \textit{ t.q. } M \subset M' \subseteq B, M' \models \varphi\}$
La révision basique est alors définie par :

$$B \circ_N \varphi = \Big(\bigvee_{M \in B \bot \neg\varphi} M\Big) \wedge \varphi$$

On peut alors calculer le résultat de cette révision avec la logique de la virgule :

Proposition 4

$$\{\varphi_1, \ldots, \varphi_n\} \circ_N \varphi \models \mu \textit{ ssi } \langle \varphi_1 \wedge \varphi, \ldots, \varphi_n \wedge \varphi \rangle \models_V \mu$$

5.5 Fusion de croyances

Pour le moment, les approches que nous avons montré exprimables dans la logique de la virgule ne requièrent que des formules de profondeur 1. Si on se permet d'imbriquer les connecteurs virgules, on gagne en expressivité et cela nous permet de coder d'autres approches. Ainsi, la fusion de croyances est par exemple exprimable dans ce cadre. Supposons que nous désirions fusionner des bases de croyances (ou de préférences) en autorisant certaines d'entre elles à être incohérentes. Il est nécessaire dans ce cas, si on veut prendre en compte ces bases incohérentes, d'effectuer une première fusion intra-source pour extraire une information pertinente de ces bases avant de réaliser une véritable fusion inter-source. Soit un ensemble de bases $E = \{B_1, \ldots, B_n\}$ tel que pour tout $i \in 1 \ldots n$, on a $B_i = \{\varphi^i_1, \ldots, \varphi^i_{k_i}\}$. On peut alors définir la fusion de ces bases comme :

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

Cet opérateur n'est pas définissable par les opérateurs de fusion à deux étapes d'agrégation définis dans [KLM02], mais traite le même problème. Si on s'autorise d'autres sémantiques pour la virgule (voir le paragraphe 7), on peut alors capturer les opérateurs de fusion à deux étapes d'agrégation.

6 Complexité algorithmique

Commençons par un résultat immédiat mais intéressant pour la suite du paragraphe.

Proposition 5 *Pour toute formule φ de $\mathcal{L}^{Virg}_{PS}$, il existe une formule ψ de $\mathcal{L}^{Virg[0]}_{PS} = \mathcal{L}_{PS}$ qui lui est équivalente.*

La preuve de ce résultat tient en une ligne : ψ est la formule de $\mathcal{L}^{Virg[0]}_{PS}$ (unique à l'équivalence logique près) telle que $Mod(\psi) = Mod_V(\varphi)$.

Il n'est pas difficile d'exhiber un algorithme qui calcule une transformée $C(\varphi)$ d'une formule quelconque φ de $\mathcal{L}^{Virg}_{PS}$, i.e. une formule de $\mathcal{L}_{PS}$ équivalente à φ. Lorsque $\varphi \in \mathcal{L}^{Virg[s,1]}_{PS}$, c'est-à-dire $\varphi = \langle\varphi_1, \ldots, \varphi_n\rangle$, alors $C(\varphi) = \bigvee\{\bigwedge B' \mid B' \in MaxCons(\{\varphi_1, \ldots, \varphi_n\}\}$ convient. Par exemple, la transformée de $\langle a \wedge b, \neg b \vee c, \neg c\rangle$ ainsi définie est $((a \wedge b) \wedge (\neg b \vee c)) \vee ((a \wedge b) \wedge (\neg c)) \vee ((\neg b \vee c) \wedge (\neg c))$, qui est équivalente à $(a \wedge b) \vee (\neg b \wedge \neg c)$.

Pour obtenir une transformée d'une formule de $\mathcal{L}^{Virg[1]}_{PS}$, il suffit alors de commencer par calculer une transformée de chaque sous-formule de φ qui est dans $\mathcal{L}^{Virg[s,1]}_{PS}$, puis de remplacer dans φ toute occurrence d'une formule de $\mathcal{L}^{Virg[s,1]}_{PS}$ par la transformée calculée. Par exemple, une transformée de $\neg c \wedge \langle a \wedge b, \neg b \vee c, \neg c\rangle$ est $\neg c \wedge ((a \wedge b) \vee (\neg b \wedge \neg c))$, qui est équivalente à $\neg c \wedge (a \vee \neg b)$.

Soit maintenant une formule de $\mathcal{L}^{Virg}_{PS}$ de profondeur k. On commence par transformer comme précédemment toutes les sous-formules de φ qui sont dans $\mathcal{L}^{Virg[s,1]}_{PS}$: on obtient ainsi une formule de profondeur $n-1$, et on itère le processus jusqu'à obtention d'une formule de profondeur 0. Par exemple, soit $\varphi = \langle a, b, \langle\neg b \vee c, \neg c\rangle\rangle$; on commence par transformer $\langle\neg b \vee c, \neg c\rangle$ en $\neg b \wedge \neg c$: au bout d'une itération, on a donc la formule $\langle a, b, \neg b \wedge \neg c\rangle$, qu'une seconde itération transforme en $(a \wedge b) \vee (a \wedge \neg b \wedge \neg c)$, équivalente à $a \wedge (b \vee \neg c)$.

Une simple récurrence sur la profondeur des formules montre que toute formule φ de $\mathcal{L}^{Virg}_{PS}$ peut être réécrite de manière équivalente (pour $\models_V$) en une formule propositionnelle classique, que l'on notera $C(\varphi)$. Le problème est que la taille de $C(\varphi)$ est en général exponentiellement plus grande que la taille de φ, et qu'il n'est donc pas possible d'exploiter cette transformation pour dériver des résultats de complexité pour $\mathcal{L}^{Virg}_{PS}$. Plus exactement, il n'est pas possible d'associer à toute formule à virgule une formule classique logiquement équivalente et de taille polynomiale en la taille de la formule à virgule, sous les hypothèses usuelles de la théorie de la complexité[5].

Le problème de la satisfiabilité dans $\mathcal{L}^{Virg}_{PS}$ est évidemment au moins aussi difficile que pour la logique classique (il est donc NP-difficile). Que peut-on dire de plus ? Commençons par $\mathcal{L}^{Virg[1]}_{PS}$.

Proposition 6 *Le problème de satisfaisabilité dans $\mathcal{L}^{Virg[1]}_{PS}$ est Σ^p_2-complet.*

Preuve : on commence par montrer que le problème de vérification qu'une interprétation M satisfait une formule φ de $\mathcal{L}^{Virg[1]}_{PS}$ est dans Δ^p_2. En effet, soit $V_1(\varphi)$ l'ensemble des sous-formules de φ qui appartiennent à $\mathcal{L}^{Virg[s,1]}_{PS}$, c'est-à-dire de la forme $\langle\psi_1, \ldots, \psi_n\rangle$ où les ψ_i sont des formules de $\mathcal{L}_{PS}$. Posons $V_1(\varphi) = \{\alpha_1, \ldots, \alpha_q\}$. Décider si $M \models_V \alpha_i$ est dans coNP, puisqu'il suffit, pour montrer que $M \not\models_V \alpha_i$, de deviner une interprétation M' puis de vérifier que $Filter(M', \varphi) \supset Filter(M, \varphi)$. Or, vérifier que $M \models_V \varphi$ peut être résolu par l'algorithme non-déterministe suivant :

1. *pour* $i = 1$ jusqu'à q :
2. *si* $M \models_V \alpha_i$
3. *alors* remplacer α_i par $\top$ dans φ
4. *sinon* remplacer α_i par $\bot$ dans φ ;
5. *fin* (pour)
6. vérifier que la formule φ' obtenue après les q remplacements effectués est satisfaite par M.

Le test à l'étape 2 fait appel à un oracle NP, et les autres étapes sont résolues en temps polynomial. Par conséquent, vérifier que $M \models_V \varphi$ peut être résolu en temps polynomial à l'aide d'un nombre polynomial d'appels à l'oracle, c'est-à-dire qu'il est dans Δ^p_2. Maintenant, vérifier qu'une formule φ est satisfaisable consiste à deviner une interprétation M et à vérifier que $M \models_V \varphi$: le problème de satisfiabilité dans

[5] Ce résultat est une conséquence directe de la non-compilabilité de la circonscription [CDLS99].

$\mathcal{L}_{PS}^{Virg[1]}$ est donc dans $\mathsf{NP}^{\Delta_2^p} = \mathsf{NP}^{\mathsf{P}^{\mathsf{NP}}} = \mathsf{NP}^{\mathsf{NP}} = \Sigma_2^p$. Quant à la Σ_2^p-difficulté du problème de satisfaisabilité dans $\mathcal{L}_{PS}^{Virg[1]}$, c'est un corollaire de la Proposition 2 et de la Σ_2^p-difficulté de l'inférence crédule à partir de sous-bases maximales cohérentes [Neb98, CLS95].

$\diamond$

Qu'en est-il maintenant de la satisfaisabilité de formules avec imbrication de virgules ? Une application itérée de la preuve précédente nous conduit facilement au résultat suivant :

Proposition 7 *Le problème de satisfaisabilité dans $\mathcal{L}_{PS}^{Virg[n]}$ est dans Σ_{n+1}^p.*

Nous n'avons pas de preuve de Σ_{n+1}^p-difficulté pour ce problème (mais nous le conjecturons). Pour le cas général où la profondeur n'est plus bornée, on a le résultat suivant :

Proposition 8 *Le problème de satisfaisabilité dans $\mathcal{L}_{PS}^{Virg}$ est dans* PSPACE.

En effet, une conséquence immédiate de la Proposition 7 est qu'il existe une réduction polynomiale du problème de satisfaisabilité de $\mathcal{L}_{PS}^{Virg}$vers le problème QBF.
Là encore, nous n'avons qu'un résultat d'appartenance mais il nous manque la PSPACE-difficulté (que nous conjecturons).
Quelques commentaires sur ces résultats. Tout d'abord, le fait que la satisfaisabilité dans $\mathcal{L}_{PS}^{Virg[1]}$ soit au second niveau de la hiérarchie polynomiale et pas au-delà est plutôt une bonne nouvelle. En effet, on savait (depuis le paragraphe 5) que les problèmes d'inférence "universelle" ou "existentielle" peuvent être réduits en temps polynomial en des problèmes de satisfaisabilité ou de validité de formules de $\mathcal{L}_{PS}^{Virg[1]}$; la satisfaisabilité dans $\mathcal{L}_{PS}^{Virg[1]}$ ne peut donc pas être en-deçà du second niveau. Le fait qu'elle ne soit donc pas non plus au-delà montre que le gain d'expressivité du langage obtenu en passant de $\mathcal{L}_{PS}^{Virg[s,1]}$ à $\mathcal{L}_{PS}^{Virg[1]}$ ne s'accompagne pas d'un saut de complexité.
Que le niveau d'imbrication de la virgule s'accompagne (vraisemblablement) d'un saut de complexité n'est pas surprenant (on peut remarquer un phénomène similaire avec le niveau d'imbrication des conditionnels, voir notamment [EG93]). Cette difficulté (qui se confirmerait si nous avions établi les résultats de difficulté que nous avons conjecturés !) ne doit pas nous empêcher de rechercher des algorithmes pour déterminer si une formule de $\mathcal{L}_{PS}^{Virg}$ est satisfaisable ou non. On a indiqué au début de ce paragraphe que la transformation d'une formule de $\mathcal{L}_{PS}^{Virg}$ en une formule de $\mathcal{L}_{PS}$ était en général exponentielle : c'est en tout cas ce qui se passe dans le pire des cas. En pratique, il se peut que cette transformation puisse être effectuée, totalement ou du moins partiellement, en limitant l'explosion en espace. Par "effectuer cette transformation partiellement", nous voulons dire qu'il est possible de n'appliquer la transformation précédente que pour les niveaux les plus internes d'imbrication de la virgule (c'est-à-dire de s'arrêter avant les $prof(\varphi)$ itérations nécessaires).

7 Variations sur la sémantique de la virgule

En posant la définition 3, nous avons choisi d'interpréter la satisfaction de $\langle \varphi_1, \ldots, \varphi_n \rangle$ comme la satisfaction par M d'un sous-ensemble de $\{\varphi_1, \ldots, \varphi_n\}$ *maximal pour l'inclusion*. Nous allons maintenant plus brièvement proposer quelques autres définitions pertinentes et examiner leurs conséquences les plus immédiates.

7.1 Maximalité "non triviale" pour l'inclusion

Une particularité avec la sémantique de $\langle \varphi_1, \ldots, \varphi_n \rangle$ donnée au paragraphe 3.2, apparaît dans le cas dégénéré où toutes les formules φ_i sont incohérentes – ce qui signifie que $MaxCons(\{\varphi_1, \ldots, \varphi_n\})$ est vide, et par conséquent que toute interprétation M satisfait (trivialement) $\langle \varphi_1, \ldots, \varphi_n \rangle$. $\langle \varphi_1, \ldots, \varphi_n \rangle$ est donc dans ce cas équivalente à $\top$.
Une conséquence de cet état de fait est que $\langle \varphi \rangle$ n'est pas toujours équivalente à φ (cf. paragraphe 4) : plus précisément, $\langle \varphi \rangle$ est équivalente à φ si et seulement si φ est cohérente, et équivalente à $\top$ sinon. Ce n'est un problème que si l'on désire, pour être en accord avec l'intuition, que $\langle \varphi \rangle$ soit équivalente à φ. Dans ce cas, il est facile de modifier la définition de $\models_V$ pour garantir l'équivalence de $\langle \varphi \rangle$ et φ dans tous les cas de figure : il suffit de remplacer, dans la définition 3, l'item définissant $M \models_V \langle \varphi_1, \ldots, \varphi_n \rangle$ par celui-ci :

– $M \models_V^{NT} \langle \varphi_1, \ldots, \varphi_n \rangle$ si et seulement si $\{i \mid M \models_V \varphi_i\}$ est non vide et il n'existe pas d'interprétation $M' \in \mathcal{M}_{PS}$ tel que $\{i \mid M' \models_V \varphi_i\}$ contienne strictement $\{i \mid M \models_V \varphi_i\}$.

Avec cette nouvelle définition, la propriété (4) n'est évidemment plus vraie ; la plupart des autres propriétés restent inchangées, et certaines peuvent être renforcées : ainsi, dans (9), on n'a plus besoin de la condition "si $\varphi_1 \vee \ldots \vee \varphi_n \not\vdash \bot$". (10) devient

$$\langle \varphi, \psi \rangle = \begin{cases} \varphi \wedge \psi & \text{si cohérent} \\ \varphi \vee \psi & \text{sinon} \end{cases} \tag{17}$$

Les résultats de complexité restent inchangés.

7.2 Maximalité pour la cardinalité

On peut choisir, comme modèles de $\langle \varphi_1, \ldots, \varphi_n \rangle$, les interprétations qui maximisent le *nombre* de formules φ_i satisfaites. Ce qui revient à remplacer, dans la définition 3, l'item définissant $M \models_V \langle \varphi_1, \ldots, \varphi_n \rangle$ par celui-ci :

– $M \models_V^{C} \langle \varphi_1, \ldots, \varphi_n \rangle$ si et seulement si il n'existe pas d'interprétation $M' \in \mathcal{M}_{PS}$ tel que $|\{i \mid M' \models_V \varphi_i\}| > |\{i \mid M \models_V \varphi_i\}|$.

Les propriétés données au paragraphe 4 sont peu affectées par cette modification, mise à part l'idempotence : $\langle \varphi_1, \varphi_1, \varphi_2, \ldots, \varphi_n \rangle$ n'est généralement plus équivalente à $\langle \varphi_1, \varphi_2, \ldots, \varphi_n \rangle$. Par exemple, $\langle a, a, \neg a \rangle$ n'est pas équivalente à $\langle a, \neg a \rangle$.

Les résultats de complexité, quant à eux, sont profondément changés. Pour commencer, la satisfaisabilité d'une formule de $\mathcal{L}_{PS}^{Virg[1]}$ est dans $\Delta_2^{\mathsf{P}}(O(\log n))$: étant donnée une formule φ de $\mathcal{L}_{PS}^{Virg[1]}$, considérons comme dans la preuve de la Proposition 5, l'ensemble $V_1(\varphi) = \{\alpha_1, \ldots, \alpha_q\}$ des sous-formules de φ qui appartiennent à $\mathcal{L}_{PS}^{Virg[s,1]}$. Pour chacune de ces formules α_i, le nombre maximal k_i de sous-formules de α_i simultanément satisfaisables peut être calculé par dichotomie sur le nombre de sous-formules de α_i : le calcul de chaque k_i peut donc être réalisé en un nombre logarithmique (en la taille de α_i) d'appels à un oracle NP, et la suite des k_i, en un nombre polynomial (en la taille de φ) d'appels à un oracle NP. Une fois la suite des k_i calculée, vérifier que $M \models_V \varphi$ devient un problème polynomial : on remplace d'abord chaque sous-formule α_i par $\top$ si $Filter(M, \alpha_i) = k_i$, et par $\perp$ sinon, puis on vérifie que M satisfait la formule de $\mathcal{L}_{PS}$ qui en résulte. Le problème de satisfaisabilité dans $\mathcal{L}_{PS}^{Virg[1]}$ avec la sémantique définie par $\models_V^C$ est donc dans Δ_2^p (il est en fait facile, en utilisant un résultat de complexité de l'inférence "universelle" avec critère de cardinalité [CLSS98], de montrer qu'il est Δ_2^p-complet).

Or, ce procédé peut être itéré, ce qui permet de montrer que *le problème de satisfaisabilité dans $\mathcal{L}_{PS}^{Virg}$ avec la sémantique définie par $\models_V^C$ est Δ_2^p-complet*. Il faut noter que le saut de complexité (entre le premier niveau de la hiérarchie polynomiale et le second) mis en évidence par [CLSS98] lorsque l'on passe du critère de cardinalité au critère de l'inclusion pour la sélection de sous-ensembles maximaux cohérents, s'agrandit ici en un saut gigantesque (de Δ_2^p à PSPACE).

On peut par ailleurs montrer que, contrairement à ce que l'on a établi dans le cas de $\models_V$, il existe une traduction de toute formule de $\mathcal{L}_{PS}^{Virg}$ en une formule équivalente de $\mathcal{L}_{PS}$, *dont la taille est polynomialement bornée*, lorsqu'on choisit la sémantique correspondant à $\models_V^C$. La preuve est constructive et s'appuie sur une fonction deed transformation C similaire à celle donnée au paragraphe 6 : on remplace d'abord toutes les occurrences des formules $\alpha_i = \langle\varphi_1, \ldots, \varphi_{n_i}\rangle$ de $\mathcal{L}_{PS}^{Virg[s,1]}$ dans φ par les *formules de cardinalité* $[= k_i] : \langle\varphi_1, \ldots, \varphi_{n_i}\rangle$, qui signifie qu'exactement k_i formules parmi les n_i formules $\varphi_1, \ldots, \varphi_{n_i}$ doivent être satisfaites ; or, une telle formule de cardinalité peut être exprimée par une formule classique de $\mathcal{L}_{PS}$ qui lui est équivalente et dont la taille est de même grandeur, à une fonction polynomiale près [BLS94]. L'itération du processus conduit à une traduction polynomiale de $\mathcal{L}_{PS}^{Virg}$vers $\mathcal{L}_{PS}$qui préserve l'équivalence $\equiv_V^C$.

7.3 Seuil de satisfaction

On peut également renoncer à la maximalité (pour l'inclusion ou la cardinalité) du sous-ensemble de formules satisfaites par une interprétation lors de la définition de $M \models_V \langle\varphi_1, \ldots, \varphi_n\rangle$, en fixant un seuil ρ ($0 < \rho < 1$), et en posant la définition :

– $M \models_V^\rho \langle\varphi_1, \ldots, \varphi_n\rangle$ ssi $\frac{|\{i \mid M' \models_V^\rho \varphi_i\}|}{n} \geq \rho$

Une approche utilisant un tel seuil a été proposée pour le raisonnement en présence d'incohérence dans [CL00], et pour la fusion de croyances dans [EKM03, Eve03].

Avec une telle sémantique, on peut montrer facilement que la complexité de la satisfaisabilité décroît encore, pour devenir un problème "seulement" NP-complet.

8 Conclusion

On a posé dans cet article des jalons pour une logique qui considère la virgule comme un connecteur à part entière. Plus exactement, on a donné une définition *sémantique* d'une telle logique. On a donné quelques tautologies intéressantes de cette logique (et aussi quelques non-tautologies intéressantes !), on a montré qu'elle permet de réexprimer dans un langage unifié divers problèmes de raisonnement en présence d'incohérence, d'inférence non monotone, de révision ou de fusion des croyances. On a partiellement identifié la complexité de la satisfaisabilité dans cette logique.

On a aussi montré comment donner encore plus de généralité à notre approche, en faisant varier la "sémantique de la virgule". Par faute de temps et d'espace, nous n'avons pas recherché exhaustivement, parmi les propriétés considérées au paragraphe 4, quelles sont celles qui tiennent encore pour les différentes interprétations de la virgule. Nous n'avons pas non plus expliqué comment introduire la notion de priorité entre formules dans notre langage et dans la sémantique associée – mais ceci ne pose aucun problème particulier.

Remerciements

Nous remercions Marie-Christine Lagasquie pour une relecture attentive de cet article. Pierre Marquis remercie l'IUT de Lens, la Région Nord/Pas-de-Calais et les Communautés Européennes pour leur support.

Références

[BH01] I. Bloch and A. Hunter, editors. *Fusion : General Concepts and Characteristics*, volume 16 (10) of *International Journal of Intelligent Systems*. Wiley, 2001. Special Issue on Data and Knowledge Fusion.

[BLS94] B. Benhamou, L.Sais, and P. Siegel. Two proof procedures for a cardinality based language in propositional calculus. In *Proceedings of the 11^{th} Annual Symposium on Theoretical Aspects of Computer Science (STACS'94)*, pages 71–82. IRISA, 1994.

[Bre89] G. Brewka. Preferred subtheories : an extended logical framework for default reasoning. In *Proceedings of the 11^{th} International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1043–1048, Detroit (MI), 1989.

[CDLS99] M. Cadoli, F. M. Donini, P. Liberatore, and Marco Schaerf. The size of a revised knowledge base. *Artificial Intelligence*, 115(1) :25–64, 1999.

[CL00] C. Cayrol and M.C. Lagasquie-Schiex. Une approche décisionnelle du traitement de l'incohérence inspirée par le raisonnement non monotone. In *Actes du 12e congrès francophone de Reconnaissance des Formes et d'Intelligence Artificielle (RFIA'00)*, pages 277–285, 2000.

[CLS94] C. Cayrol and M.C. Lagasquie-Schiex. Classification de relations d'inference nonmonotone : la prudence et les propriétés de déduction. Technical report, Institut de Recherche en Informatique de Toulouse, 1994.

[CLS95] C. Cayrol and M.C. Lagasquie-Schiex. Nonmonotonic syntax-based entailment : a classification of consequence relations. In C. Froidevaux and J. Kohlas, editors, Proceedings of ECSQARU'95, pages 107–114. Springer Verlag, 1995.

[CLSS98] C. Cayrol, M. Lagasquie-Schiex, and Th. Schiex. Nonmonotonic reasoning : from complexity to algorithms. *Annals of Mathematics and Artificial Intelligence*, 22(3-4) :207–236, 1998.

[Dal88] M. Dalal. Investigations into a theory of knowledge base revision : preliminary report. In *Proceedings of the 7^{th} National Conference on Artificial Intelligence (AAAI'88)*, pages 475–479, 1988.

[EG93] T. Eiter and G. Gottlob. The complexity of nested counterfactuals and iterated knowledge base revisions. In *Proceedings of the 13^{th} International Joint Conference on Artificial Intelligence (IJCAI'93)*, pages 526–531, 1993.

[EKM03] P. Everaere, S. Konieczny, and P. Marquis. Opérateurs de fusion à quota. In *Actes des Journées Nationales sur les Modèles de Raisonnement (JNMR'03)*, pages 91–105, 2003.

[Eve03] P. Everaere. Manipulabilité des opérateurs de fusion de croyances. *DEA d'Informatique - Université d'Artois*, 2003.

[FUV83] R. Fagin, J. D. Ullman, and M. Y. Vardi. On the semantics of updates in databases. In *Proceedings of the 2nd ACM SIGACT-SIGMOD Symposium on the Principles of Database Systems*, pages 352–365, 1983.

[KLM02] S. Konieczny, J. Lang, and P. Marquis. Distance-based merging : a general framework and some complexity results. In *Proceedings of KR2002*, pages 97–108, 2002.

[KP98] S. Konieczny and R. Pino-Pérez. On the logic of merging. In *Proceedings of the 6^{th} International Conference on Knowledge Representation and Reasoning (KR'98)*, pages 488–498, 1998.

[KP02] S. Konieczny and R. Pino Pérez. Merging information under constraints : a logical framework. *Journal of Logic and Computation*, 12 :773–808, 2002.

[LD00] P. Liberatore and F. M. Donini. Verification programs for abduction. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI'00)*, pages 166–170, 2000.

[LM00a] J. Lang and P. Marquis. In search of the right extension. In *Proceedings of the 7^{th} International Conference on Knowledge Representation and Reasoning (KR'00)*, pages 625–636, Breckenridge (CO), 2000.

[LM00b] J. Lang and P. Marquis. Resolving inconsistencies by variable forgetting. In *Proceedings of KR2002*, pages 239–250, 2000.

[LS98] P. Liberatore and M. Schaerf. Arbitration (or how to merge knowledge bases). *IEEE Transactions on Knowledge and Data Engineering*, 10(1) :76–90, 1998.

[Neb91] B. Nebel. Belief revision and default reasoning : Syntax-based approaches. In *Proceedings of the 2^{nd} International Conference on Knowledge Representation and Reasoning (KR'91)*, pages 417–428, 1991.

[Neb98] B. Nebel. *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, chapter How hard is it to revise a knowledge base ? Kluwer Academics, 1998.

[Pap94] Ch. H. Papadimitriou. *Computational complexity*. Addison–Wesley, 1994.

[Poo88] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36 :27–47, 1988.

[Rev97] P.Z. Revesz. On the semantics of arbitration. *Int, Journal of Algebra and Computation*, pages 133–160, 1997.

[RM70] N. Rescher and R. Manor. On inference from inconsistent premises. *Theory and Decision*, 1 :179–219, 1970.

[Win88] M. Winslett. Reasoning about action using a possible models approach. In *Proceedings of the 7^{th} National Conference on Artificial Intelligence*, pages 89–93, St. Paul, 1988.