Premiers pas vers une résolution du problème de satisfaisabilité pour la logique monadique du premier ordre par une reformulation en logique des connaissances subjective mono-agent

Alexandre Niveau[1] **François Schwarzentruber**[2] **Bruno Zanuttini**[1]

[1] Normandie Univ. ; UNICAEN, ENSICAEN, CNRS ; GREYC, 14 000 Caen, France
[2] Univ. Rennes, CNRS, IRISA, France

alexandre.niveau@unicaen.fr
francois.schwarzentruber@ens-rennes.fr
bruno.zanuttini@unicaen.fr

Résumé

Nous nous intéressons au problème de la satisfaisabilité d'une formule en logique monadique du premier ordre. Cette logique constitue un fragment décidable de la logique du premier ordre, et il est bien connu que le problème de satisfaisabilité associé est NEXPTIME-complet. Nous proposons une nouvelle approche pour le résoudre, qui consiste à traduire la formule monadique du premier ordre vers une extension de la logique de la connaissance S5, puis à résoudre le problème de satisfaisabilité associé. L'intérêt de cette traduction est qu'elle permet de s'appuyer sur la littérature sur la compilation de connaissances pour S5, et d'ainsi tirer parti de représentations compactes sur lesquelles les opérations sont efficaces. L'évaluation expérimentale de cette nouvelle approche étant en cours, cet article ne peut rendre compte formellement de son efficacité en pratique, mais fournit plusieurs pistes selon lesquelles son utilité potentielle est manifeste.

Abstract

We investigate the problem of deciding the satisfiability of a formula in a decidable fragment of first-order logic, namely its restriction to *monadic* predicates. The satisfiability problem in monadic first-order logic is known to be NEXPTIME-complete. We propose a new approach to solve it, which relies on an original translation of the monadic first-order formula into one of an extension of the epistemic logic S5, then on solving the satisfiability problem for the latter. The interest of this translation is that it makes it possible to leverage existing knowledge compilation techniques for S5, making use of compact representations with efficient operations. While the experimental evaluation of this new approach is still ongoing, this paper lists several ways in which it could potentially be useful.

1 Introduction

La logique du premier ordre a une importance fondamentale en mathématiques et en informatique ; elle constitue le langage utilisé pour de nombreuses théories axiomatiques (théorie des nombres, théorie des ensembles), car elle dispose à la fois d'une expressivité appréciable tout en étant dotée de systèmes de déduction corrects et complets. C'est aussi le paradigme dans lequel sont construites les logiques de description, qui ont une grande importance en intelligence artificielle [1].

Bien que la logique du premier ordre soit indécidable, la conception d'algorithmes permettant de prouver la validité de formules fait l'objet de nombreux efforts dans plusieurs champs de recherche en mathématiques et en informatique, notamment pour la vérification formelle [8]. Il y a de nombreux prouveurs disponibles pour la logique du premier ordre, tels que Vampire [10], SPASS [18], ou encore iProver [9].

Dans cet article, nous nous intéressons à un fragment décidable de la logique du premier ordre, la logique *monadique* du premier ordre (MFO). Dans ce fragment, tous les prédicats sont unaires, et il n'y a ni symbole de fonction ni

symbole d'égalité [1]. Cette expressivité réduite, qui rend la logique décidable, n'empêche pas MFO d'avoir des applications en pratique, notamment dans la vérification formelle pour la sûreté de fonctionnement [4]. Plus récemment, elle a été utilisée pour faire du *model checking symbolique* pour la logique des annonces publiques arbitraires [7].

À notre connaissance, la résolution en pratique du problème de satisfiabilité dans MFO a été peu étudiée (nous n'avons trouvé qu'un article, qui propose un encodage vers SAT [16]). Bien que décidable, le problème reste NEXPTIME-complet, ce qui rend inenvisageable une résolution efficace dans tous les cas ; cependant, on peut constater en pratique que les prouveurs de premier ordre ont de bonnes performances sur ces formules, alors même qu'ils traitent une classe beaucoup plus difficile – ce qui nous mène à penser qu'il est possible de faire mieux en s'appuyant sur les spécificités du fragment. C'est dans cette optique que s'inscrit le présent article : il vise à ouvrir la voie vers des algorithmes de résolution du problème de satisfiabilité pour MFO qui soient plus efficaces en pratique, en utilisant une approche par *compilation de connaissances*.

La compilation de connaissances propose de traduire une formule d'entrée dans un langage pour lequel les opérations dont on a besoin sont efficaces. Cette approche a été employée avec succès pour la logique propositionnelle dans plusieurs domaines [14]. Elle n'est cependant pas spécialement adaptée si on veut se contenter de résoudre le problème classique de satisfiabilité : en pratique la construction de la forme compilée prend beaucoup plus de temps que la recherche d'une solution, d'autant que les solveurs SAT modernes sont très efficaces.

Cependant, dans le contexte de MFO, l'espace de recherche est beaucoup plus grand ; en particulier, les solutions sont de taille exponentielle. Il semble donc qu'une approche par compilation, où l'on manipule une représentation efficace des formules, puisse donner de meilleurs résultats. En outre, ce type d'approche a l'avantage d'être « réutilisable » pour d'autres problèmes que la satisfiabilité.

Au lieu de s'intéresser à des représentations efficaces *ad hoc* pour les formules MFO, cet article part de l'observation que les modèles (canoniques) de ces formules peuvent être encodés de manière simple comme des ensembles d'affectation propositionnelles, ce qui suggère une traduction possible de MFO vers la logique des connaissances S5. De fait, nous exhibons une traduction polynomiale de toute formule de MFO vers le langage $\mathsf{S5_F}$, que nous définissons comme la logique S5 augmentée d'un connecteur représentant l'opération d'*oubli de variable*. L'intérêt de cette reformulation est que S5 a, dans les dernières années, fait l'objet de travaux évaluant plusieurs langages de représentation sur le plan de la compilation de connaissances [3, 15], c'est-à-dire en termes de compacité et d'efficacité des opérations, notamment l'oubli. L'idée est donc d'utiliser $\mathsf{S5_F}$ comme un *langage intermédiaire* pour tester une approche de compilation pour MFO.

L'évaluation expérimentale de cette approche étant toujours en cours, cet article se contente de définir et prouver la correction de notre reformulation de MFO vers $\mathsf{S5_F}$, et de présenter les pistes que nous comptons explorer. Après des préliminaires introduisant nos notations pour MFO et S5 (partie 2), nous définissons $\mathsf{S5_F}$ en ajoutant à S5 le connecteur d'oubli (partie 3), puis nous montrons que pour MFO le problème de satisfaisabilité est équivalent au problème de satisfaisabilité dans la classe des structures canoniques (partie 4). Nous définissons ensuite l'encodage des modèles, c'est-à-dire la traduction d'une structure MFO en un ensemble d'affectations propositionnelles, et le décodage associé (partie 5). Nous pouvons alors introduire notre reformulation syntaxique d'une formule MFO vers une formule $\mathsf{S5_F}$, et prouver qu'elle préserve les modèles canoniques modulo encodage (partie 6). Pour finir, nous détaillons plusieurs pistes que nous envisageons de suivre afin d'exploiter la reformulation ainsi présentée pour la résolution du problème de satisfaisabilité de MFO (partie 7), avant de conclure.

2 Préliminaires

Premier ordre monadique Nous notons MFO (pour *Monadic First-Order logic*) le fragment monadique de la logique du premier ordre, c'est-à-dire le fragment restreint aux prédicats unaires, sans symbole de fonction, et sans égalité. Une *signature MFO* est un couple $\Sigma = \langle \mathcal{X}, \mathcal{P} \rangle$, où $\mathcal{X}$ est un ensemble fini de variables, et $\mathcal{P}$ est un ensemble fini de symboles de prédicats unaires. Une *formule MFO* est une formule du premier ordre sur une signature MFO Σ.

Exemple 1 *La formule* $\Phi^{ex} = \forall X \cdot \exists Y \cdot (\neg P(X) \vee P(Y)) \wedge (P(X) \vee \neg P(Y)) \wedge Q(Z)$ *est une formule MFO, sur la signature* $\Sigma^{ex} = \langle \mathcal{X}^{ex}, \mathcal{P}^{ex} \rangle$ *avec l'ensemble de variables* $\mathcal{X}^{ex} = \{X, Y, Z\}$ *et l'ensemble de prédicats* $\mathcal{P}^{ex} = \{P, Q\}$.

La grammaire définissant les formules MFO sur une signature MFO donnée $\Sigma = \langle \mathcal{X}, \mathcal{P} \rangle$ est

$$\Phi ::= P(X) \mid \neg\Phi \mid \Phi \vee \Phi \mid \Phi \wedge \Phi \mid \exists X \cdot \Phi \mid \forall X \cdot \Phi$$

où X est une variable dans $\mathcal{X}$, P un prédicat dans $\mathcal{P}$, et, pour les constructions $\exists X \cdot \Phi$ et $\forall X \cdot \Phi$, Φ ne contient ni $\exists X$, ni $\forall X$. On note $\mathrm{Free}(\Phi)$ l'ensemble des variables *libres* de Φ, c'est-à-dire celles qui apparaissent non quantifiées dans Φ. À titre d'illustration, dans l'exemple 1, on a $\mathrm{Free}(\Phi^{ex}) = \{Z\}$. Pour clarifier les exemples, on s'autorise à suivre les conventions habituelles de notation : $\Phi \rightarrow \Psi$ est un raccourci pour $\neg\Phi \vee \Psi$, et $\Phi \leftrightarrow \Psi$ pour

1. Dans cet article, nous n'autorisons pas l'égalité dans MFO. Nous pensons toutefois que notre approche pourrait être adaptée à la logique monadique du premier ordre avec symbole d'égalité (voir partie 7).

$(\Phi \to \Psi) \wedge (\Psi \to \Phi)$. Ainsi, la formule de notre exemple peut s'écrire $\Phi^{ex} = \forall X \cdot \exists Y \cdot (P(X) \leftrightarrow P(Y)) \wedge Q(Z)$.

La sémantique des formules MFO s'appuie sur la notion de *structure MFO*. Étant donnée une signature MFO $\Sigma = \langle \mathcal{X}, \mathcal{P} \rangle$, une *structure MFO sur* Σ est un triplet $S = \langle \mathcal{D}, I_{\mathcal{X}}, I_{\mathcal{P}} \rangle$, où $\mathcal{D}$ est un ensemble non vide d'éléments appelés *valeurs*, $I_{\mathcal{X}} : \mathcal{I} \to \mathcal{D}$ est une interprétation d'un sous-ensemble $\mathcal{I}$ de $\mathcal{X}$, et $I_{\mathcal{P}} : \mathcal{P} \to (\mathcal{D} \to \{\top, \bot\})$ est une interprétation des prédicats de Σ. Les éléments du domaine de $I_{\mathcal{X}}$ sont appelés les *variables interprétées* de S ; pour une structure S, on note $\mathcal{I}(S)$ l'ensemble de ses variables interprétées.

La satisfaction d'une formule MFO Φ est définie pour toute structure $S = \langle \mathcal{D}, I_{\mathcal{X}}, I_{\mathcal{P}} \rangle$ sur la même signature que Φ et vérifiant $\mathcal{I}(S) \supseteq \text{Free}(\Phi)$. On dit qu'une telle structure *satisfait* Φ, et on note $S \models \Phi$, si l'une des conditions suivantes est vérifiée :

— $\Phi = P(X)$ et $I_{\mathcal{P}}(P)(I_{\mathcal{X}}(X)) = \top$,
— $\Phi = \neg\Psi$ et $S \not\models \Psi$,
— $\Phi = \Psi \vee \Theta$ et ($S \models \Psi$ ou $S \models \Theta$),
— $\Phi = \Psi \wedge \Theta$ et ($S \models \Psi$ et $S \models \Theta$),
— $\Phi = \exists X \cdot \Psi$ et il existe $v \in \mathcal{D}$ vérifiant $S_{X \leftarrow v} \models \Psi$, où $S_{X \leftarrow v}$ est obtenue à partir de S en définissant $I_{\mathcal{X}}(X) = v$ (en remplaçant l'ancienne valeur, le cas échéant),
— $\Phi = \forall X \cdot \Psi$ et pour tout $v \in \mathcal{D}$ on a $S_{X \leftarrow v} \models \Psi$.

On note $Mods(\Phi)$ l'ensemble des structures MFO qui satisfont la formule Φ, que l'on appelle des *modèles* de Φ. Le problème de *satisfaisabilité* consiste à déterminer si une formule MFO donnée a au moins un modèle.

Remarquons que la notion de modèle d'une formule Φ requiert d'interpréter *au moins* toutes les variables libres de la formule. De façon générale, l'ensemble $Mods(\Phi)$ contient donc des structures dont l'ensemble de variables interprétées $\mathcal{I}(S)$ est exactement $\text{Free}(\Phi)$, mais également, pour toute telle structure, toutes celles qui interprètent des variables non libres de Φ, de façon arbitraire.

Exemple 2 (suite) *Soient* $\mathcal{D}^{ex} = \{v, w, u\}$, $\mathcal{I}^{ex} = \{X, Z\}$, *et les interprétations* $I^{ex}_{\mathcal{X}}$ *et* $I^{ex}_{\mathcal{P}}$ *définies par :*

$$I^{ex}_{\mathcal{X}}(X) = I^{ex}_{\mathcal{X}}(Z) = v,$$
$$I^{ex}_{\mathcal{P}}(P)(v) = I^{ex}_{\mathcal{P}}(Q)(v) = \top,$$
$$I^{ex}_{\mathcal{P}}(P)(w) = I^{ex}_{\mathcal{P}}(Q)(w) = \bot,$$
$$I^{ex}_{\mathcal{P}}(P)(u) = I^{ex}_{\mathcal{P}}(Q)(u) = \bot.$$

Alors $S^{ex} = \langle \mathcal{D}^{ex}, I^{ex}_{\mathcal{X}}, I^{ex}_{\mathcal{P}} \rangle$ *est une structure MFO sur la signature* Σ^{ex}, *dont l'ensemble des variables interprétées est* $\mathcal{I}^{ex}$. *On a par ailleurs* $S^{ex} \models \Phi^{ex}$ *; en particulier, en notant* $\Psi^{ex} = (P(X) \leftrightarrow P(Y)) \wedge Q(Z)$, *on a*

$$(S^{ex}_{X \leftarrow v})_{Y \leftarrow v} \models \Psi^{ex},$$
$$(S^{ex}_{X \leftarrow w})_{Y \leftarrow w} \models \Psi^{ex},$$
$$(S^{ex}_{X \leftarrow u})_{Y \leftarrow w} \models \Psi^{ex}.$$

Le résultat suivant est connu [2, 12, 13]. Intuitivement, le problème est dans NEXPTIME car on peut deviner une structure de taille au plus exponentielle et vérifier que c'est un modèle en temps polynomial (en sa taille et celle de la formule). Le fait que l'on puisse se limiter à une structure de taille exponentielle provient du fait qu'il est inutile d'envisager deux valeurs qui satisfont exactement les mêmes prédicats (voir la partie 4 sur les structures canoniques).

Théorème 3 *Le problème de satisfaisabilité pour MFO est NEXPTIME-complet.*

Logique des connaissances Le langage de la logique des connaissances S5 (mono-agent) est obtenu en ajoutant la modalité $\Box$ à la logique propositionnelle ; intuitivement, $\Box\varphi$ représente le fait que l'agent *sait* que la formule φ est vraie. La modalité duale $\Diamond$ est définie par $\Diamond\varphi = \neg\Box\neg\varphi$, et $\Diamond\varphi$ se lit « l'agent considère comme possible que φ soit vraie ».

Dans toute la suite, nous nous intéressons au fragment *subjectif* de S5, c'est-à-dire à l'ensemble des formules qui ne parlent que de la connaissance des agents, et non directement de l'état des variables. Techniquement, cela signifie que les variables propositionnelles n'apparaissent que dans la portée de la modalité $\Box$ (ou de la modalité duale $\Diamond$). Par ailleurs, étant donné que dans S5, $\Box\Box\varphi$ est équivalent à $\Box\varphi$, et $\Box\neg\Box\varphi$ à $\neg\Box\varphi$, nous n'autorisons pas l'imbrication de modalités. Par exemple, $\Gamma^{ex} = \Box(x \vee y) \wedge \Diamond(\neg x)$ est une formule subjective de S5.

Plus précisément, étant donné un ensemble V de variables propositionnelles, l'ensemble des formules de S5 que nous considérons dans cet article est défini par la production Γ dans la grammaire suivante, où $x \in V$:

$$\varphi ::= x \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi$$
$$\Gamma ::= \Box\varphi \mid \Diamond\varphi \mid \neg\Gamma \mid \Gamma \vee \Gamma \mid \Gamma \wedge \Gamma.$$

De façon générale, la sémantique de S5 s'appuie sur la notion de *structure de Kripke pointée*. Toutefois, étant donné que nous considérons le fragment mono-agent et subjectif de S5, nous pouvons utiliser une sémantique plus simple [2]. Pour cela, étant donné un ensemble de variables propositionnelles V, on appelle *état de connaissance sur* V un ensemble K, non vide, d'affectations de ces variables ($\varnothing \subsetneq K \subseteq 2^V$). Intuitivement, un état de connaissance d'un agent (sur l'état de l'environnement) est l'ensemble des états dans lesquels, selon lui, l'environnement peut se trouver.

Étant donnée une formule Γ de S5 et un état de connaissance K sur les mêmes variables, on dit que K *satisfait* Γ (ou est un *modèle* de Γ), et on écrit $K \models \Gamma$, si l'une des conditions suivantes est vérifiée, où pour une affectation m

2. Il s'agit simplement de constater que la satisfaction d'une formule S5 subjective ne dépend que de la classe d'équivalence du monde pointé.

et une formule propositionnelle φ, la notation $m \models \varphi$ signifie que m satisfait φ selon la sémantique usuelle :

— $\Gamma = \Box\varphi$ et pour tout $m \in K$, on a $m \models \varphi$,
— $\Gamma = \Diamond\varphi$ et il existe $m \in K$ tel que $m \models \varphi$,
— $\Gamma = \neg\Delta$ et $K \not\models \Delta$,
— $\Gamma = \Delta \vee \Lambda$ et ($K \models \Delta$ ou $K \models \Lambda$),
— $\Gamma = \Delta \wedge \Lambda$ et ($K \models \Delta$ et $K \models \Lambda$).

Comme pour MFO, on note $Mods(\Gamma)$ l'ensemble des états de connaissance qui satisfont la formule Γ, que l'on appelle des *modèles* de Γ. Par exemple, $K^{ex} = \{xy, \bar{x}y, x\bar{y}\}$ est un état de connaissance qui satisfait la formule Γ^{ex} précédente : toutes les affectations satisfont bien $x \vee y$, et il en existe une qui ne satisfait pas x (à savoir $\bar{x}y$). En revanche, l'état de connaissance $\{xy, \bar{x}\bar{y}\}$ ne la satisfait pas, car la deuxième affectation contredit l'atome $\Box(x \vee y)$, et l'état de connaissance $\{xy, x\bar{y}\}$ ne la satisfait pas non plus, car il ne contient pas de support pour l'atome $\Diamond(\neg x)$.

3 Oubli

La reformulation de MFO dans S5 que nous proposons dans cet article utilise la notion d'*oubli* de variables pour S5 [19]. Intuitivement, oublier une variable propositionnelle x dans une formule Γ de S5 consiste à calculer une formule de S5 qui ne mentionne pas x, mais qui a les mêmes conséquences que Γ sur le langage restreint aux variables propositionnelles restantes.

Nous utilisons la définition suivante ; pour un état de connaissance K sur un ensemble de variables V et un sous-ensemble V' de V, la *projection* de K sur V', notée $K|_{V'}$, est définie par $K|_{V'} = \{m|_{V'} \mid m \in K\}$, où $m|_{V'}$ est la restriction de l'affectation propositionnelle m aux variables de V'.

Définition 4 (oubli dans un état de connaissance) *Soit V un ensemble de variables propositionnelles, et soit K un état de connaissance sur V. Pour une variable $x \in V$, on note* $\mathrm{Forget}(x, K)$ *l'ensemble des états de connaissance K' sur V qui vérifient $K'|_{V\setminus\{x\}} = K|_{V\setminus\{x\}}$.*

Nous insistons sur le fait que nous définissons les états de connaissance de l'ensemble $\mathrm{Forget}(x, K)$ comme des états de connaissance sur l'ensemble V initial (incluant donc la variable oubliée x). Ceci explique le fait que l'oubli d'une variable dans un état de connaissance ne donne pas un état de connaissance unique. À titre d'exemple, $\{\bar{x}\bar{y}z, xy\bar{z}\}$ et $\{\bar{x}\bar{y}z, \bar{x}y\bar{z}, xy\bar{z}\}$ (sur les variables x, y, z) sont dans $\mathrm{Forget}(x, \{x\bar{y}z, \bar{x}y\bar{z}, xy\bar{z}\})$, car les trois états de connaissance résultent en $\{\bar{y}z, y\bar{z}\}$ lorsque l'on oublie x. Ce choix simplifiera la définition de notre reformulation de MFO vers S5.

Le connecteur d'oubli L'oubli d'une variable est souvent vu comme une *opération* sur les formules [3, 19], et non comme un constructeur syntaxique. Afin de présenter notre construction sous une forme la plus générale possible, indépendamment des approches algorithmiques qu'elle permet d'envisager pour le problème de satisfaisabilité de MFO, nous adjoignons dans cet article, à la syntaxe de S5, un constructeur représentant l'oubli. Plus formellement, nous définissons le langage de formules $\mathsf{S5_F}$ par la production Γ dans la grammaire suivante, où $x \in V$:

$$\varphi ::= x \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi$$
$$\Gamma ::= \Box\varphi \mid \Diamond\varphi \mid \neg\Gamma \mid \Gamma \vee \Gamma \mid \Gamma \wedge \Gamma \mid \mathsf{F}x \cdot \Gamma.$$

La sémantique de $\mathsf{S5_F}$ est celle de S5, avec la règle additionnelle suivante :

— $K \models \mathsf{F}x \cdot \Gamma$ s'il existe un modèle K' de Γ tel que l'on ait $K \in \mathrm{Forget}(x, K')$.

Comme pour MFO, on s'autorise à réécrire les formules en utilisant les connecteurs $\rightarrow$ et $\leftrightarrow$ pour des raisons de lisibilité, sans considérer qu'ils font partie de la grammaire.

Le langage $\mathsf{S5_F}$ ainsi défini peut paraître artificiel, mais il nous semble intéressant de l'introduire pour au moins deux raisons. Tout d'abord, la reformulation que nous proposons par la suite met en évidence la difficulté du problème de satisfaisabilité pour ce langage. En effet, le problème de la satisfaisabilité dans S5 est NP-complet, tandis que nos résultats fournissent une réduction du problème de satisfaisabilité pour MFO, qui est NEXPTIME-complet, au problème de la satisfaisabilité dans $\mathsf{S5_F}$. Il est facile de voir que ce dernier est dans NEXPTIME, et il est donc NEXPTIME-complet. Il deviendrait dès lors intéressant d'étudier le problème de satisfaisabilité pour $\mathsf{S5_F}$ en pratique, notamment en essayant d'étendre des approches dédiées à la satisfaisabilité de S5 [6, 11].

D'autre part, l'oubli apparaît naturellement dans la progression des connaissances d'un agent à des étapes de temps discrètes, lorsque l'état du monde peut changer. Par exemple, considérons un agent qui sait initialement que x est vraie ou y est vraie ($\Box(x \vee y)$). Supposons qu'il effectue une action qui, si x est vraie, met y à la même valeur, et sinon laisse les variables inchangées. On peut construire de façon syntaxique une formule caractérisant ses connaissances dans l'état résultant, en dupliquant les variables pour chaque pas de temps : en représentant le temps en indice, les connaissances de l'agent, après exécution de l'action, deviennent alors $\Gamma_{01} \stackrel{\text{def}}{=} \Box(x_0 \vee y_0) \wedge \Box(x_0 \rightarrow (y_1 \leftrightarrow x_0 \wedge x_1 \leftrightarrow x_0)) \wedge \Box(\neg x_0 \rightarrow (x_1 \leftrightarrow x_0 \wedge y_1 \leftrightarrow y_0))$.

Si l'environnement possède une dynamique markovienne, il est suffisant pour l'agent de raisonner sur les valeurs des variables au nouvel instant (sur un état de connaissances *instantané*). Par exemple, dans un algorithme de planification, on peut couper l'arbre de recherche dès lors que l'on retombe sur un état de connaissances instantané déjà visité. Dans S5, cela nécessite l'*opération* d'oubli des variables x_0, y_0, tandis que dans $\mathsf{S5_F}$, l'état de connaissances instantané peut être représenté syntaxiquement, en l'occurrence par la formule $\mathsf{F}x_0 \cdot \mathsf{F}y_0 \cdot \Gamma_{01}$ (qui se trouve bien

être logiquement équivalente, comme attendu, à la formule $\Box(y_1)$).

4 Structures MFO canoniques

Nous nous intéressons dans cet article au problème de la *satisfaisabilité* de MFO. Nous commençons par simplifier le problème, en montrant qu'il est équivalent au problème de la satisfaisabilité par une structure MFO *canonique*. Intuitivement, une structure MFO est canonique lorsque les valeurs de son domaine ont toutes une interprétation distincte pour les prédicats. Une autre façon de le voir est de considérer chaque valeur du domaine comme le sous-ensemble de prédicats pour lesquels elle est interprétée à $\top$: puisque les formules MFO n'utilisent pas de symbole d'égalité, deux valeurs du domaine qui correspondent au même sous-ensemble sont redondantes, au sens où les identifier ne change pas l'ensemble des formules satisfaites par la structure.

Définition 5 (structure canonique) *Soit* $S = \langle \mathcal{D}, I_\mathcal{X}, I_\mathcal{P} \rangle$ *une structure MFO sur une signature* $\Sigma = \langle \mathcal{X}, \mathcal{P} \rangle$. *Notons* $\text{truePred} \colon \mathcal{D} \to 2^\mathcal{P}$ *la fonction définie par* $\text{truePred}(v) = \{P \in \mathcal{P} \mid I_\mathcal{P}(P)(v) = \top\}$. *La* structure canonique *associée à* S, *notée* $S^\downarrow$, *est la structure MFO sur la même signature* Σ *définie par* $S^\downarrow = \langle \mathcal{D}^\downarrow, I_\mathcal{X}{}^\downarrow, I_\mathcal{P}{}^\downarrow \rangle$, *où*

— $\mathcal{D}^\downarrow = \{\text{truePred}(v) \mid v \in \mathcal{D}\}$,

— $I_\mathcal{X}{}^\downarrow = \text{truePred} \circ I_\mathcal{X}$,

— $\forall P \in \mathcal{P}, \forall v^\downarrow \in \mathcal{D}^\downarrow \colon I_\mathcal{P}{}^\downarrow(P)(v^\downarrow) = \top \iff P \in v^\downarrow$.

Une structure MFO S *est dite* canonique *si elle vérifie* $S = S^\downarrow$.

Par la suite, nous noterons v_E pour désigner la valeur du domaine d'une structure canonique qui satisfait exactement les prédicats de $E \subseteq \mathcal{P}$. Par exemple, v_{PR} désigne la valeur qui satisfait les prédicats P et R, mais aucun autre prédicat.

Exemple 6 (suite) *La structure* S^{ex} *de l'exemple 2 contient trois valeurs, d'une part w et u, qui jouent le même rôle vis-à-vis des prédicats (ils n'en satisfont aucun), et la valeur v (qui satisfait les deux prédicats P, Q). La structure canonique associée à* S^{ex} *est ainsi une structure ne contenant que deux valeurs, l'une satisfaisant les deux prédicats et l'autre n'en satisfaisant aucun. Plus précisément, la structure* $(S^{ex})^\downarrow$ *est la structure ayant pour domaine* $\mathcal{D} = \{v_{PQ}, v_\varnothing\}$, *pour interprétation des variables* $I_\mathcal{X}(X) = I_\mathcal{X}(Z) = v_{PQ}$, *et pour interprétation des prédicats* $I_\mathcal{P}(P)(v_{PQ}) = I_\mathcal{P}(Q)(v_{PQ}) = \top$ *et* $I_\mathcal{P}(P)(v_\varnothing) = I_\mathcal{P}(Q)(v_\varnothing) = \bot$.

Étant donné que la satisfaction d'une formule MFO ne dépend que de la satisfaction des prédicats par les valeurs, et non des valeurs elles-mêmes, la proposition suivante est évidente.

Proposition 7 *Soient* Φ *une formule MFO, et* S *une structure MFO sur la même signature. Alors* S *satisfait* Φ *si et seulement si* $S^\downarrow$ *satisfait* Φ.

Dans la suite, nous donnons une reformulation du problème de satisfaisabilité pour MFO *ainsi restreint aux structures canoniques* en un problème de satisfaisabilité pour $\mathsf{S5_F}$.

5 Reformulation : sémantique

Nous sommes maintenant à même de proposer une reformulation du problème de satisfaisabilité pour MFO en un problème de satisfaisabilité pour $\mathsf{S5_F}$. Plus généralement, étant donnée une formule MFO Φ, nous exhibons une formule, notée $\text{Enc}(\Phi)$, de $\mathsf{S5_F}$ telle que les modèles (canoniques) de Φ peuvent être obtenus en « décodant » les modèles de $\text{Enc}(\Phi)$.

Nous commençons par définir les opérations d'encodage (de MFO vers $\mathsf{S5_F}$) et de décodage (de $\mathsf{S5_F}$ vers MFO) au niveau sémantique. Pour l'encodage, étant donnée une structure MFO canonique S sur une signature Σ, l'idée essentielle consiste à représenter chaque prédicat P de Σ par une variable propositionnelle x_P, puis chaque valeur v du domaine de S par une affectation m_v qui satisfait exactement les x_P pour lesquels on a $P(v) = \top$. Par ailleurs, on introduit une variable propositionnelle x_X pour chaque variable X de Σ, afin d'encoder l'inteprétation des variables. On note alors $V(\mathcal{X}) = \{x_X \mid X \in \mathcal{X}\}$, $V(\mathcal{P}) = \{x_P \mid P \in \mathcal{P}\}$, et $V(\Sigma) = V(\mathcal{X}) \cup V(\mathcal{P})$.

Définition 8 (encodage des valeurs) *Soit* $S = \langle \mathcal{D}, I_\mathcal{X}, I_\mathcal{P} \rangle$ *une structure MFO sur une signature* $\Sigma = \langle \mathcal{X}, \mathcal{P} \rangle$. *Pour toute valeur* $v \in \mathcal{D}$ *dans le domaine de* S, *on appelle* encodage *de* v *(pour* S*) l'affectation propositionnelle* m_v^S *à* $V(\Sigma)$ *satisfaisant*

$$\text{pour tout } P \in \mathcal{P} \colon m_v^S \models x_P \iff I_\mathcal{P}(P)(v) = \top$$
$$\text{pour tout } X \in \mathcal{X} \colon m_v^S \models x_X \iff I_\mathcal{X}(X) = v.$$

Définition 9 (encodage) *Soit* $S = \langle \mathcal{D}, I_\mathcal{X}, I_\mathcal{P} \rangle$ *une structure MFO sur une signature* $\Sigma = \langle \mathcal{X}, \mathcal{P} \rangle$. *L'*encodage *de* S, *noté* $\text{Enc}(S)$, *est l'état de connaissance sur* $V(\Sigma)$ *défini par* $\text{Enc}(S) = \{m_v^S \mid v \in \mathcal{D}\}$.

Lorsqu'il n'y aura pas d'ambiguïté, nous noterons m_v plutôt que m_v^S.

Exemple 10 (suite) *Nous reprenons la structure* S^{ex} *de l'exemple 2. L'ensemble des variables propositionnelles* $V(\Sigma^{ex})$ *est* $\{x_X, x_Y, x_Z, x_P, x_Q\}$. *Si l'on considère la valeur v, qui satisfait les deux prédicats et par laquelle X et Z sont interprétées, on obtient donc l'encodage* $m_v = x_X \bar{x_Y} x_Z x_P x_Q$. *Quant aux valeurs w et u, elles*

sont encodées en $m_w = m_u = x_X\bar{x}_Y\bar{x}_Z\bar{x}_P\bar{x}_Q$. *L'encodage de la structure* S^{ex} *est donc l'état de connaissance* $\{x_X\bar{x}_Y x_Z x_P x_Q, \bar{x}_X\bar{x}_Y\bar{x}_Z\bar{x}_P\bar{x}_Q\}$.

Considérons maintenant la structure $(S^{ex})^{\downarrow}$ *explicitée dans l'exemple 6. L'encodage de cette structure porte par définition sur le même ensemble de variables propositionnelles que celui de* S^{ex}, *puisqu'elles sont sur la même signature. Si l'on considère la valeur* v_{PQ}, *on obtient l'encodage* $m_{v_{PQ}} = x_X\bar{x}_Y x_Z x_P x_Q$, *et si l'on considère la valeur* $v_\varnothing$, *on obtient l'encodage* $m_{v_\varnothing} = \bar{x}_X\bar{x}_Y\bar{x}_Z\bar{x}_P\bar{x}_Q$. *L'encodage de la structure* $(S^{ex})^{\downarrow}$ *est donc le même état de connaissance que celui de la structure* S^{ex}, *ce qui, comme nous allons le voir un peu plus loin, n'est pas une coïncidence.*

Nous montrons à présent comment on peut *décoder* un état de connaissance, c'est-à-dire comment, étant donné un état de connaissance K, on peut retrouver (si elle existe) la structure canonique MFO S qui vérifie $\mathrm{Enc}(S) = K$.

Pour des raisons techniques liées à l'encodage syntaxique proposé dans la partie 6, nous définissons toutefois une fonction de décodage plus générale. En effet, l'encodage de la définition 9 a la propriété de générer des états de connaissance dans lesquels chaque valeur v du domaine de S est représentée par exactement une affectation m_v ; cette unique affectation propositionnelle renseigne par ailleurs sur les variables que S interprète par v. La fonction de décodage que nous considérons permet de décoder des états de connaissance plus généraux, dans lesquels plusieurs affectations peuvent représenter la même valeur.

Exemple 11 (suite) *Considérons l'état de connaissance* $\{x_X\bar{x}_Y\bar{x}_Z x_P x_Q, \bar{x}_X\bar{x}_Y x_Z x_P x_Q, \bar{x}_X\bar{x}_Y\bar{x}_Z\bar{x}_P\bar{x}_Q\}$. *On remarque qu'il ne peut pas être obtenu par l'encodage d'une structure canonique via la définition 9, car les deux premières affectations représentent la même valeur : la première (resp. la seconde) encode le fait que X (resp. Z) est interprétée par cette valeur ; à cette exception près, cet état de connaissance est le même que celui encodant la structure* S^{ex} *de l'exemple 10. La fonction que nous proposons permet tout de même de le décoder, et retrouvera une structure dans laquelle X et Z seront interprétées par la même valeur ; en l'occurrence, la structure obtenue sera la structure* $(S^{ex})^{\downarrow}$.

On doit toutefois se restreindre aux états de connaissance K qui permettent de décoder les interprétations des variables de façon non ambiguë. Pour cela, nous demandons à K de ne pas contenir deux affectations qui encoderaient des interprétations différentes d'une même variable. Par ailleurs, on note qu'un état de connaissance non ambigu n'encode en général l'interprétation que d'un sous-ensemble des variables de la signature sous-jacente. Tout ceci motive les définitions suivantes.

Définition 12 (décodabilité) *Soit* $\Sigma = \langle \mathcal{X}, \mathcal{P}\rangle$ *une signature MFO. Un état de connaissances K sur* $V(\Sigma)$ *est dit* décodable *(pour* Σ*) s'il vérifie*

$$\text{pour tout } X \in \mathcal{X},\ \text{pour tout } m, m' \in K\text{:}\quad m \models x_X \text{ et } m' \models x_X \implies m|_{V(\mathcal{P})} = m'|_{V(\mathcal{P})}.$$

On note D_Σ, *ou simplement D, l'ensemble des états de connaissance qui sont décodables pour* Σ.

Définition 13 ($\mathcal{I}$**-décodabilité)** *Soit* $\Sigma = \langle \mathcal{X}, \mathcal{P}\rangle$ *une signature MFO, et soit* $\mathcal{I} \subseteq \mathcal{X}$ *un sous-ensemble des variables. Un état de connaissances K sur* $V(\Sigma)$ *est dit* $\mathcal{I}$-décodable *pour* $\mathcal{I}$ *(et* Σ*) s'il est décodable et vérifie* $\forall X \in \mathcal{I}, \exists m \in K\text{: } m \models x_X$. *On note* $D(\mathcal{I})$ *l'ensemble des états de connaissance qui sont* $\mathcal{I}$*-décodables (la signature sera toujours claire).*

Exemple 14 (suite) *L'état de connaissance* $\{x_X\bar{x}_Y\bar{x}_Z x_P x_Q, \bar{x}_X\bar{x}_Y x_Z x_P x_Q, x_X\bar{x}_Y\bar{x}_Z\bar{x}_P\bar{x}_Q\}$ *n'est pas décodable, car deux affectations, différentes sur* $V(\mathcal{P}^{ex})$, *satisfont la variable* x_X *; l'interprétation de la variable X y est donc ambiguë. Par ailleurs, si l'on reprend l'état de connaissance* $\{x_X\bar{x}_Y\bar{x}_Z x_P x_Q, \bar{x}_X\bar{x}_Y x_Z x_P x_Q, \bar{x}_X\bar{x}_Y\bar{x}_Z\bar{x}_P\bar{x}_Q\}$ *de l'exemple 11, on voit qu'il est* $\{X, Z\}$*-décodable, mais qu'il* n'*est* pas $\{X, Y, Z\}$*-décodable, puisqu'aucune de ses affectations ne satisfait la variable* x_Y.

Par construction, l'encodage d'une structure MFO S est décodable pour $\mathcal{I}(S)$, mais il existe des états de connaissances qui sont décodables sans être l'encodage d'aucune structure MFO canonique, comme on vient de le voir (exemple 11). Par ailleurs, il est facile de voir que si un état de connaissance est décodable pour $\mathcal{I}$ et pour $\mathcal{I}'$, il l'est également pour $\mathcal{I} \cup \mathcal{I}'$. L'ensemble des variables étant fini, il s'ensuit qu'un état de connaissance K a un unique ensemble maximal (pour l'inclusion) de variables pour lequel il est décodable ; on note cet ensemble $\mathcal{I}(K)$.

Pour une signature MFO $\Sigma = \langle \mathcal{X}, \mathcal{P}\rangle$, un état de connaissance K sur $V(\Sigma)$ et une affectation $m \in K$, on note $\mathrm{Pred}(m)$ l'ensemble de prédicats $\{P \in \mathcal{P} \mid m \models x_P\}$. Cet ensemble correspond donc à une valeur (celle encodée par m) du domaine d'une structure canonique.

Définition 15 (décodage) *Soit* $\Sigma = \langle \mathcal{X}, \mathcal{P}\rangle$ *une signature MFO, et soit K un état de connaissance décodable pour* Σ. *Le* décodage *de K est la structure MFO canonique* $S = \langle \mathcal{D}, I_\mathcal{X}, I_\mathcal{P}\rangle$ *définie par* $\mathcal{I}(S) = \mathcal{I}(K)$ *et*

- $\mathcal{D} = \{\mathrm{Pred}(m) \mid m \in K\}$,
- $\forall X \in \mathcal{I}(K)\text{: } I_\mathcal{X}(X) = \mathrm{Pred}(m)$, *pour une affectation m arbitraire vérifiant* $m \models x_X$,
- $\forall P \in \mathcal{P}, \forall m \in K\text{: } I_\mathcal{P}(P)(\mathrm{Pred}(m)) = \top \iff P \in \mathrm{Pred}(m)$.

Remarquons que l'interprétation d'une variable est bien définie car K est supposée décodable. Par définition, la structure MFO $\mathrm{Dec}(K)$ est canonique ; par ailleurs, la proposition suivante découle directement des définitions.

Proposition 16 *Pour toute structure MFO S, on a* $\mathrm{Dec}(\mathrm{Enc}(S)) = S^{\downarrow}$. *En particulier, si S est canonique, on a* $\mathrm{Dec}(\mathrm{Enc}(S)) = S$.

6 Reformulation : syntaxe

Nous nous intéressons désormais à la reformulation, dans la logique $\mathsf{S5_F}$, d'une *formule MFO* Φ. Étant donnée une telle formule, nous proposons la construction d'une formule de $\mathsf{S5_F}$, notée $\mathcal{R}(\Phi)$ (pour « reformulation »), pour laquelle décoder les modèles au sens de la définition 15 donne l'ensemble des modèles canoniques de Φ. La formule $\mathcal{R}(\Phi)$ a en fait pour modèles *exactement* les états de connaissance K qui sont décodables et dont le décodage donne un modèle canonique de Φ. En particulier, elle a en général plus de modèles que Φ, étant donné que plusieurs modèles de $\mathcal{R}(\Phi)$ peuvent se décoder en la même structure MFO. Cela ne prête cependant pas à conséquence, en fonction du problème auquel on s'intéresse ; notre construction permet en particulier de traiter le problème de la satisfaisabilité pour MFO comme un problème de satisfaisabilité pour $\mathsf{S5_F}$, comme nous en discuterons dans la partie 7.

La sémantique de la reformulation étant définie dans la partie 5, la construction est très directe. Les points non triviaux sont la restriction aux états de connaissance décodables, et la traduction de la quantification existentielle du premier ordre en l'oubli de S5.

Définition 17 (reformulation de la décodabilité) *Soit $\Sigma = \langle \mathcal{X}, \mathcal{P} \rangle$ une signature MFO, et soit $\mathcal{I} \subseteq \mathcal{X}$ un sous-ensemble de ses variables. On note Δ la formule de S5 définie par*

$$\Delta = \bigwedge_{X \in \mathcal{X}} \bigwedge_{P \in \mathcal{P}} (\Box(x_X \to x_P) \vee \Box(x_X \to \neg x_P))$$

et par $\Delta(\mathcal{I})$ celle définie par

$$\Delta(\mathcal{I}) = \Delta \wedge \bigwedge_{X \in \mathcal{I}} \Diamond x_X.$$

La formule Δ impose que toutes les x_X-affectations satisfassent exactement les même variables x_P. La formule $\Delta(\mathcal{I})$ impose en plus l'existence d'une x_X-affectation pour toute variable X dans $\mathcal{I}$.

Proposition 18 *Soit $\Sigma = \langle \mathcal{X}, \mathcal{P} \rangle$ une signature MFO, et soit $\mathcal{I} \subseteq \mathcal{X}$ un sous-ensemble de ses variables. Alors pour un état de connaissance K, on a $K \models \Delta(\mathcal{I}) \iff K \in D(\mathcal{I})$.*

Preuve. Supposons tout d'abord $K \models \Delta(\mathcal{I})$. Pour montrer $K \in D_\Sigma$, considérons deux affectations $m, m' \in K$ vérifiant $m \models x_X$ et $m' \models x_X$ pour une certaine variable $X \in \mathcal{X}$. Supposons par l'absurde $m|_{V(\mathcal{P})} \neq m'|_{V(\mathcal{P})}$: disons, sans perte de généralité, $m \models x_P$ et $m \models \neg x_P$ pour un certain $P \in \mathcal{P}$. Alors on a $K \models \Diamond(x_X \wedge x_P)$ et $K \models \Diamond(x_X \wedge \neg x_P)$, ce qui est absurde puisque l'on a $K \models \Delta$ et donc en particulier $K \models \Box(x_X \to x_P) \vee \Box(x_X \to \neg x_P)$. Les deux affectations sont donc identiques sur $V(\mathcal{P})$, et l'on a $K \in D_\Sigma$. Pour montrer que K est $\mathcal{I}$-décodable, il suffit de remarquer que pour toute variable $X \in \mathcal{I}$, on a $K \models \Diamond x_X$; par définition, cela signifie que K contient au moins une assignation m vérifiant $m \models x_X$. On a donc finalement $K \in D(\mathcal{I})$.

Réciproquement, supposons $K \in D(\mathcal{I})$. Pour montrer $K \models \Delta$, supposons par l'absurde qu'il existe $X \in \mathcal{X}$ et $P \in \mathcal{P}$ tels que K ne satisfait pas la sous-formule $\Box(x_X \to x_P) \vee \Box(x_X \to \neg x_P)$. Alors K satisfait à la fois $\Diamond\neg(x_X \to x_P)$ et $\Diamond\neg(x_X \to \neg x_P)$, et contient donc deux affectations m et m' satisfaisant respectivement $x_X \wedge \neg x_P$ et $x_X \wedge x_P$. Puisqu'il contient deux affectations satisfaisant x_X dont la projection sur $V(\mathcal{P})$ diffère, K n'est pas décodable, ce qui contredit l'hypothèse $K \in D(\mathcal{I})$. Enfin, pour chaque variable $X \in \mathcal{I}$, K contient par définition de $D(\mathcal{I})$ une assignation m vérifiant $m \models x_X$, et satisfait donc chaque sous-formule $\Diamond x_X$. On a donc finalement $K \models \Delta(\mathcal{I})$. □

Avec cette définition, nous pouvons désormais donner notre reformulation des formules MFO en formules de $\mathsf{S5_F}$. Par souci de simplicité, nous omettons la quantification universelle ; une formule $\forall X \cdot \Phi$ peut simplement être réécrite en $\neg\exists X \cdot \neg\Phi$ avant la reformulation.

Définition 19 (reformulation) *Soit $\Sigma = \langle \mathcal{X}, \mathcal{P} \rangle$ une signature MFO. Pour une formule MFO Φ sur Σ, la* reformulation *de Φ, notée $\mathcal{R}(\Phi)$, est la formule de $\mathsf{S5_F}$ sur $V(\Sigma)$ définie inductivement par :*

$$\begin{aligned}
\mathcal{R}(P(X)) &= \Delta(\{X\}) \wedge \Box(x_X \to x_P) \\
\mathcal{R}(\neg\Phi) &= \Delta(\mathrm{Free}(\Phi)) \wedge \neg\mathcal{R}(\Phi) \\
\mathcal{R}(\Phi \wedge \Psi) &= \mathcal{R}(\Phi) \wedge \mathcal{R}(\Psi) \\
\mathcal{R}(\Phi \vee \Psi) &= \Delta(\mathrm{Free}(\Phi \vee \Psi)) \wedge (\mathcal{R}(\Phi) \vee \mathcal{R}(\Psi)) \\
\mathcal{R}(\exists X \cdot \Phi) &= \Delta(\mathrm{Free}(\Phi) \setminus \{X\}) \wedge \mathsf{F} x_X \cdot \mathcal{R}(\Phi).
\end{aligned}$$

On remarque que, en utilisant ainsi un *connecteur* pour l'oubli, la reformulation que nous proposons est polynomiale.

Proposition 20 *Soit Σ une signature MFO, et soit Φ une formule sur Σ. Alors la formule $\mathcal{R}(\Phi)$ vérifie*

$$K \models \mathcal{R}(\Phi) \iff K \in D(\mathrm{Free}(\Phi)) \text{ et } \mathrm{Dec}(K) \models \Phi$$

Preuve. On procède par induction sur la structure de Φ. Le cas de base est évident. Pour la négation, on conclut facilement après avoir remarqué $\mathrm{Free}(\neg\Phi) = \mathrm{Free}(\Phi)$. Pour la conjonction et la disjonction, on remarque tout d'abord $\mathrm{Free}(\Phi \wedge \Psi) = \mathrm{Free}(\Phi \vee \Psi) = \mathrm{Free}(\Phi) \cup \mathrm{Free}(\Psi)$. On en déduit qu'un état de connaissances est décodable pour $\mathrm{Free}(\Phi \wedge \Psi)$ si et seulement s'il l'est pour $\mathrm{Free}(\Phi \vee \Psi)$, si et seulement s'il l'est pour $\mathrm{Free}(\Phi)$ et pour $\mathrm{Free}(\Psi)$. On conclut alors facilement avec la sémantique de la conjonction et de la disjonction.

Finalement, soit Φ une formule de la forme $\exists X \cdot \Psi$. Soit tout d'abord K un modèle de $\mathcal{R}(\Phi)$, c'est-à-dire de la formule $\Delta(\text{Free}(\Psi) \setminus \{X\}) \wedge \mathsf{F} x_X \cdot \mathcal{R}(\Psi)$. Puisque l'on a clairement $\text{Free}(\Phi) = \text{Free}(\Psi) \setminus \{X\}$, on déduit de la proposition 18 que l'on a $K \in D(\text{Free}(\Phi))$. Nous montrons maintenant $\text{Dec}(K) \models \Phi$. Pour cela, soit K' un état de connaissance vérifiant $K' \models \mathcal{R}(\Psi)$ et $K \in \text{Forget}(x_X, K')$, qui existe puisque K satisfait $\mathsf{F} x_X \cdot \mathcal{R}(\Psi)$. Par hypothèse d'induction, K' est décodable pour $\text{Free}(\Psi)$ et l'on a $\text{Dec}(K') \models \Psi$. Notons $S = \text{Dec}(K)$ et $S' = \text{Dec}(K')$. Par définition de l'oubli et de la fonction de décodage Dec, on voit aisément que K et K' encodent le même domaine de valeurs et la même interprétation de toutes les variables, sauf éventuellement de X. On déduit $S' = S_{X \leftarrow v}$ pour une certaine valeur v, et comme on a $S' \models \Psi$, on conclut finalement $S \models \exists X \cdot \Psi$.

Réciproquement, soit K un état de connaissance décodable pour $\text{Free}(\Phi)$ et satisfaisant $\text{Dec}(K) \models \Phi$. Par la proposition 18, on a bien $K \models \Delta(\text{Free}(\Phi))$. Nous montrons maintenant $K \models \mathsf{F} x_X \cdot \mathcal{R}(\Psi)$. Pour cela, notons $S = \text{Dec}(K)$. On sait donc qu'il existe une valeur v du domaine de S satisfaisant $S_{X \leftarrow v} \models \Psi$. Par hypothèse d'induction, l'état de connaissance K_v défini comme $\text{Enc}(S_{X \leftarrow v})$ satisfait donc la formule $\mathcal{R}(\Psi)$. Par définition de $S_{X \leftarrow v}$ et de la fonction d'encodage Enc, on voit aisément que K et K_v vérifient $K|_{V(\Sigma) \setminus \{x_X\}} = (K_v)|_{V(\Sigma) \setminus \{x_X\}}$. On a donc $K \in \text{Forget}(x_X, K_v)$, et donc on a bien $K \models \mathsf{F} x_X \cdot \mathcal{R}(\Psi)$. □

7 Vers un prouveur efficace ?

Comme déjà évoqué, cet article se concentre sur la reformulation, mais nous n'avons pas pour l'instant de résultats expérimentaux mettant cette reformulation en œuvre pour le problème de satisfaisabilité de MFO. Nous décrivons toutefois ici les grandes lignes qui guideront les premiers pas de cette mise en œuvre.

Compilation Si l'on met de côté le développement de prouveurs dédiés pour le problème de satisfaisabilité dans $\mathsf{S5_F}$, ce qui peut être une perspective intéressante en soi, l'approche naturelle pour utiliser notre reformulation consiste à l'utiliser comme un guide pour *compiler* la formule MFO donnée Φ. En d'autres termes, plutôt que de calculer telle quelle la formule $\mathcal{R}(\Phi)$, puis de tester sa satisfaisabilité, on peut calculer directement une représentation de $\mathit{Mods}(\mathcal{R}(\Phi))$, dans un langage adéquat. Pour cela, on peut alors partir, *bottom-up*, de la représentation des atomes de la forme $\Box\varphi$, puis les combiner avec les opérations dédiées du langage correspondant à la négation, la conjonction, la disjonction et l'oubli. Une fois toute la formule MFO ainsi compilée, il suffit de tester si la représentation résultante, dans le langage choisi, contient au moins un état de connaissance, pour savoir si la formule Φ est satisfaisable.

Une première cartographie de langages et de la complexité des opérations a été établie pour S5, pour plusieurs langages utilisant la forme normale disjonctive au niveau épistémique [3]. Depuis, un nouveau langage a été proposé, celui des ESD (*epistemic splitting diagrams*) [15]. Tous ces langages sont des candidats pour l'approche par compilation pour le problème de la satisfaisabilité dans MFO, mais il s'agit désormais de tester leur efficacité en pratique (le problème étant NEXPTIME-complet alors que notre reformulation est polynomiale, on sait que les transformations feront dans le cas général exploser la taille de la représentation dans le langage choisi, si celui-ci permet de vérifier facilement la satisfaisabilité).

Outre la comparaison de l'efficacité de ces divers langages dans ce nouveau contexte, notre évaluation expérimentale cherchera à valider l'utilité de l'approche en comparant les résultats à l'état de l'art de la résolution du problème de satisfiabilité pour MFO. En l'occurrence, il s'agirait (i) de se mesurer à des prouveurs de premier ordre plus généraux, tels que iProver, SPASS ou Vampire [9, 18, 10], et (ii) d'essayer d'extraire la partie spécifique à MFO de l'approche par réduction à SAT [16]. Nous avons des *benchmarks* issus de travaux sur le *model checking* symbolique d'existence d'annonces arbitraires [7], et nous comptons récupérer des instances monadiques dans les dépôts de problèmes pour la preuve automatique de théorème, comme TPTP [17].

Variantes de la reformulation De façon orthogonale au paragraphe précédent, on peut s'intéresser à différentes variantes de la reformulation que nous avons proposée. On peut pour cela jouer sur la notion de décodabilité. On pourrait par exemple la restreindre aux états de connaissances dans lesquels chaque interprétation $I_{\mathcal{X}}(X) = v$ est représentée par une affectation propre, différente des autres (ne satisfaisant que x_X dans l'ensemble $V(\mathcal{X})$, en particulier), et redéfinir la notion d'encodage et la formule représentant la décodabilité selon les mêmes lignes. On aurait alors des reformulations possédant moins de modèles que celle proposée dans cet article, pour une même formule MFO. Il reste toutefois à évaluer si cela rend le problème de satisfaisabilité ou le calcul des versions compilées plus facile ou plus complexe en pratique, ceci pouvant bien entendu dépendre du langage pour ce qui est de la compilation.

Dans le même esprit, on peut montrer assez facilement que la reformulation proposée dans cet article peut être modifiée en supprimant la conjonction avec la formule $\Delta(\cdot)$ à toutes les étapes, sauf lors de l'appel sur la formule initiale. La reformulation de Φ devient alors $\Delta(\text{Free}(\Phi)) \wedge \mathcal{R}'(\Phi)$, où

$\mathcal{R}'(\Phi)$ est simplement définie inductivement par

$$\begin{aligned}\mathcal{R}(P(X)) &= \Box(x_X \to x_P)\\ \mathcal{R}(\neg\Phi) &= \neg\mathcal{R}(\Phi)\\ \mathcal{R}(\Phi \wedge \Psi) &= \mathcal{R}(\Phi) \wedge \mathcal{R}(\Psi)\\ \mathcal{R}(\Phi \vee \Psi) &= \mathcal{R}(\Phi) \vee \mathcal{R}(\Psi)\\ \mathcal{R}(\exists X \cdot \Phi) &= \mathsf{F}x_X \cdot \mathcal{R}(\Phi).\end{aligned}$$

On peut imaginer qu'une telle reformulation engendrera des formules de $\mathsf{S5_F}$ moins complexes, mais ceci reste également à vérifier en pratique.

Traitement de l'égalité Nous avons supposé ici que le langage MFO ne contenait pas le prédicat binaire d'égalité $=$. Nous pensons qu'il est toutefois possible d'adapter notre approche pour qu'elle traite MFO avec égalité. Le problème principal est le suivant : deux valeurs peuvent alors être nécessairement distinctes bien qu'elles satisfassent exactement les mêmes prédicats. Une approche naïve consisterait donc à ajouter suffisamment de prédicats « fictifs » pour que ces valeurs soient vues comme distinctes dans un état de connaissance. Nous laissons cette piste pour un travail futur.

Autres problèmes Parmi les utilisations potentielles de la logique MFO, on peut montrer que le problème consistant à déterminer si un Dec-POMDP qualitatif [5] a une solution à un horizon donné peut être encodé comme la satisfaisabilité d'une formule MFO « naturelle ». En un mot, un Dec-POMDP qualitatif est une instance d'un problème de planification multi-agent, coopérative, et à observabilité partielle, pour laquelle on cherche de façon centralisée une solution qui sera exécutée de façon décentralisée ; par exemple, une stratégie pour chaque agent qui ne dépende que des observations qu'il reçoit. Ce sont ces stratégies que l'on peut représenter comme les modèles d'une formule MFO.

Pour ce type d'exemples, on peut s'intéresser à d'autres problèmes que la simple satisfaisabilité. Dans le cas de la planification, on peut vouloir des stratégies de taille minimale, ou encore des contrôleurs possédant un nombre d'états fixé *a priori*. On est alors amené à rechercher des modèles MFO ayant le plus petit domaine possible, par exemple, ou encore à générer tous les modèles d'une formule donnée. La reformulation que nous avons proposée ici, ou des variantes, pourraient s'avérer utiles pour ces objectifs, notamment en utilisant des approches de compilation pour S5 ou $\mathsf{S5_F}$.

8 Conclusion

Dans cet article, nous avons montré comment la satisfaisabilité pour la logique monadique du premier ordre peut être reformulée en la satisfaisabilité pour la logique S5, étendue avec une opération d'oubli de variable propositionnelle (voir la définition 19). Cette reformulation, assez naturelle, préserve en outre l'ensemble des modèles canoniques, au sens où tout modèle de la formule $\mathsf{S5_F}$ obtenue peut être décodé en un modèle canonique de la formule MFO de départ, et où tout modèle canonique de la formule MFO de départ peut être obtenu en décodant un modèle de la formule $\mathsf{S5_F}$ obtenue.

Il s'agit d'un premier pas vers l'application à MFO des techniques de compilation de connaissances envisagées pour S5 dans la littérature ; on peut raisonnablement espérer obtenir des résultats intéressants en pratique pour le problème de la satisfaisabilité, ou pour des problèmes plus « globaux » comme la recherche d'un modèle minimal, où le fait de disposer d'une représentation de l'ensemble des modèles pourrait s'avérer un atout significatif.

Remerciements

Nous tenons à remercier un relecteur anonyme pour ses commentaires particulièrement pertinents, et notamment pour avoir vu une erreur dans une précédente généralisation de notre approche à MFO avec égalité.

Références

[1] Baader, Franz, Ian Horrocks, Carsten Lutz et Ulrike Sattler: *An Introduction to Description Logic*. Cambridge University Press, 2017, ISBN 978-0-521-69542-8.

[2] Bachmair, Leo, Harald Ganzinger et Uwe Waldmann: *Set constraints are the monadic class*. Dans *Proceedings of the Eighth Annual Symposium on Logic in Computer Science (LICS '93), Montreal, Canada, June 19-23, 1993*, pages 75–83, 1993. https://doi.org/10.1109/LICS.1993.287598.

[3] Bienvenu, Meghyn, Hélène Fargier et Pierre Marquis: *Knowledge Compilation in the Modal Logic S5*. Dans *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, 2010. http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1696.

[4] Bloem, Roderick, Robert Könighofer et Martina Seidl: *SAT-based synthesis methods for safety specs*. Dans *Verification, Model Checking, and Abstract Interpretation - 15th International Conference, VMCAI 2014, San Diego, CA, USA, January 19-21, 2014, Proceedings*, pages 1–20, 2014. https://doi.org/10.1007/978-3-642-54013-4_1.

[5] Brafman, Ronen I., Guy Shani et Shlomo Zilberstein: *Qualitative Planning under Partial Observability in*

Multi-Agent Domains. Dans *Proc. 27th AAAI Conference on Artificial Intelligence (AAAI 2013)*, pages 130–137. AAAI Press, 2013.

[6] Caridroit, Thomas, Jean Marie Lagniez, Daniel Le Berre, Tiago de Lima et Valentin Montmirail: *A SAT-Based Approach for Solving the Modal Logic S5-Satisfiability Problem*. Dans *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3864–3870, 2017. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14380.

[7] Charrier, Tristan, Sophie Pinchinat et François Schwarzentruber: *Model checking against arbitrary public announcement logic: A first-order-logic prover approach for the existential fragment*. Dans *Workshop Dynamic Logic: new trends and applications, LNCS*, 2017.

[8] Fitting, Melvin: *First-Order Logic and Automated Theorem Proving, Second Edition*. Graduate Texts in Computer Science. Springer, 1996, ISBN 978-1-4612-7515-2.

[9] Korovin, Konstantin: *iProver - An Instantiation-Based Theorem Prover for First-Order Logic (System Description)*. Dans *Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings*, pages 292–298, 2008. https://doi.org/10.1007/978-3-540-71070-7_24.

[10] Kovács, Laura et Andrei Voronkov: *First-Order Theorem Proving and Vampire*. Dans *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, pages 1–35, 2013. https://doi.org/10.1007/978-3-642-39799-8_1.

[11] Lagniez, Jean Marie, Daniel Le Berre, Tiago de Lima et Valentin Montmirail: *A Recursive Shortcut for CEGAR : Application To The Modal Logic K Satisfiability Problem*. Dans *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 674–680, 2017. https://doi.org/10.24963/ijcai.2017/94.

[12] Lewis, Harry R.: *Complexity results for classes of quantificational formulas*. J. Comput. Syst. Sci., 21(3):317–353, 1980. https://doi.org/10.1016/0022-0000(80)90027-6.

[13] Löwenheim, Leopold: *Über möglichkeiten im relativkalkül*. Mathematische Annalen, 76(4) :447–470, 1915.

[14] Marquis, Pierre: *Compile!* Dans *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 4112–4118, 2015. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9596.

[15] Niveau, Alexandre et Bruno Zanuttini: *Efficient representations for the modal logic S5*. Dans *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1223–1229, 2016. http://www.ijcai.org/Abstract/16/177.

[16] Ramachandran, Deepak et Eyal Amir: *Compact Propositional Encodings of First-Order Theories*. Dans *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 340–345, 2005. http://www.aaai.org/Library/AAAI/2005/aaai05-054.php.

[17] Sutcliffe, Geoff: *The TPTP Problem Library and Associated Infrastructure. From CNF to TH0, TPTP v6.4.0*. Journal of Automated Reasoning, 59(4) :483–502, 2017.

[18] Weidenbach, Christoph, Bernd Gaede et Georg Rock: *SPASS & FLOTTER, version 0.42*. Dans McRobbie, M. A. et J. K. Slaney (rédacteurs): *Proceedings of the 13th International Conference on Automated Deduction (CADE-13)*, Lecture Notes in Artificial Intelligence, pages 141–145. Springer, 2003, ISBN 3-540-61511-3.

[19] Zhang, Yan et Yi Zhou: *Knowledge forgetting : Properties and applications*. Artif. Intell., 173(16-17) :1525–1537, 2009. https://doi.org/10.1016/j.artint.2009.07.005.