Oubli de littéraux avec symboles propositionnels variables

Forgetting literals with varying propositional variables

Yves Moinard
INRIA/IRISA
Campus de Beaulieu, 35042 RENNES-Cedex FRANCE
courriel : moinard@irisa.fr

Résumé

*La notion d'*oubli de symboles propositionnels *(réduction du vocabulaire) vient d'être généralisée en la notion d'*oubli de littéraux. *Le but est de fournir des méthodes constructives facilitant le calcul effectif de divers formalismes de représentation des connaissances. La notion est encore étendue ici, en permettant à des symboles propositionnels de varier. Les définitions (syntaxiques et sémantiques) sont une extension naturelle des notions précédentes. L'application au calcul effectif de la circonscription est détaillée. L'apport des symboles variables est clair dans ce cas : les deux étapes des méthodes antérieures sont réduites à une seule. Cela permet de réexaminer un résultat vieux de quinze ans, et de fournir des points de départ en vue d'une extension à d'autres formalismes connus.*

Mots Clef

Représentation des connaissances, calcul propositionnel, raisonnement non monotone.

Abstract

The old logical notion of forgetting propositional symbols *(or reducing the logical vocabulary) has just been generalized to* forgetting literals *in order to help the computation of various formalisms in knowledge representation. We extend this notion, by allowing propositional symbols to vary. We describe the new notion, on the syntactical and the semantical side. Then, we show how to apply it to the computation of circumscription. This has been done already with standard literal forgetting, but here we show how introducing varying propositional symbols simplifies significantly the computation. We revisit a fifteen years old result about computing circumscription, showing that it can be improved in the same way. We provide hints in order to apply this forgetting method also to other logical formalisms.*

Keywords

Knowledge representation, propositional calculus, non monotonic reasoning

1 Introduction

La notion d'*oubli de symboles propositionnels*, que l'on fait remonter à un papier de Boole de 1854 sous le nom d'*elimination of middle terms*, est bien connue en logique et en représentation des connaissances (voir par exemple [5, 4, 11]). Il s'agit de réduire le langage, en supprimant des symboles propositionnels. Par exemple, dans la formule

$$(Oiseau \wedge \neg exceptionnel \rightarrow vole) \wedge \neg exceptionnel,$$

"oublier" le symbole $exceptionnel$, considéré ici comme auxiliaire, fournira la formule $Oiseau \rightarrow vole$.
Récemment, Lang et al. ont étendu cette notion de façon significative en permettant plus finement l'*oubli de littéraux* [2]. Dans l'exemple ci-dessus, on oublie en fait le *littéral* $\neg exceptionnel$.
Les nouvelles définitions constituent une extension naturelle des définitions classiques. Lang et al. ont montré que, comme la notion originale, la nouvelle version présente un grand intérêt pour la représentation et le traitement des connaissances. Cela permet dans certains cas de faciliter les calculs, et Lang et al. détaillent diverses manières d'obtenir les formules qui oublient les littéraux, afin de fournir des méthodes pratiques de calcul de divers formalismes. L'exemple fourni par les auteurs concerne la circonscription, qui constitue en fait l'exemple le plus direct d'utilisation de cette méthode. La circonscription est un mécanisme de *minimisation* de certains symboles. Circonscrire le symbole $exceptionnel$ dans la sous-formule $(Oiseau \wedge \neg exceptionnel \rightarrow vole)$ de l'exemple précédant fournirait la seconde clause de la conjonction donnée ci-dessus : $\neg exceptionnel$.
Même sur cet exemple simple, une complication apparaît : il faut autoriser au moins un des symboles non circonscrits à *varier* afin d'obtenir un résultat intéressant. Si les deux symboles $oiseau$ et $vole$ varient, la circonscription fournit bien $\neg exceptionnel$, mais ce n'est pas le cas sinon. Le résultat de Lang et al., que les auteurs montrent équivalent, d'une certaine façon, à un résultat connu depuis une quin-

zaine d'année [9], exige deux étapes. L'étape (1) traite la circonscription sans symbole variable et l'étape (2) se ramène à (1) au prix d'une certaine complication.

La notion d'oubli de littéraux originale conduit naturellement à la notion d'oubli de littéraux avec symboles propositionnels variables. De plus, l'application à la circonscription se trouvera améliorée.

La section 2 fournit les définitions et notations utiles (calcul propositionnel fini, "oubli de symboles propositionnels"). La section 3 rappelle ce qu'est la notion d'oubli de littéraux introduite par Lang et al. La section 4 étend cette notion au cas où des symboles propositionnels sont autorisés à varier. Plusieurs définitions et caractérisations, syntaxiques et sémantiques, sont détaillées, ainsi que quelques résultats utiles dans la suite. La section 5 décrit la méthode de calcul de la circonscription grâce à la notion d'oubli de littéraux.

2 Définitions et notations de base, oubli de symboles propositionnels

On se place dans un langage propositionnel fini $\mathbf{PL}$. Comme d'habitude, $\mathbf{PL}$ désigne aussi l'ensemble de toutes les formules du langage, le *vocabulaire de* $\mathbf{PL}$ est un ensemble fini de *symboles propositionnels*, noté $\mathcal{V}(\mathbf{PL})$.

Les lettres φ, ψ désignent des formules de $\mathbf{PL}$.

Deux constantes logiques $\top$ et $\perp$ désignent respectivement les formules vraies et fausses de $\mathbf{PL}$. Les lettres ω, μ, ν désignent des *interprétations de* $\mathbf{PL}$, assimilées à des sous-ensembles de $\mathcal{V}(\mathbf{PL})$. Les notations $\omega \models \varphi$ et $\omega \models X$ pour un ensemble X de formules sont définies classiquement. Pour un ensemble E, $\mathcal{P}(E)$ désigne l'ensemble des sous-ensembles de E. L'ensemble $\mathcal{P}(\mathcal{V}(\mathbf{PL}))$ des interprétations pour $\mathbf{PL}$ est noté **Mod**. Un *modèle* de X est une interprétation ω telle que $\omega \models X$, $\mathbf{Mod}(\varphi)$ et $\mathbf{Mod}(X)$ désignent respectivement les ensembles de modèles de $\{\varphi\}$ et de X. Un *littéral* est un symbole p de $\mathcal{V}(\mathbf{PL})$ (*littéral positif*) ou sa négation $\neg p$ (*littéral négatif*). Une *clause* (respectivement un *terme*) est une disjonction (respectivement une conjonction) de littéraux. Des sous-ensembles de $\mathcal{V}(\mathbf{PL})$ pourront être désignés par P, Q, V et alors P^+ (respectivement P^-) désigne l'ensemble des littéraux positifs (respectivement négatifs) construits sur P, et $P^\pm$ désigne l'ensemble $P^+ \cup P^-$ de tous les littéraux construits sur P (P et P^+ sont souvent assimilés).

Pour tout ensemble (fini) X de formules, $\bigwedge X$ (respectivement $\bigvee X$) désigne la conjonction (respectivement disjonction) de toutes les formules de X. On a : $\bigwedge X \equiv X$, $\bigwedge \emptyset \equiv \top$ et $\bigvee \emptyset \equiv \perp$. L'ensemble des symboles propositionnels apparaissant dans X est noté $\mathcal{V}(X)$.

Une *forme normale disjonctive* ou FND (respectivement *forme normale conjonctive* ou FNC) de φ est une disjonction de termes consistants (respectivement une conjonction de clauses non triviales) équivalente à φ. Une disjonction (respectivement conjonction) vide est équivalente à $\perp$ (respectivement $\top$).

Une *FNC primitive* est une FNC qui contient seulement des *impliqués premiers* : [deux clauses de la FNC ne sont jamais équivalentes, et] pour toutes clauses c, c_i, si $\varphi \models c$ et $c \models c_i$ pour une c_i de la FNC, alors $c_i \models c$.

Un ensemble L de littéraux de $V^\pm$ (et le terme $\bigwedge L$) est *consistant et complet en* V si chaque symbole propositionnel de V apparaît exactement une fois dans L ; la clause $\bigvee L$ est alors *non triviale et complète en* V. Pour tout ensemble L de littéraux de $\mathbf{PL}$, $\neg L = \{p \,/\, p \in \mathcal{V}(\mathbf{PL}), \neg p \in L\} \cup \{\neg p \,/\, p \in \mathcal{V}(\mathbf{PL}), p \in L\}$ désigne l'ensemble des *littéraux complémentaires* de ceux L. On utilisera aussi les notions et notations suivantes (la plupart provenant de [2]) :

Si φ est une formule et p un symbole propositionnel de $\mathbf{PL}$, alors $\varphi_{p \leftarrow 1}$ (respectivement $\varphi_{p \leftarrow 0}$) est la formule obtenue à partir de φ en remplaçant chaque occurrence de p par $\top$ (respectivement $\perp$). Si l est le littéral positif p (respectivement le littéral négatif $\neg p$), alors $\varphi_{l \leftarrow i}$ désigne la formule $\varphi_{p \leftarrow i}$ (respectivement $\varphi_{p \leftarrow (1-i)}$), pour $i \in \{0, 1\}$.

Notations 2.1 1. *Si $v_1, \cdots, v_n$ sont des symboles propositionnels, $\varphi_{(v_1 \leftarrow i_1, \cdots, v_n \leftarrow i_n)}$, où $i_j \in \{0, 1\}$, désigne la formule $(\cdots((\varphi_{v_1 \leftarrow i_1})_{v_2 \leftarrow i_2})\cdots)_{v_n \leftarrow i_n}$.*
Si les v_j de la liste sont tous distincts, alors l'ordre des v_j est sans conséquence pour le résultat final. Ainsi, si V_1 et V_2 sont des sous-ensembles disjoints de V, on peut définir $\varphi_{[V_1 \leftarrow 1, V_2 \leftarrow 0]}$ par $\varphi_{(v_1 \leftarrow 1, \cdots, v_n \leftarrow 1, v_{n+1} \leftarrow 0, \cdots, v_{n+m} \leftarrow 0)}$, où $(v_1, \cdots, v_n)$ et $(v_{n+1}, \cdots, v_{n+m})$ sont deux énumérations de tous les éléments de V_1 et V_2 respectivement.

2. *Si $L = (l_1, \cdots, l_n)$ est une liste de littéraux, $\varphi_{(l_1 \leftarrow i_1 \cdots l_n \leftarrow i_n)}$ désigne la formule $(\cdots((\varphi_{l_1 \leftarrow i_1})_{l_2 \leftarrow i_2})\cdots)_{l_n \leftarrow i_n}$.*

3. *Supposons $\mathcal{V}(\mathbf{PL})^\pm$ arbitrairement ordonné. Si $L_1, \cdots, L_m$ sont des ensembles disjoints de littéraux, $\varphi_{\langle L_1 \leftarrow i_1, \cdots, L_m \leftarrow i_m \rangle}$ désigne la formule $\varphi_{(l_1 \leftarrow k_1, \cdots, l_n \leftarrow k_n)}$, où $(l_1, \cdots, l_n)$ est l'énumération de $L_1 \cup \cdots \cup L_m$ qui respecte l'ordre choisi pour l'ensemble de tous les littéraux, et où, pour chaque $j \in \{1, \cdots, n\}$, si l_j est dans la partie L_r pour une valeur $r \in \{1, \cdots, m\}$, alors k_j est égal à i_r.*

Rappelons maintenant une notion bien connue :

Définition 2.2 (Oubli de symboles)
Si $V \subseteq \mathcal{V}(\mathbf{PL})$ et $\varphi \in \mathbf{PL}$, $ForgetV(\varphi, V)$ désigne une formule du langage $\mathbf{PL}_{\overline{V}}$, construite sur le vocabulaire

$\overline{V} = \mathcal{V}(\mathbf{PL}) - V$, équivalente à φ sur ce langage restreint :

$$ForgetV(\varphi, V) \equiv Th(\varphi) \cap \mathbf{PL}_{\overline{V}} \quad \textit{où}$$

$Th(\varphi) = \{\varphi' \in \mathbf{PL} / \varphi \models \varphi'\}$.

Pour tout $\psi \in \mathbf{PL}_{\overline{V}}$ on a $\varphi \models \psi$ ssi $ForgetV(\varphi, V) \models \psi$.

Différentes façons d'obtenir $ForgetV(\varphi, V)$ sont bien connues, voici les deux plus faciles (à décrire) :

1. Dans une FND φ, supprimer tous les littéraux de $V^{\pm}$.
2. Dans une FNC primitive φ, supprimer les clauses contenant un symbole de V.

Les remarques suivantes, de démonstration immédiates, vont être très utiles.

Remarque 2.3 *Quand on considère une formule équivalente à un ensemble $Th(\varphi) \cap X$, l'ensemble de formules X peut être remplacé par tout ensemble Y ayant la même $\wedge$-fermeture que X, c'est-à-dire satisfaisant*
$\{\bigwedge X' / X' \subseteq X\} = \{\bigwedge X' / X' \subseteq Y\}$.
En effet, nous avons :

– *Si X et Y ont la même $\wedge$-fermeture, alors $Th(\varphi) \cap X \equiv Th(\varphi) \cap Y$.*
– *La réciproque est vraie,* à condition d'identifier les formules équivalentes *: si $Th(\varphi) \cap X \equiv Th(\varphi) \cap Y$ pour toute formule φ alors X et Y ont la même $\wedge$-fermeture.*

Puisque l'on se limite aux langages propositionnels finis, il existe un unique plus petit (pour l'inclusion, et à l'équivalence logique près) ensemble possible, le $\wedge$-réduit de X, *égal à l'ensemble*
$X - \{\varphi \in X / \varphi$ *est dans la $\wedge$-fermeture de* $X - \{\varphi\}\}$.
Ainsi, X peut être remplacé par tout ensemble qui contient le $\wedge$-réduit de X et est inclus dans la $\wedge$-fermeture de X.

Ainsi, au lieu de considérer tout l'ensemble $X = \mathbf{PL}_{\overline{V}}$ dans $ForgetV(\varphi, V) \equiv Th(\varphi) \cap \mathbf{PL}_{\overline{V}}$ (définition 2.2), peut-on considérer l'ensemble $X' = \{\bigvee Y / Y \subseteq \overline{V}^{\pm}\}$ de toutes les clauses construites sur $\overline{V}$, le plus petit (pour $\subseteq$) ensemble X'' qui peut être considéré ici étant l'ensemble de ces clauses qui sont non triviales et complètes en $\overline{V}$.

Voyons maintenant l'aspect sémantique. L'ensemble des modèles de $ForgetV(\varphi, V)$ est l'ensemble de toutes les interprétations de $\mathbf{PL}$ qui coïncident avec un modèle de φ pour tous les symboles propositionnels extérieurs à V :
$\mathbf{Mod}(ForgetV(\varphi, V)) =$
$$\{\omega \in \mathbf{Mod} \ / \ \exists \omega', \ \omega' \models \varphi \ \text{ et } \ \omega \cap \overline{V} = \omega' \cap \overline{V}\}.$$
Ces caractérisations syntaxiques et sémantiques justifient bien le nom $ForgetV$ (“Oubli de Variables”, ici “variable” = “symbole propositionnel”).

(Deux exemples sont fournis en fin de section 4.)

3 Oubli de littéraux

L'oubli de symboles du vocabulaire a été généralisé en la notion plus fine d'oubli de littéraux. Examinons d'abord l'aspect sémantique, le plus aisé à définir.

Définition 3.1 *[2, pp. 396–397] Soit ω une interprétation pour $\mathbf{PL}$, p un symbole propositionnel de $\mathbf{PL}$ et L un ensemble consistant de littéraux de $\mathbf{PL}$.*
On définit les interprétations suivantes :
$Force(\omega, p) = \omega \cup \{p\}$ *et* $Force(\omega, \neg p) = \omega - \{p\}$
et plus généralement, $Force(\omega, L) =$
$\omega \cup \{p / p \in \mathcal{V}(\mathbf{PL}), p \in L\} - \{p / p \in \mathcal{V}(\mathbf{PL}), \neg p \in L\}$.

On a donc, pour tout symbole propositionnel p :
$p \in Force(\omega, L)$ si $p \in L$, $p \in Force(\omega, L)$ si $p \in \omega$ et $\neg p \notin L$, $p \notin Force(\omega, L)$ sinon.
Ainsi, $Force(\omega, L)$ est l'interprétation de $\mathbf{PL}$ égale à ω pour les symboles propositionnels de $\mathcal{V}(\mathbf{PL}) - \mathcal{V}(L)$ et satisfaisant les littéraux de L.

Définition 3.2 (Oubli de littéraux) *[2, Prop. 15] Si φ est une formule et L un ensemble de littéraux de $\mathbf{PL}$, $ForgetLit(\varphi, L)$ est une formule ayant pour modèles l'ensemble des interprétations de $\mathbf{PL}$ qui peuvent être transformées en modèle de φ une fois forcées par un sous-ensemble consistant de L :*

$$\mathbf{Mod}(ForgetLit(\varphi, L)) = \{\omega \ / \ Force(\omega, L_1) \models \varphi \text{ et } L_1 \textit{ est un sous-ensemble consistant de } L\}.$$

Cela revient à dire que les modèles de $ForgetLit(\varphi, L)$ sont obtenus à partir des modèles de φ en mettant à “Faux” un nombre arbitraire de valeurs des littéraux de L :

$$\mathbf{Mod}(ForgetLit(\varphi, L)) = \{Force(\omega', L'_1) \ / \ \omega' \models \varphi \text{ et } L'_1 \textit{ est un sous-ensemble consistant de } \neg L\}.$$

Considérons maintenant l'aspect syntaxique. Le plus simple est de commencer par une formule FND :

Proposition 3.3 *[2] Si $\varphi = t_1 \vee \cdots \vee t_n$ est une FND, $ForgetLit(\varphi, L)$ équivaut à la formule $t'_1 \vee \cdots \vee t'_n$ où t'_i est le terme t_i sans les littéraux de L.*

(Là encore, se reporter à la fin de section 4 pour deux exemples.)

La méthode similaire qui fournit $ForgetV(\varphi, V)$ quand φ est une FND a été rappelée au point 1 suivant la définition 2.2. Rappelons également que, pour toute formule φ, $ForgetV(\varphi, V)$ peut être obtenue ainsi (Notations 2.1-1) :

$$ForgetV(\varphi, V) = \bigvee_{V' \subseteq V} \varphi_{[V' \leftarrow 1, \ (V - V') \leftarrow 0]}.$$

De même, on peut donner la définition suivante pour les cas où φ n'est pas une FND :

Définition 3.4 *Si φ est une formule et $L = \{l_1, \cdots, l_n\}$ est une ensemble de littéraux de $\mathbf{PL}$, alors*

$$ForgetLit(\varphi, L) = \bigvee_{L' \subseteq L} \left(\left(\bigwedge \neg L' \right) \wedge \varphi_{\langle (L-L') \leftarrow 1 \rangle} \right).$$

Il s'agit d'une paraphrase de [2, Definition 7]. Nous renvoyons le lecteur à ce texte, qui montre également ceci :

1. Il y a adéquation entre la Définition 3.2 et la Proposition 3.3,
2. Le choix de l'ordre des littéraux ne modifie pas le sens de la formule finale (cf Notations 2.1-3).
3. On a aussi :

$$ForgetLit(\varphi, L) \equiv \bigvee_{L' \subseteq L} \left(\left(\bigwedge \neg L' \right) \wedge \varphi_{\langle (L-L') \leftarrow 1, L' \leftarrow 0 \rangle} \right).$$

La présence de $\left(\bigwedge_{l' \in L'} \neg l' \right)$, qui est ce qui différencie $ForgetLit(\varphi, ...)$ de $ForgetV(\varphi, ...)$, provient du fait qu'ici on veut oublier chaque $l \in L$, mais pas les littéraux complémentaires $l' \in \neg L$.

Une démonstration de [2], utilisant la Proposition 3.3, montre que l'on a $ForgetLit(\varphi, V^{\pm}) \equiv ForgetV(\varphi, V)$ (voir [2, Note 5.2] où la petite faute de frappe [le dernier $l \leftarrow 0$ avant $\equiv$ devrait être $l \leftarrow 1$] est bénigne).
Cette démonstration peut facilement être étendue afin d'obtenir le résultat suivant :

Remarque 3.5 *Puisque tout ensemble L de littéraux peut être écrit comme une union disjointe entre un ensemble consistant L' et un ensemble $V^{\pm}$ de paires de littéraux complémentaires, voici une formulation utile et applicable à tous les ensembles L pour $ForgetLit$:*

$$ForgetLit(\varphi, L' \cup V^{\pm}) \equiv ForgetLit(ForgetV(\varphi, V), L').$$

(Remarquons que L' peut même être inconsistant dans cette équivalence.)

Cette formulation présente l'intérêt, pour ce qui nous concerne ici, de séparer clairement les *littéraux* oubliés et les *symboles propositionnels* oubliés. Soit V' l'ensemble $\mathcal{V}(L')$ des symboles propositionnels figurant dans L', et $V'' = \mathcal{V}(\mathbf{PL}) - V - V'$ l'ensemble des symboles restant. On a alors :

1. *Les symboles propositionnels V sont oubliés*, donc les littéraux de $V^{\pm}$ sont a fortiori oubliés.
2. *Les littéraux de L' sont oubliés*, mais leurs symboles propositionnels (dans V') peuvent demeurer, puisque les littéraux de $\neg L'$ ne sont pas oubliés.
3. *Les littéraux de $V''^{\pm}$ ne sont pas oubliés*, donc *les symboles propositionnels de V'' sont* **fixés**.

Ainsi, $ForgetLit(\varphi, L_1)$ peut-il être défini comme suit : ***oubli de littéraux avec des symboles propositionnels fixés***.
Il est alors tentant et naturel de généraliser la notion, en permettant à certains ***symboles propositionnels*** de ***varier*** lors de l'*oubli des littéraux*.

4 Ajout des symboles variables

Comme pour la notion originale, il est plus facile de présenter les définitions sémantiques d'abord.

Définition 4.1 *Soit φ une formule, V un ensemble de symboles propositionnels, et L un ensemble consistant de littéraux de $\mathbf{PL}$, avec V et $\mathcal{V}(L)$ disjoints dans $\mathcal{V}(\mathbf{PL})$. $ForgetLitVar(\varphi, L, V)$ est une formule ayant l'ensemble de modèles suivant :*

$$\mathbf{Mod}(ForgetLitVar(\varphi, L, V)) = \{\omega \; / \; Force(\omega, L_1 \cup L_2) \models \varphi \textit{ avec } L_1 \subseteq L, L_2 \subseteq V^{\pm}, \; L_2 \textit{ consistant et } (\omega \not\models L_1 \textit{ ou } L_2 = \emptyset)\}.$$

Ceci équivaut à :

$$\mathbf{Mod}(ForgetLitVar(\varphi, L, V)) = \{Force(\omega, L_1 \cup L_2) \; / \; \omega \models \varphi \textit{ avec } L_1 \subseteq \neg L, \; L_2 \subseteq V^{\pm}, \; L_2 \textit{ consistant et } (\omega \not\models L_1 \textit{ ou } L_2 = \emptyset)\}.$$

Comme $\omega \models L_2$ ssi $Force(\omega, L_2) = \omega$, la condition "*$(\omega \not\models L_1$ ou $L_2 = \emptyset)$*" peut être remplacé par "*$(\omega \not\models L_1$ ou $\omega \models L_2)$*", et alors, on peut également remplacer "*L_2 consistant*" par "*L_2 consistant et complet en V*".

On pourrait être plus général, en permettant d'oublier quelques symboles propositionnels, ce qui revient à permettre des ensembles non consistants L. Cette généralisation ne présente pas des difficultés, (il suffit de réduire le vocabulaire, en oubliant les symboles propositionnels concernés par les paires de littéraux complémentaires). Cependant, puisque nous n'avons trouvé aucune application de cette petite généralisation supplémentaire, nous la laissons pour des travaux futurs.
Si on compare avec la définition 3.2, ce qui se produit ici est que la partie non consistante de l'ensemble de littéraux, qui a permis d'oublier un certain ensemble V de symboles propositionnels, a été remplacée par un ensemble de symboles propositionnels autorisés à varier. (Comme écrit juste ci-dessus, ceci n'interdirait pas de permettre des ensembles non consistants L dans la nouvelle définition.)

Remarque 4.2 *Comme on sait d'après [2] qu'on a*

$$ForgetLit(L_1, \varphi) \models ForgetLit(L_1 \cup L_2, \varphi),$$

on a aussi :

$$\varphi \models ForgetV(\varphi, V) \models ForgetLit(\varphi, L \cup V^{\pm}).$$

Il est immédiat de démontrer que l'on obtient également :

$$\varphi \models ForgetLitVar(\varphi, L, V) \models ForgetLit(\varphi, L \cup V^{\pm}).$$

Voici les motivations pour l'introduction de la notion $ForgetLitVar$: nous voulons "oublier" les littéraux de L, même au prix d'une modification des littéraux de $V^{\pm}$. C'est ce qui explique que, si nous oublions effectivement au moins une littéral de L, nous permettons n'importe quelle modification pour les littéraux de $V^{\pm}$. Cependant, nous ne voulons pas modifier les littéraux de $V^{\pm}$ "pour rien" : notre but est d'oublier autant de littéraux de L que possible. Ceci justifie la condition "*($\omega \not\models L_1$ ou $L_2 = \emptyset$)*" dans la définition [et la condition correspondante $\omega \not\models L'_1$ ou $L_2 = \emptyset$ dans la formulation alternative].

L'aspect syntaxique est à peine plus compliqué, et il permet de revoir et d'améliorer des résultats déjà connus. Comme avec la notion originale $ForgetLit$ (voir la proposition 3.3), la manière la plus simple est de partir d'une FND, laquelle est aisément obtenue à partir des modèles.
Sans perte de généralité nous pouvons considérer que L est un ensemble de littéraux négatifs (sinon, remplacer chaque $p \in \mathcal{V}(L)$ tel que $p \in L$ par $\neg p'$, p' étant un nouveau symbole propositionnel, puis, après les calculs, remplacer p' par $\neg p$). Afin de simplifier les notations, à partir de maintenant, nous considérerons donc deux sous-ensembles P et V disjoints de $\mathcal{V}(\mathbf{PL})$, et $L = \neg P$ avec $Q = \mathcal{V}(\mathbf{PL}) - V - P$ désignant l'ensemble des symboles propositionnels restants.

Proposition 4.3 *Soit $\varphi = t_1 \vee \cdots \vee t_n$ une FND, avec*

$$t_i = (\bigwedge P_{i,1}) \wedge (\bigwedge \neg(P_{i,2})) \wedge (\bigwedge V_{i,l}) \wedge (\bigwedge Q_{i,l}),$$

où $P_{i,1} \subseteq P$, $P_{i,2} \subseteq P - P_{i,1}$, et où $V_{i,l}$ et $Q_{i,l}$ sont des ensembles consistants de littéraux de $V^{\pm}$ et $Q^{\pm}$ respectivement. Alors $ForgetLitVar(\varphi, P^-, V) \equiv t'_1 \vee \cdots \vee t'_n$ où

$$t'_i = (\bigwedge P_{i,1}) \wedge (\bigwedge Q_{i,l}) \wedge ((\bigvee (P - P_{i,1})) \vee (\bigwedge V_{i,l})), i.e.$$

$$t'_i = (\bigwedge P_{i,1}) \wedge (\bigwedge Q_{i,l}) \wedge \bigwedge_{l \in V_{i,l}} (l \vee (\bigvee (P - P_{i,1}))).$$

Ainsi, t_i est remplacé par t'_i dans lequel les littéraux de $\neg P$ (c'à-d. les littéraux de $\neg P_{i,2}$) sont supprimés tandis que chaque littéral de $V^{\pm}$ (c'à-d. chaque littéral de $V_{i,l}$) doit figurer maintenant dans un disjonction avec la clause $\bigvee(P - Pi_{,1})$, cette clause dénotant la disjonction de tous les littéraux (positifs) de P^+ qui n'apparaissent pas (positivement) dans t_i. Il est important de remarquer que les littéraux de P^- qui apparaissent dans t_i sont ignorés (ce qui, est normal, puisque on doit les "oublier") mais surtout que les littéraux de $P^{\pm}$ qui comptent (ceux qui restent) sont ceux qui n'apparaissent pas positivement dans t_i (voir "incomplétude (1)" dans la démonstration ci-dessous). En effet, seuls les littéraux positifs (de P^+) peuvent "empêcher d'oublier" un littéral de P^-. Comme ce point est important, on va maintenant ré-écrire ce résultat, en revenant au cas général, où L est quelconque (au lieu de $L = \neg P$).

Soit L un ensemble consistant de littéraux et V un ensemble de symboles propositionnels ne figurant pas dans L. On appelle Q l'ensemble des symbole propositionnels restant : ceux qui ne figurent ni dans L ni dans V.
Alors on a : $ForgetLitVar(\varphi, L, V) \equiv ForgetLitVar(t_1, L, V) \vee \cdots \vee ForgetLitVar(t_n, L, V)$.
Soit t un terme consistant, écrit
$t = (\bigwedge L_1) \wedge (\bigwedge L_2) \wedge (\bigwedge V_l) \wedge (\bigwedge Q_l)$ où

- $L_1 \subseteq \neg L$ désigne l'ensemble des littéraux de t qui sont complémentaires d'un littéral de L,
- $L_2 \subseteq L$ désigne l'ensemble des littéraux de t dans L,
- V_l et Q_l désignent respectivement les ensembles des littéraux de t dans $V^{\pm}$ et $Q^{\pm}$ respectivement.

Alors $ForgetLitVar(t, L, V) \equiv$

$$(\bigwedge L_1) \wedge (\bigwedge Q_l) \wedge ((\bigvee \neg(L - \neg L_1)) \vee (\bigwedge V_l)).$$

La *démonstration de la proposition*, donnée maintenant, peut être omise par le lecteur pressé. (On revient donc au cas $L = \neg P$.)
Considérons d'abord les termes complets, comme

$$t_i = t = (\bigwedge P_1) \wedge (\bigwedge \neg(P - P_1)) \wedge (\bigwedge V_l) \wedge (\bigwedge Q_l),$$

où $P_1 \subseteq P$, et où V_l et Q_l sont des ensembles de littéraux de V et Q respectivement qui sont consistants et complets (contenant tous leurs symboles propositionnels respectifs) et où t correspond à une interprétation ω ($\omega' \models t$ ssi $\omega' = \omega$). L'ensemble

$F(\omega) = \{Force(\omega, L_1 \cup L_2) \ / \ L_1 \subseteq P,\ L_2 \subseteq V^{\pm},$ L_2 consistant et complet en V, et $(\omega \not\models L_1$ ou $\omega \models L_2)\}$

est l'ensemble des modèles de la formule $t^1 \wedge t^2$ où
$t^1 = (\bigwedge P_1) \wedge (\bigwedge Q_l)$ et $t^2 = (\bigvee (P - P_1)) \vee (\bigwedge V_l))$.
En effet, chaque $\omega' \in F(\omega)$ satisfait t^1 puisque t^1 est satisfait dans ω, et les symboles de $P - P_1$ et V peuvent prendre n'importe quelle valeur satisfaisant la condition $\omega \not\models L_1$ ou $\omega \models L_2$. Comme $\omega \models t$, ceci signifie $L_1 \cap (P - P_1) \neq \emptyset$ ou $L_2 \subseteq V_l$, qui équivaut à $\omega' \models t_2$. Réciproquement, tout modèle ω'' de $t_1 \wedge t_2$ est clairement dans $F(\omega)$.

Ce résultat est aussi vrai pour tout terme (consistant) qui n'est plus nécessairement complet, $t = t_i = (\bigwedge P_1) \wedge$

$(\bigwedge\neg(P_2)) \wedge (\bigwedge V_l) \wedge (\bigwedge Q_l)$, où $P_1 \subseteq P$, $P_2 \subseteq P - P_1$, V_l et Q_l étant des sous-ensembles consistants de $V^\pm$ et $Q^\pm$ respectivement : Considérons d'abord séparément les cas dans où quelques symboles de P sont absents, puis des symboles de V, puis des symboles de Q. Il sera ensuite immédiat de combiner ces trois "incomplétudes".

(1) Si $p \in P$ est absent dans t (les littéraux p et $\neg p$ sont absents), pour tout modèle ω' de t, $\omega'' = Force(\omega', \{\neg p\})$ et $Force(\omega'', \{p\})$ sont deux modèles de t (l'un d'eux est ω'). En considérant tous les p manquants, nous obtenons que l'ensemble
$\{Force(\omega', L_1 \cup L_2)\ /\ \omega' \models t,\ L_1 \subseteq \neg P, L_2$ consistant $\subseteq V^\pm, \omega' \not\models L_1$ ou $L_2 = \emptyset\}$ est inclus dans l'ensemble $\{Force(\omega'', L_1 \cup L_2)\ /\ \omega'' \models t \wedge \bigwedge\neg(P - P_1),\ L_1 \subseteq \neg P, L_2$ consistent $\subseteq V^\pm, \omega'' \not\models L_1$ ou $L_2 = \emptyset\}$.
Ainsi, tout p manquant dans t se comporte comme si le littéral négatif $\neg p$ était présent : on peut ajouter "$\wedge\neg p$" à t, obtenant ainsi un terme "complet en P", satisfaisant, $ForgetLitVar(t, P^-, V) \equiv ForgetLitVar(t \wedge \neg(P - P_1), P^-, V)$.

(2) Le raisonnement pour des q manquants dans t ($q \in Q$) est encore plus simple : si $q \in Q$ n'apparaît pas dans t, il peut être interprété par faux ou vrai dans n'importe quel modèle de $ForgetLitVar(t, L, Q)$, ce qui signifie que la partie $\bigwedge Q_l$ reste inchangée, exactement comme dans le cas où Q_l est complet en Q. Les q de Q apparaissant dans Q_l sont "fixés", les autres q de Q peuvent prendre n'importe quelle valeur (le reste étant inchangé).

(3) Le cas des littéraux en V est semblable (la disjonction de toutes les formules avec toutes les possibilités pour les symboles absents donne la formule où ces symboles sont absents) : Si un certain $v \in V$ est absent dans t, alors tout modèle ω' de t a sa contrepartie où la valeur de v est modifiée (v présent s'il étaient absent dans ω', et absent s'il était présent). On obtient ainsi toutes les possibilités pour n'importe quel $v \in V$ manquant dans t. Appelons V_m l'ensemble des symboles de V qui sont absents dans t : $V_m = V - \mathcal{V}(t)$. En considérant les disjonctions de toutes les possibilités, on obtient la formule $(\bigwedge P_1) \wedge (\bigwedge Q_l) \wedge ((\bigvee(P - P_1)) \vee (\bigwedge V_l))$.

En combinant les "trois incomplétudes" (1) – (3), on obtient le résultat suivant :
$ForgetLitVar(t_i, P^-, V) \equiv$
$(\bigwedge P_1) \wedge (\bigwedge Q_l) \wedge ((\bigvee(P - P_1)) \vee (\bigwedge V_l))$.
La disjonction de tous les t_i donne le résultat annoncé. □

Nous venons de fournir la définition sémantique (dans la ligne de la définition 3.2) et une caractérisation à partir d'une formule FND (dans la ligne de la proposition 3.3). Voici maintenant d'autres caractérisations, et une comparaison avec $ForgetLit$.

Proposition 4.4 *Soit φ une formule de $\mathbf{PL}$, et P, Q et V trois ensembles disjoints deux à deux de symboles propositionnels tels que $P \cup Q \cup V = \mathcal{V}(\mathbf{PL})$ ("partition" de $\mathcal{V}(\mathbf{PL})$ autorisant les ensembles vides).*

1. *$ForgetLit(\varphi, P^- \cup V^\pm)$ est équivalent à l'ensemble $Th(\varphi) \cap X$ où X est l'ensemble de toutes les formules de $\mathbf{PL}$ qui sont des disjonctions de termes du type $(\bigwedge P_1) \wedge (\bigwedge Q_l)$ avec $P_1 \subseteq P$ et $Q_l \subseteq Q^\pm$ (nous pouvons clairement ne considérer que les ensembles consistants Q_l).*
2. *$ForgetLitVar(\varphi, P^-, V)$ est équivalent à l'ensemble $Th(\varphi) \cap X$ où X est l'ensemble de toutes les formules de $\mathbf{PL}$ qui sont des disjonctions de termes du type $(\bigwedge P_1) \wedge (\bigwedge Q_l) \wedge \bigwedge_{l \in V_l}(l \vee (\bigvee(P - P_1)))$, où P_1 est inclus dans P, V_l et Q_l étant des ensembles consistants de littéraux de $V^\pm$ et $Q^\pm$ respectivement.*

Ces deux résultats sont des conséquences immédiates des propositions 3.3 et 4.3 respectivement. On peut également choisir les ensembles suivants comme ensembles X : d'abord, on décrit les formules de manière duale, ensuite, on considère des ensembles ayant la même $\wedge$-fermeture que X (cf remarque 2.3) :

Proposition 4.4 (suite)

1. *Pour $ForgetLit(\varphi, P^- \cup V^\pm)$, on peut aussi choisir comme ensemble X :*
   (a) *L'ensemble de toutes les conjonctions des clauses du type $(\bigvee P_1) \vee (\bigvee Q_l)$ avec $P_1 \subseteq P$ et $Q_l \subseteq Q^\pm$ (on peut évidemment ne considérer que les ensembles consistants Q_l).*
   (b) *L'ensemble X de toutes les clauses $(\bigvee P_1) \vee (\bigvee Q_l)$ avec $P_1 \subseteq P$ et $Q_l \subseteq Q^\pm$.*
   (c) *Le plus petit ensemble X possible est l'ensemble de toutes les clauses $(\bigvee P_1) \vee (\bigvee Q_l)$ avec $P_1 \subseteq P$, $Q_l \subseteq Q^\pm$, Q_l consistent et complet en Q.*
2. *Pour $ForgetLitVar(\varphi, P^-, V)$, on peut aussi choisir comme ensemble X :*
   (a) *L'ensemble de toutes les conjonctions des formules $flv(P_1, Q_l, V_l) =$ $(\bigvee P_1) \vee (\bigvee Q_l) \vee \bigvee_{l \in V_l}(l \wedge (\bigwedge(P - P_1)))$, où $P_1 \subseteq P$, V_l et Q_l étant des ensembles consistants de littéraux de $V^\pm$ et $Q^\pm$ respectivement.*
   (b) *L'ensemble de toutes les formules $flv(P_1, Q_l, V_l)$ de ce type.*
   (c) *Le plus petit ensemble X possible est l'ensemble des formules $flv(P_1, Q_l, V_l)$ avec $P_1 \subseteq P$, Q_l et V_l étant des ensembles de littéraux consistants et complets en Q et V respectivement.*

Ces résultats fournissent l'analogue, pour $ForgetLit$ et $ForgetLitVar$, des résultats pour $ForgetV$ rappelés dans la définition 2.2 [$ForgetV(\varphi, V) \equiv Th(\varphi) \cap \mathbf{PL}_{\overline{V}}$], et des résultats qui s'en déduisent rappelés dans la remarque 2.3. La définition suivante, analogue à la définition 3.4, sera notre dernier résultat général au sujet de $ForgetLitVar$:

Définition 4.5 *Si φ est une formule et P et V sont deux sous-ensembles disjoints de $\mathcal{V}(\mathbf{PL})$, alors $ForgetLitVar(\varphi, P^-, V)$ est la formule*

$$\bigvee_{P_1 \subseteq P} \Big(\bigwedge P_1 \wedge (\varphi_{[P_1 \leftarrow 1, (P-P_1) \leftarrow 0]} \vee (ForgetV(\varphi_{[P_1 \leftarrow 1, (P-P_1) \leftarrow 0]}, V) \wedge (\bigvee (P - P_1)))) \Big) .$$

Démonstration de l'adéquation avec la définition 4.1 :
Tout modèle ω de φ donne naissance aux modèles suivants de $ForgetLitVar(\varphi, P^-, V)$:

– ω lui-même qui, avec $P_1 = \omega \cap P$, est modèle de $\psi_1 = \bigwedge P_1 \wedge \bigwedge \neg(P - P_1) \wedge \varphi_{[P_1 \leftarrow 1,\ (P-P_1) \leftarrow 0]}$, ainsi que
– toutes les interprétations différant de ω en ce qu'elles ont au moins un $p \in P$ supplémentaire, et aucune contrainte pour les symboles de V ; cet ensemble d'interprétations étant l'ensemble des modèles de la formule $\psi_2 = \bigwedge P_1 \wedge ForgetV(\varphi_{[P_1 \leftarrow 1,\ (P-P_1) \leftarrow 0]}, V) \wedge \bigvee (P - P_1)$.

Comme on a
$\varphi_{[P_1 \leftarrow 1, (P-P_1) \leftarrow 0]} \models ForgetV(\varphi_{[P_1 \leftarrow 1, (P-P_1) \leftarrow 0]}, V)$ et $\bigwedge \neg(P - P_1) \equiv \neg(\bigvee (P - P_1))$,
quand on considère la disjonction $\psi_1 \vee \psi_2$, on peut supprimer la partie $\wedge \bigwedge \neg(P - P_1)$ dans ψ_1. La disjonction de toutes ces formules $\psi_1 \vee \psi_2$ pour chaque modèle ω de φ, donne la formule écrite dans cette définition. Ainsi, la définition 4.1 est la contrepartie sémantique de la définition 4.5. $\square$
Voici un premier exemple :

Exemple 1 $P = \{a, b\}, V = \{c\}$, $Q = \{d\}$ *avec* $\mathcal{V}(\mathbf{PL}) = P \cup V \cup Q$, *et on considère la formule* $\varphi = (\neg a \wedge b \wedge c) \vee (a \wedge \neg b \wedge \neg c \wedge \neg d)$.
Puisque φ est une FND, les règles pour une FND données pour $ForgetV$, $ForgetLit$ et $ForgetLitVar$ respectivement dans la Section 2 et dans les Propositions 3.3 et 4.3, donnent les trois résultats suivants :

– $ForgetV(\varphi, V) \equiv (\neg a \wedge b) \vee (a \wedge \neg b \wedge \neg d)$.
– $ForgetLit(\varphi, P^- \cup V^\pm) \equiv b \vee (a \wedge \neg d)$.
– $ForgetLitVar(\varphi, P^-, V) \equiv$
$$(a \wedge b) \vee (a \wedge \neg c \wedge \neg d) \vee (b \wedge c). \qquad FLV1$$

On pourrait également utiliser les définitions 3.4 et 4.5, ce qui est fait maintenant pour la définition 4.5.
Dans chaque cas, ψ est la formule $\varphi_{[P_1 \leftarrow 1, (P-P_1) \leftarrow 0]}$:

$$P_1 = \emptyset : \psi \vee (ForgetV(\psi, c) \wedge (a \vee b)) \equiv \bot \vee (\bot \wedge (a \vee b)) \equiv \bot. \qquad (\varphi_1)$$

$$P_1 = \{a\} : a \wedge (\psi \vee (ForgetV(\psi, c) \wedge b)) \equiv a \wedge ((\neg c \wedge \neg d) \vee (\neg d \wedge b)). \qquad (\varphi_2)$$

$$P_1 = \{b\} : b \wedge (\psi \vee (ForgetV(\psi, c) \wedge a)) \equiv b \wedge (c \vee (\top \wedge a)). \qquad (\varphi_3)$$

$$P_1 = \{a, b\} : a \wedge b \wedge (\psi \vee (ForgetV(\psi, c) \wedge \bot)) \equiv a \wedge b \wedge (\bot \vee \bot) \equiv \bot. \qquad (\varphi_4)$$

La disjonction $\bigvee_{i=1}^4 \varphi_i$ est $(a \wedge \neg c \wedge \neg d) \vee (a \wedge \neg d \wedge b) \vee (b \wedge c) \vee (b \wedge a)$, ce qui est bien équivalent à $FLV1$.
Utilisons maintenant les résultats concernant la sémantique. On a $\mathbf{Mod}(\varphi) = \{\mu_1, \mu_2, \mu_3\}$ *avec*
$\mu_1 = \{a\}, \mu_2 = \{b, c\}, \mu_3 = \{b, c, d\}$.

– *Les six modèles de $ForgetV(\varphi, V)$ sont obtenus en ajoutant les trois interprétations différant des trois modèles de φ par la valeur attribuée à c :*
$\mu'_1 = \{a, c\}, \mu'_2 = \{b\}, \mu'_3 = \{b, d\}$.
– *Les dix modèles de $ForgetLit(\varphi, P^- \cup V^\pm)$ sont obtenus en ajoutant aux modèles de φ les sept interprétations différant de ces modèles en ajoutant n'importe quel sous-ensemble de $\{a, b\}$ et aussi, soit en ne faisant rien, soit en modifiant la valeur de c (ajoutant c s'il n'est pas présent et l'enlevant s'il est présent). Ceci donne les modèles de $ForgetV(\varphi, V)$ plus les quatre interprétations qui incluent $\{a, b\}$.*
– *Les sept modèles de $ForgetLitVar(\varphi, P^-, V)$ sont obtenus en ajoutant aux modèles de φ les quatre interprétations différant de ces modèles en ajoutant un sous-ensemble* non vide *de $\{a, b\}$ et, soit en ne faisant rien d'autre, soit en modifiant la valeur de c, ce qui donne ici les quatre interprétations contenant $\{a, b\}$.*

Remarquons qu'une fois que nous avons tous les modèles de φ, la complexité de la construction de tous les modèles de $ForgetLitVar(\varphi, P^-, V)$ n'est pas plus grande que la complexité de la construction de tous les modèles de $ForgetLit(\varphi, P^- \cup V^\pm)$.

Remarquons aussi que sur cet exemple nous obtenons (cf remarque 4.2) :
– $ForgetV(\varphi, V) \vee ForgetLitVar(\varphi, P^-, V) \equiv ForgetLit(\varphi, P^- \cup V^\pm)$
– $ForgetV(\varphi, V) \wedge ForgetLitVar(\varphi, P^-, V) \equiv \varphi$.

Il est facile de vérifier que, si le premier résultat est général, ce n'est pas le cas du second :

Exemple 2 $P, V, Q, \mathbf{PL}$ *comme en exemple 1,* $\varphi = t = a \wedge c$. *On a ici un terme non complet en P. On obtient :*
– $ForgetV(t, V) \equiv a$.
– $ForgetLit(t, P^- \cup V^\pm) \equiv a$.
– $ForgetLitVar(t, P^-, V) \equiv$
$ForgetLitVar(a \wedge \neg b \wedge c, P^-, V) \equiv a \wedge (b \vee c)$.

(Ré)-examinons maintenant une application de l'oubli de littéraux

5 Application à la circonscription

L'oubli de littéraux a de fortes connexions avec la circonscription, un formalisme bien connu en représentation des connaissances [6]. La circonscription minimise les "exceptions", et elle est utilisée dans des langages d'action et d'autres formalisations du raisonnement de sens commun. Un point clé est le calcul efficace, et donc toute amélioration est utile. Rappelons la définition de la circonscription propositionnelle.

Définition 5.1 $\mathcal{V}(\mathbf{PL}) = P \cup V \cup Q$ *(ensembles disjoints). Les symboles propositionnels de* P, V, Q *sont respectivement* circonscrits, variables, *et* fixes. *On définit la relation* $\prec_{(P,Q,V)}$ *dans* **Mod** *ainsi*
$\mu \prec_{(P,Q,V)} \nu$ *si* $P \cap \mu \subset P \cap \nu$ *et* $Q \cap \mu = Q \cap \nu$
($\subset$ *strict, aucune condition sur* V*).*
Un modèle μ *de* φ *est* minimal pour $\prec_{(P,Q,V)}$ *si aucun modèle* ν *de* φ *ne satisfait* $\nu \prec \mu$.
La circonscription $CIRC(P,Q,V)$ *est l'application* $\mathbf{PL} \hookrightarrow \mathcal{P}(\mathbf{PL})$ *définie ainsi : pour tout* φ, ψ *de* $\mathbf{PL}$, $\psi \in CIRC(P,Q,V)(\varphi)$ *si tout modèle de* φ *minimal pour* $\prec_{(P,Q,V)}$ *est un modèle de* ψ.

Il s'agit de la définition sémantique de la circonscription [3], dans le cas propositionnel comme donné dans [8]. L'ensemble $CIRC(P,Q,V)(\varphi)$ est équivalent à la formule ayant pour modèles les modèles de φ minimaux pour $\prec_{(P,Q,V)}$, et donc $CIRC(P,Q,V)(\varphi)$ sera parfois assimilé à cette formule.

On a le résultat suivant :

Théorème 5.2 $Circ(P,Q,V)(\varphi) \models \psi$ *ssi*
$$\varphi \models ForgetLitVar(\varphi \wedge \psi, P^-, V).$$

Il s'agit en fait d'une réécriture d'un résultat déjà connu. Quelques indications techniques sont utiles afin de relier ce résultat à la littérature. D'abord, ce résultat améliore le résultat suivant :

Proposition 5.3 *[2, Proposition 22]*

1. *Si* φ *ne contient pas de symboles propositionnels de* V*, alors* $Circ(P,Q,V)(\varphi) \models \psi$ *ssi*
$$\varphi \models ForgetLit(\varphi \wedge \psi, P^- \cup V^\pm).$$
2. *Dans le cas général,* $Circ(P,Q,V)(\varphi) \models \psi$ *ssi*

$$\varphi \models ForgetLit(\varphi \wedge \neg ForgetLit(\varphi \wedge \neg\psi, P^- \cup V^\pm), P^- \cup V^\pm).$$

Le point 1 concerne la circonscription sans symbole variable puisque, si ψ n'a aucun symbole dans V,
$Circ(P,Q,V)(\varphi) \models \psi$ ssi
$Circ(P,Q,\emptyset)(ForgetV(\varphi,V)) \models \psi$

Le second point est en deux étapes, afin de se ramener au cas du premier point. Ces deux étapes augmentent sensiblement la complexité. Comme montré dans [2], ce résultat est très proche d'un résultat plus ancien :

Proposition 5.4 *[9, Theorems 2.5 – 2.6]*

1. *[Theorem 2.5] Si* ψ *est sans symbole de* V*, alors* $Circ(P,Q,V)(\varphi) \models \psi$ *ssi pour toute clause* c *sans littéral de* $P^- \cup V^\pm$*,* $\varphi \wedge \psi \models c$ *implique* $\varphi \models c$.
2. *[Theorem 2.6]* $Circ(P,Q,V)(\varphi) \models \psi$ *ssi il existe une formule* $\gamma = ForgetLit(\gamma', P^- \cup V^\pm)$ *pour quelque formule* γ'*, telle que* $CIRC(P,Q,V)(\varphi) \models \neg\gamma$ *et* $\varphi \models \gamma \vee \psi$.

Là aussi, le deuxième point est une méthode en deux étapes, ce qui augmente sa complexité : il faut examiner toutes les formules γ dans un ensemble assez grand, et vérifier grâce au premier point si $\neg\gamma$ est conséquence de la circonscription ou pas. Le premier point est là encore en fait limité aux circonscriptions sans symbole propositionnel variable.
Voici maintenant un autre formalisme non monotone, qui généralise les formalismes implicitement utilisés dans les premiers points des propositions 5.3 et 5.4.

Définition 5.5 *[10] Une* X-logique *est une application* $f_X : \mathbf{PL} \hookrightarrow \mathcal{P}(\mathbf{PL})$ *d'un ensemble* X *de formules dans* $\mathbf{PL}$ *définie ainsi :*

$$\varphi \in f_X(\psi) \quad \textit{ssi} \quad Th(\psi \wedge \varphi) \cap X \subseteq Th(\psi).$$

Cette notion a été introduite afin de faciliter le calcul d'un formalisme non-monotone. En fait, cette notion est apparue avant le papier référencé, par exemple dans [12] et même dans des papiers antérieurs au sujet du calcul des formalismes de minimisation. La nouveauté de [10] était une identification claire du formalisme considéré, en particulier il était signalé que, contrairement à la circonscription par exemple, on n'obtient pas toujours une théorie : l'ensemble $f_X(\varphi)$ n'est généralement pas fermé pour $\wedge$, ainsi $f_X(\varphi)$ dans le cas général (fini) ne peut pas être assimilé à une formule.
D'après la remarque 2.3, on peut remplacer l'ensemble X par n'importe quel ensemble ayant la même $\wedge$-fermeture.

Soit f une application $\mathbf{PL} \hookrightarrow \mathcal{P}(\mathbf{PL})$. L'ensemble des formules *inaccessibles pour* f est l'ensemble des formules ψ pour lesquelles il n'existe aucune formule φ telle que $\psi \in f(\varphi)$ et $\varphi \not\models \psi$. Il est connu (et facile à vérifier) que pour toute X-logique f_X, l'ensemble des formules inaccessibles pour f_X est (à l'équivalence logique près), la $\wedge$-fermeture de X.

Les points 1 des propositions 5.3 et 5.4 peuvent s'écrire respectivement comme suit, où $\equiv_{\overline{V}}$ signifie "équivalents sur l'ensemble des formules sans symbole de V" :

– $Circ(P,Q,V)(\varphi) \equiv_{\overline{V}} f_X(\varphi)$ où X est l'ensemble indiqué dans la proposition 4.4-1 [ou n'importe lequel des ensembles décrits dans la proposition 4.4 (suite)-1].
– $Circ(P,Q,V)(\varphi) \equiv_{\overline{V}} f_X(\varphi)$ où X est l'ensemble de toutes les clauses ayant leurs littéraux dans $P^+ \cup Q^\pm$.

Il est clair, d'après la remarque 2.3 et la proposition 4.4 que ces deux "points 1" sont en quelque sorte équivalents, comme l'a montré [2].
On va montrer maintenant que, dans la proposition 5.4 (les mêmes considérations s'appliquent à la proposition 5.3), le point 2 est (presque) une conséquence directe du point 1, en utilisant un vieux résultat au sujet de l'élimination des symboles variables dans la circonscription [3, Proposition 3.2.1] (d'abord paru en 1985) :

$$\begin{array}{l} Circ(P,Q,V)(\varphi) \models \psi \quad \text{ssi} \\ Circ(P,Q,\emptyset)(ForgetV(\varphi,V)) \cup \{\varphi\} \models \psi \end{array} \qquad (\mathrm{Eq}_L)$$

En effet, grâce à (Eq_L), on obtient $Circ(P,Q,V)(\varphi) \models \psi$ ssi il existe une formule $\neg\gamma$ sans symbole dans V telle que $Circ(P,Q,\emptyset)(ForgetV(\varphi,V)) \models \neg\gamma$ et $\varphi \wedge \neg\gamma \models \psi$. Puisque $\neg\gamma$ est sans symbole de V, cela équivaut à : $Circ(P,Q,V)(\varphi) \models \neg\gamma$ et $\varphi \models \gamma \vee \psi$. La différence (qui justifie le "presque" écrit ci-dessus) est que Przymusinski a amélioré un peu ce résultat en limitant l'ensemble des formules disponibles pour γ, puisque P^-peut être supprimé ("oublié") également, et pas seulement V.

Les connexions entre la X-logique et la circonscription sont maintenant bien connues [12, 10, 7]. Voici d'abord quelques définitions techniques.

Définition 5.6 *On appelle* positive en $\prec_{(p,Q,V)}$ *une formule φ dont l'ensemble des modèles est "finissant" pour $\prec_{(p,Q,V)}$: si $\mu \models \varphi$ et $\mu \prec_{(P,Q,V)} \nu$, alors $\nu \models \varphi$.*

La définition 4.1 montre que $ForgetLitVar(\varphi,P,V)$ est positive en $\prec_{(P,Q,V)}$.
La "réciproque" est d'ailleurs vraie aussi, puisque toute formule φ positive en $\prec_{(P,Q,V)}$ est équivalente à $ForgetLitVar(\varphi,P,V)$.

Exemple 3 (Exemple 1 suite)
Avec $\mathbf{PL}$, P,V,Q *et* φ *comme dans l'exemple 1, on a*
$\mathbf{Mod}(ForgetLitVar(\varphi,P^-,V)) = \{\nu \in \mathbf{Mod}\,/$
$(\mu \prec_{(P,Q,V)} \nu$ *ou* $\mu = \nu)$ *pour quelque* $\mu \in \mathbf{Mod}(\varphi)\}$
Cet ensemble est clairement finissant pour $\prec_{(P,Q,V)}$.

Ces indications techniques permettraient facilement d'obtenir une démonstration directe du théorème 5.2. Cependant, une nouvelle démonstration n'est pas nécessaire puisque le théorème 5.2 est une conséquence immédiate de la proposition 4.4 ainsi que du résultat déjà connu suivant :

Proposition 5.7 *[8, Theorem 6.40] $Circ(P,Q,V) = f_X$ pour tout ensemble X qui a (à l'équivalence logique près) comme $\wedge$-fermeture l'ensemble des formules positives en $\prec_{(P,Q,V)}$.*

Voici un exemple afin d'illustrer le fonctionnement pratique du théorème 5.2.

Exemple 4 *P,Q,V sont comme dans l'exemple 1.*
$\varphi' = (b \wedge c) \vee (a \wedge \neg b \wedge \neg c \wedge \neg d)$ *et*
$\psi' = (\neg a \wedge b) \vee (a \wedge \neg b \wedge \neg c \wedge \neg d) \vee (\neg a \wedge c \wedge d)$.
$\varphi' \wedge \psi'$ équivaut au φ de l'exemple 1.
Comme $\varphi' \models (ForgetLitVar(\varphi,P^-,V))$, on en conclut $Circ(P,Q,V)(\varphi') \models \psi'$.

L'approche $ForgetLitVar$ fournit de nouvelles méthodes pour le calcul effectif des circonscriptions : En particulier, elle fournit un processus simple en une seule étape, au lieu des deux étapes des propositions 5.3 et du 5.4. Elle fournit également une manière constructive pour calculer la circonscription grâce à la proposition 5.7, fournissant une concrétisation précise à la méthode de calcul des circonscriptions à l'aide de leurs formules inaccessibles qui a été décrite dans [7, 8].

6 Conclusion et perspectives

Nous avons donné plusieurs caractérisations sémantiques et syntaxiques pour une nouvelle notion, étendant la notion d'oubli de littéraux présentée dans [2] aux cas où des symboles propositionnels sont autorisés à varier. Ces résultats prouvent que la nouvelle notion n'est pas sensiblement plus difficile que l'oubli de littéraux sans symboles variables. Les diverses caractérisations fournissent diverses façons de calculer les résultats, plus ou moins efficaces selon la forme dans laquelle les formules de départ apparaissent.
Nous avons ensuite développé l'application au calcul de la circonscription. Nous avons ainsi renforcé les connexions (déjà trouvées dans [2]) entre la méthode d'oubli de littéraux et une méthode plus ancienne de Przymunsinski [9]. Cela a permis de transformer les méthodes en deux étapes qui étaient connues, par l'oubli explicite de littéraux, ou grâce au théorème de Przymunsinski, en une méthode plus simple en une seule étape. Nous avons également démontré que la méthode d'oubli de littéraux (ainsi que la méthode de Przymunsinski) est fortement liée à la méthode des X-logiques. À notre connaissance, ce lien n'avait pas été réalisé auparavant, ce qui explique sans doute pourquoi la méthode de Przymunsinski n'a pas été simplifiée plus tôt. (Il faut noter que [9] cite [1] qui peut être considéré, pour certains de ses aspects, comme travail très préliminaire sur ce qui est maintenant connu sous le nom de X-logique, mais il reste que le lien précis n'avait été jamais fait, à notre connaissance.)

Ces liens fournissent une présentation plus simple de divers résultats, y compris la simplification (évoquée ci-dessus) de la méthode de Przymunsinski pour calculer la circonscription.

La notion de "formule inaccessible" d'une inférence non classique (qui englobe l'inférence classique) a une signification claire du point de vue de la représentation des connaissances. Ce sont les formules qui ne peuvent pas être obtenues comme un résultat nouveau (nouveau vis à vis de l'inférence classique) par l'inférence considérée. La description donnée ici de la méthode de calcul de la circonscription à partir de ses formules inaccessibles utilise la notion d'"oubli de littéraux" généralisée au cas de symboles variables. On peut considérer alors que la notion de X-logique permet d'étendre encore la notion d' "oubli" au cas d' "oubli des formules". Il reste à préciser cette notion d'"oubli de formules" et les méthodes de calcul associées.

Il reste en particulier à voir si la méthode donnée ici peut se généraliser, et comment, afin de fournir de nouvelles méthodes constructives et relativement efficaces de calculs de certaines autres X-logiques. Un des intérêts de cette généralisation est qu'on atteindrait alors peut-être aussi des formalismes qui falsifient la règle du ET (ou AND), c'est-à-dire (en utilisant le vocabulaire de la logique des défauts), qui admettent plusieurs extensions.

D'autres applications, comme celles aux formalismes de représentation et traitement des connaissances évoqués [2, p. 413] (dont l'abduction) pourraient sans doute tirer profit de cette généralisation aussi, mais il s'agit d'un point que nous n'avons pas abordé.

Il reste que l'un des intérêts de la méthode d'oubli des littéraux, qui est, dans certains cas favorables, une simplification des calculs effectifs [2], ne peut que tirer profit de l'extension de la notion d'oubli développée ici. La complexité de la définition sémantique de la nouvelle notion est semblable à celle de la notion sans symbole variable. Le cas des définitions syntaxiques reste à étudier de ce point de vue, mais il ne semble pas que la complexité augmente de façon trop importante.

Remerciements

L'auteur tient à remercier les relecteurs pour leurs commentaires et indications pertinents, avec un grand merci à celui qui a suggéré une meilleure présentation d'une des définitions principales.

Références

[1] G. Bossu and P. Siegel. Saturation, nonmonotonic reasoning and the closed-world assumption. *Artificial Intelligence*, 25(1) :13–63, Jan. 1985.

[2] J. Lang, P. Liberatore, and P. Marquis. Propositional Independence - Formula-Variable Independence and Forgetting. *(Electronic) Journal of Artificial Intelligence Research*, 18 :391–443, 2003. `http ://WWW.JAIR.ORG/`.

[3] V. Lifschitz. Circumscription. In D. M. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3 : Non-Monotonic and Uncertainty Reasoning*, pages 297–352. Oxford University Press, 1994.

[4] F. Lin. On strongest necessary and weakest sufficient conditions. *Artficial Intelligence*, 128(1–2) :143–159, May 2001.

[5] F. Lin and R. Reiter. Forget it ! In C. S. Mellish, editor, *AAAI Fall Symposium on Relevance*, pages 1985–1991, New Orleans, USA, Nov. 1994. Morgan Kaufmann.

[6] J. McCarthy. Application of circumscription to formalizing common sense knowledge. *Artificial Intelligence*, 28(1) :89–116, Feb. 1986.

[7] Y. Moinard and R. Rolland. Circumscriptions from what they cannot do (Preliminary report). In *Common Sense'98*, pages 20–41, London, Jan. 1998. `http ://www.ida.liu.se/ext/etai/nj/fcs-98/listing.html`.

[8] Y. Moinard and R. Rolland. Propositional circumscriptions. Technical report, INRIA, Research Report RR-3538 (ou IRISA, Publication Interne 1211) Rennes, France, Oct. 1998.

[9] T. C. Przymusinski. An Algorithm to Compute Circumscription. *Artificial Intelligence*, 38(1) :49–73, Feb. 1989.

[10] P. Siegel and L. Forget. A representation theorem for preferential logics. In L. C. Aiello, J. Doyle, and S. Shapiro, editors, *KR'96*, pages 453–460, Cambridge, Nov. 1996. Morgan Kaufmann.

[11] K. Su, G. Lv, and Y. Zhang. Reasoning about Knowledge by Variable Forgetting. In D. Dubois, C. A. Welty, and M.-A. Williams, editors, *KR'04*, pages 576–586. AAAI Press, 2004.

[12] M. A. Suchenek. First-Order Syntactic Characterizations of Minimal Entailment, Domain-Minimal Entailment, and Herbrand Entailment. *Journal of Automated reasoning*, 10 :237–263, 1993.