

Fusion de bases propositionnelles : une méthode basée sur les R-ensembles

Hue Julien[†] Wurbel Eric[†] Papini Odile[‡]
hue@univ-tln.fr wurbel@univ-tln.fr odile.papini@esil.univmed.fr

[†]LSIS UMR CNRS 6168 - Équipe INCA
Université du Sud Toulon-Var
Avenue de l'Université - BP20132
83957 LA GARDE CEDEX - FRANCE

[‡]LSIS UMR CNRS 6168 - Équipe INCA
Université de la Méditerranée - ESIL Avenue de Luminy
13288 MARSEILLE CEDEX - FRANCE

Résumé :

La prise de décision collective conduit à l'interaction de plusieurs agents afin d'élaborer une décision commune cohérente. D'un point de vue informatique, ce problème peut se ramener à celui de la fusion de différentes sources d'informations. Dans le domaine de la représentation des connaissances pour l'intelligence artificielle, plusieurs approches ont été proposées pour la fusion de bases de croyances propositionnelles, cependant, la plupart d'entre elles l'ont été sur un plan sémantique et sont peu utilisables en pratique. Ce papier propose une nouvelle approche syntaxique pour la fusion de bases de croyances, appelée Fusion par R-ensembles (ou RSF). La notion de R-ensemble, initialement définie dans le contexte de la révision de croyances, est étendue à la fusion et la plupart des opérations classiques de fusion sont capturées syntaxiquement par RSF. Afin d'implanter efficacement RSF, ce papier montre comment RSF peut être codé en un programme logique avec sémantique des modèles stables, puis présente une adaptation du système Smodels permettant de calculer efficacement les R-ensembles. Finalement, une étude expérimentale préliminaire montre que la mise en œuvre utilisant la programmation logique avec sémantique des modèles stables semble prometteuse pour réaliser la fusion de bases de croyances sur des applications réelles.

Mots-clés : Fusion de croyances, Raisonnement sur les croyances, Représentation des connaissances

Abstract:

Collective decision making leads to interaction between agents in order to elaborate a consistent common decision. From a data-processing point of view, this problem can be brought back to the merging of different sources of information. In knowledge representation for artificial intelligence, several approaches have been proposed for propositional bases fusion, however, most of them are de-

finied at a semantic level and are untractable. This paper proposes a new syntactic approach of belief bases fusion, called Removed Sets Fusion (RSF). The notion of removed-set, initially defined in the context of belief revision is extended to fusion and most of the classical fusion operations are syntactically captured by RSF. In order to efficiently implement RSF, the paper shows how RSF can be encoded into a logic program with answer set semantics, then presents an adaptation of the smodels system devoted to efficiently compute the removed sets in order to perform RSF. Finally a preliminary experimental study shows that the answer set programming approach seems promising for performing belief bases fusion on real scale applications.

Keywords: Belief merging, reasoning about beliefs, Knowledge representation

1 Introduction

Dans le contexte de la prise de décision collective, plusieurs experts ou agents intelligents sont amenés à interagir afin d'élaborer une décision commune de façon rationnelle. La prise de décision collective a été étudiée dans le domaine de la théorie du choix social et de récents travaux [28], [26], [15] ont montré le lien entre ces travaux et ceux développés dans le domaine de l'intelligence artificielle, en particulier concernant la fusion. La fusion d'informations issues de différentes sources est un problème important dans plusieurs domaines de l'informatique comme la représentation des connaissances pour l'intelligence artificielle, la prise de décision ou les bases de données. Le but de la fu-

sion est d'obtenir un point de vue global exploitant la complémentarité des différentes sources, résolvant les conflits possibles et enlevant les redondances existantes entre les sources. Parmi les différentes approches de la fusion d'informations provenant de sources multiples, les approches logiques ont obtenues un intérêt croissant ces dix dernières années [1, 23, 16, 24, 5]. La plupart de ces approches ont été définies dans le cadre de la logique classique, le plus souvent propositionnelle, et ont été définies sémantiquement. Différents postulats caractérisant le comportement rationnel des opérateurs de fusion ont été proposés [12] et plusieurs opérateurs ont été définis selon que des priorités (implicites ou explicites) sont prises en compte ou non [14], [13], [7], [25], [19]. Plus récemment, de nouvelles approches ont été proposées comme la fusion sémantique de bases propositionnelles, notamment à partir de la distance de Hamming [11] ou la fusion syntaxique dans le cadre possibiliste [8, 2] qui est un véritable avantage du point de vue de l'efficacité calculatoire.

Ce papier propose une nouvelle approche pour la fusion syntaxique de bases de croyances propositionnelles. Nous montrons que les opérateurs classiques de fusion, *Card*, Σ , *Max*, *Gmax*, initialement définis au niveau sémantique, peuvent être capturés dans notre cadre syntaxique. Nous montrons ensuite qu'une implantation efficace de ces opérateurs, basée sur la programmation logique avec sémantique des modèles stables, peut être réalisée. En particulier, ce papier se concentre sur les trois points suivants :

- Nous étendons la Revision par R-ensembles (RSR¹) à la fusion de bases de croyances propositionnelles, que nous appelons Fusion par R-ensembles (RSF²). Nous montrons comment la notion de R-ensemble, c'est à dire les sous-

ensembles de clauses à retirer pour restaurer la cohérence, initialement définie dans le contexte de la révision de bases de croyances [21, 29] est généralisée au cas de fusion de bases de croyances. Nous montrons ensuite comment les opérateurs classiques de fusion sont capturés dans ce cadre en associant à chaque stratégie de fusion une relation de préférence entre les sous-ensembles de clauses.

- Ces dix dernières années, la programmation logique avec sémantique des modèles stables est apparue comme étant un outil efficace pour manipuler des systèmes de raisonnement non-monotone. De plus, plusieurs systèmes efficaces ont été développés [9], [4], [22], [20], [18]. Nous proposons de formaliser la Fusion par R-ensembles dans le cadre de la programmation logique avec sémantique des modèles stables et d'adapter le système *Smodels* pour calculer les modèles stables préférés qui correspondent aux R-ensembles ce qui permet de définir une méthode effective de fusion.
- Une étude expérimentale préliminaire permet d'illustrer le comportement de RSF pour les stratégies *Card* et Σ qui semble prometteur pour réaliser la fusion sur des applications réelles.

Ce papier est organisé comme suit. La section 2 fixe les notations et donne un rappel sur la fusion, la Revision par R-ensembles et la programmation logique avec sémantique des modèles stables. La section 3 présente ensuite la Fusion par R-ensembles. La section 4 montre comment la Fusion par R-ensembles est mise en œuvre dans la programmation logique avec sémantique des modèles stables et présente une adaptation du système *Smodels* pour le calcul des modèles stables préférés et la réalisation de la Fusion par R-ensembles. La section 5 présente ensuite une première étude expérimentale qui montre que cette implantation, grâce à la programmation logique avec sémantique des modèles stables, semble prometteuse

¹RSR : Removed Set Revision en anglais

²RSF : Removed Set Fusion en anglais

avant de conclure en section 6.

2 Préliminaires et notations

On considère un langage propositionnel \mathcal{L} défini sur un alphabet \mathcal{P} d'atomes. Un littéral est un atome ou la négation de cet atome. Les connecteurs usuels de la logique propositionnelle sont notés : \neg , \wedge , \vee et on note Cn la conséquence logique. Une base de croyances K est un ensemble fini de formules propositionnelles définies sur le langage propositionnel \mathcal{L} .

2.1 Fusion

Soit $E = \{K_1, \dots, K_n\}$ un multi-ensemble de n bases de croyances cohérentes à fusionner, E est appelé ensemble de croyances. Les n bases de croyances K_1, \dots, K_n ne sont pas nécessairement différentes et l'union des bases de croyances prenant en compte la répétition est notée \sqcup ; la conjonction et la disjonction sont respectivement notées \wedge et \vee . Pour simplifier les notations, on note K l'ensemble de croyances constitué du singleton $E = \{K\}$.

L'opération de fusion Δ est définie comme étant une fonction qui, à chaque ensemble de croyances, associe une base de croyances cohérente, notée $\Delta(E)$. Dans la littérature, $\Delta(E)$ est généralement définie de deux manières différentes : en tenant compte ou pas de l'existence d'une priorité implicite. Ce papier ne tient pas compte des priorités implicites.

Il y a deux façons immédiates de définir $\Delta(E)$ selon que les sources sont conflictuelles ou pas, la conjonction classique : $\Delta(E) = \bigwedge_{K_i \in E} K_i$ dans le cas où les sources ne sont pas contradictoires et la disjonction classique : $\Delta(E) = \bigvee_{K_i \in E} K_i$ dans le cas de sources conflictuelles. Entre ces deux cas opposés, plusieurs méthodes ont été proposées suivant l'importance des différentes bases.

En particulier, les opérateurs classiques de fusion suivants ont été proposés. L'opérateur de cardinalité, noté *Card* [1], tient en compte du nombre de bases de croyances de E . L'opérateur Somme, noté Σ [17, 23], suit le point de vue de la majorité. L'opérateur *Max* [24] essaie de satisfaire au mieux toutes les bases de E . L'opérateur *Gmax* [12], qui est un raffinement de l'opérateur *Max*.

Différents postulats permettant de caractériser le comportement rationnel des opérateurs de fusion ont été proposés [12] et ces opérateurs ont été classés en deux familles : les opérateurs majoritaires et les opérateurs d'arbitrage.

2.2 La programmation logique avec sémantique des modèles stables

Un programme logique normal P est un ensemble de règles de la forme $c \leftarrow a_1, \dots, a_n, \text{not } b_1, \dots, \text{not } b_m$ où les $c, a_i (1 \leq i \leq n), b_j (1 \leq j \leq m)$ sont des atomes et le symbole *not* représente la négation par échec. Pour une règle r comme ci-dessus, on introduit les notations $tête(r) = c$ et $corps(r) = \{a_1, \dots, a_n, b_1, \dots, b_m\}$. De plus, $corps^+(r) = \{a_1, \dots, a_n\}$ représente l'ensemble des atomes positifs présents dans le corps et $corps^-(r) = \{b_1, \dots, b_m\}$ en représente l'ensemble des atomes négatifs. Enfin, $corps(r) = corps^+(r) \cup corps^-(r)$.

Soit r une règle, r^+ représente la règle $tête(r) \leftarrow corps^+(r)$, que l'on obtient à partir de r en supprimant tous les atomes négatifs dans le corps de r .

Un ensemble d'atomes X est clos sous un programme P ssi $\forall r \in P, tête(r) \in X$ lorsque $corps(r) \subseteq X$. Le plus petit ensemble d'atomes qui est clos sous un programme P est noté $CN(P)$.

La réduction, ou transformation de

Gelfond-Lifschitz [10], P^X , est un programme P qui est défini, relativement à un ensemble X d'atomes, par : $P^X = \{r^+ \mid r \in P \text{ et } \text{corps}^-(r) \cap X = \emptyset\}$.

Un ensemble d'atomes X est un modèle stable de P ssi $CN(P^X) = X$.

Définition 1. Soit L un ensemble de littéraux et A un ensemble d'atomes. On dit que L couvre A ssi $A \subseteq \text{Atome}(L)$.

L'exemple suivant permet d'illustrer le concept de modèle stable.

exemple. Les modèles stables sont les conséquences que l'on peut tirer du programme logique. Soit P le programme logique constitué des trois règles suivantes :

$$p \quad f \leftarrow p, \text{not } h \quad h \leftarrow p, \text{not } f$$

Pour le programme précédent, les modèles stables du programme P sont $\{p, h\}$ et $\{p, f\}$. Contrairement à la programmation logique standard, comme par exemple, PROLOG, où on dispose d'un seul ensemble de conséquences, appelé modèle minimal, (intersection des modèles de Herbrand associés au programme) la programmation logique avec sémantique des modèles stables fournit plusieurs ensembles de conséquences qui peuvent être contradictoires et permet de formaliser le raisonnement non monotone.

2.3 Smodels

Smodels est la première et la plus simple des méthodes de calcul de modèles stables [27]. C'est un algorithme de Branch and Bound (voir Algorithme 1) qui construit, au fur et à mesure, un ensemble d'atomes A représentant un modèle stable potentiel. Pour ce faire, il utilise les fonctions suivantes : $expand(A)$ qui calcule les conséquences immédiates de A , $conflict(A)$ qui détecte les conflits qui peuvent se produire et $heuristic(A)$ qui tente de réduire

la taille de l'espace de recherche en choisissant l'atome qui permet le plus de déductions. La fonction $heuristic(A)$ tente de réduire le nombre d'atomes qu'il reste à choisir et permet une détection plus rapide des conflits.

Algorithme 1 $smodels(A)$

```

A ← expand(A)
si conflict(A) alors
  renvoyer Faux
sinon si A couvre Atome(E) alors
  renvoyer Vrai
sinon
  x ← heuristic(A)
  si smodels(A ∪ {x}) alors
    renvoyer Vrai
  sinon
    renvoyer smodels(A ∪ {not x})
  fin si
fin si

```

2.4 Revision par R-ensembles

Nous rappelons brièvement l'approche Revision par R-ensembles (RSR). L'approche RSR [29] traite de la révision d'un ensemble de formules propositionnelles par un ensemble de formules propositionnelles³. Soient K et A deux ensembles finis de clauses. L'approche RSR consiste à choisir un ensemble minimal de clauses à retirer de K , appelé R-ensemble [21], afin de restaurer la cohérence de $K \cup A$. Plus formellement :

Définition 2. Soient K et A deux ensembles de clauses cohérents tels que $K \cup A$ est incohérent. R , un sous-ensemble de clauses de K , est un R-ensemble de $K \cup A$ ssi (i) $(K \setminus R) \cup A$ est cohérent ; (ii) $\forall R' \subseteq K$, si $(K \setminus R') \cup A$ est cohérent, alors $|R| \leq |R'|$ ⁴.

³ À partir de maintenant, on considère les formules propositionnelles dans leurs formes normales conjonctives (CNF) associées.

⁴ $|R|$ représente le nombre de clauses de R .

On note $\mathcal{R}(K \cup A)$ la collection des R-ensembles de $K \cup A$, la Revision par R-ensembles est définie comme suit :

Définition 3. Soient K et A deux ensembles de clauses cohérents. La Revision par R-ensembles est définie par : $K \circ_{RSR} A =_{def} \bigvee_{R \in \mathcal{R}(K \cup A)} Cn((K \setminus R) \cup A)$.

3 Fusion par R-ensembles

Nous proposons un nouveau cadre syntaxique, la Fusion par R-ensembles (RSF), qui consiste à fusionner plusieurs bases de croyances propositionnelles. L'approche consiste à retirer un sous-ensemble des clauses de l'union des bases de croyances, d'après une stratégie P donnée, dans le but de restaurer la cohérence. Ce cadre capture les opérateurs classiques de fusion et peut être mis en œuvre efficacement. Elle généralise l'approche RSR que nous avons brièvement rappelée, et requiert donc la généralisation de la notion de R-ensemble.

Soient $E = \{K_1, \dots, K_n\}$ un ensemble de croyances où $K_i, 1 \leq i \leq n$ est une base de croyances cohérente et X, X' deux sous-ensembles de $K_1 \sqcup \dots \sqcup K_n$.

Définition 4. Soit $E = \{K_1, \dots, K_n\}$ un ensemble de croyances tel que $K_1 \sqcup \dots \sqcup K_n$ est incohérent, $X \subseteq K_1 \sqcup \dots \sqcup K_n$ est un R-ensemble Potentiel de E ssi $(K_1 \sqcup \dots \sqcup K_n) \setminus X$ est cohérent.

Exemple. Soient 3 bases de croyances propositionnelles, $K_1 = \{\neg d, s \vee o\}$, $K_2 = \{\neg s, d \vee o, \neg d \vee \neg o\}$, $K_3 = \{s, d, o\}$.

Les R-ensembles potentiels de $E = \{K_1 \sqcup K_2 \sqcup K_3\}$ sont :

- $R_1 = \{s \vee o, d \vee o, s, d, o\}$;
- $R_2 = \{\neg s, d \vee o, d, o\}$;
- $R_3 = \{d, s\}$;
- $R_4 = \{\neg s, d\}$;
- $R_5 = \{\neg d, s, o, s \vee o\}$;
- $R_6 = \{\neg d, o, \neg s\}$;
- $R_7 = \{\neg d, s, \neg d \vee \neg o\}$;

- $R_8 = \{\neg d, \neg s, \neg d \vee \neg o\}$.

Ainsi que tous leurs sur-ensembles.

Le nombre de R-ensembles Potentiels est exponentiel par rapport au nombre de clauses de E . Ainsi, seuls les R-ensembles Potentiels les plus pertinents, selon la stratégie P choisie, doivent être sélectionnés. Pour cela, une relation de préférence selon la stratégie P , notée \leq_P , est définie sur l'ensemble des R-ensembles potentiels et $X \leq_P X'$ signifie que X est préféré à X' selon la stratégie P .

Définition 5. Soit $E = \{K_1, \dots, K_n\}$ un ensemble de croyances tel que $K_1 \sqcup \dots \sqcup K_n$ est incohérent, $X \subseteq K_1 \sqcup \dots \sqcup K_n$ est un R-ensemble de E selon P ssi

1. X est un R-ensemble Potentiel de E ;
2. Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $X' <_P X$.

On note $\mathcal{F}_P \mathcal{R}(E)$ la collection des R-ensembles⁵ de E selon P . La Fusion par R-ensembles est définie comme suit.

Définition 6. Soit $E = \{K_1, \dots, K_n\}$ un ensemble de croyances. L'opération de fusion $\Delta^P(E)$ est définie par :

$$\Delta^P(E) = \bigvee_{X \in \mathcal{F}_P \mathcal{R}(E)} \{Cn(K_1 \sqcup \dots \sqcup K_n \setminus X)\}$$

Nous montrons maintenant comment capturer les opérateurs classiques de fusion dans notre cadre.

3.1 Représentation des opérateurs classiques de fusion avec RSF

Les différentes stratégies de fusion correspondant aux opérateurs classiques peuvent être capturées grâce à une relation de préférence définie sur les R-ensembles Potentiels.

⁵Si $K_1 \sqcup \dots \sqcup K_n$ est cohérent $\mathcal{F}_P \mathcal{R}(E) = \emptyset$.

L'opérateur *Card*. L'opérateur *Card* est capturé dans notre cadre comme suit :

- Soient X et X' deux R-ensembles Potentiels de E : $X \leq_{Card} X'$ ssi $|X| \leq |X'|$.
- Soit $\mathcal{F}_{Card}\mathcal{R}(E)$ la collection des R-ensembles de E selon *Card*, l'opération de fusion *Card* est représentée par : $\Delta^{Card}(E) = \bigvee_{X \in \mathcal{F}_{Card}\mathcal{R}(E)} \{Cn((K_1 \sqcup \dots \sqcup K_n) \setminus X)\}$

La stratégie *Card* minimise le nombre de clauses à retirer de E et ne tient pas compte des répétitions. Elle est équivalente à l'opérateur *Comb4* défini dans [1].

L'opérateur Σ . L'opérateur Σ est capturé dans notre cadre comme suit :

- Soient X et X' deux R-ensembles Potentiels de E : $X \leq_{\Sigma} X'$ ssi $\sum_{1 \leq i \leq n} |X \cap K_i| \leq \sum_{1 \leq i \leq n} |X' \cap K_i|$.
- Soit $\mathcal{F}_{\Sigma}\mathcal{R}(E)$ la collection de R-ensembles de E selon Σ , l'opération de fusion Σ est représentée par : $\Delta^{\Sigma}(E) = \bigvee_{X \in \mathcal{F}_{\Sigma}\mathcal{R}(E)} \{Cn((K_1 \sqcup \dots \sqcup K_n) \setminus X)\}$

La stratégie Σ minimise le nombre de clauses à retirer de E en prenant en compte la répétition. Il est identique à l'opérateur intersection développé dans [25].

L'opérateur *Max*. L'opérateur *Max* est capturé dans notre cadre comme suit :

- Soient X et X' deux R-ensembles potentiels de E : $X \leq_{max} X'$ ssi $\max_{1 \leq i \leq n} |X \cap K_i| \leq \max_{1 \leq i \leq n} |X' \cap K_i|$ et $X \subseteq X'$.
- Soit $\mathcal{F}_{Max}\mathcal{R}(E)$ la collection de R-ensembles de E selon *Max*, l'opération de fusion *Max* est représentée par : $\Delta^{Max}(E) = \bigvee_{X \in \mathcal{F}_{Max}\mathcal{R}(E)} \{Cn((K_1 \sqcup \dots \sqcup K_n) \setminus X)\}$

La stratégie *Max* essaie de répartir au mieux les clauses à retirer entre les bases de croyances de E et minimise le nombre de clauses à retirer dans la base de croyance la plus impliquée dans l'incohérence.

L'opération *Gmax*. L'opérateur *Gmax* est capturé dans notre cadre comme suit :

- Pour chaque R-ensemble Potentiel X et chaque base de croyances K_i , on définit $p_X^i = |X \cap K_i|$. Soit L_X^E la séquence (p_X^1, \dots, p_X^n) triée par ordre décroissant. Soient X et X' deux R-ensembles Potentiels de E : $X \leq_{Gmax} X'$ ssi $L_X^E <_{lex} L_{X'}^E$.

- Soit $\mathcal{F}_{Gmax}\mathcal{R}(E)$ la collection de R-ensembles de E selon *Gmax*, l'opération de fusion *Gmax* est représentée par : $\Delta^{Gmax}(E) = \bigvee_{X \in \mathcal{F}_{Gmax}\mathcal{R}(E)} \{Cn((K_1 \sqcup \dots \sqcup K_n) \setminus X)\}$

La stratégie *Gmax* est un raffinement de la stratégie *Max*, elle retire les clauses dans les bases de croyances d'après l'ordre décroissant du nombre de clauses impliquées dans l'incohérence.

Exemple. Nous illustrons notre approche grâce à l'exemple suivant, tiré de [23]. Considérons la situation suivante : Un professeur demande à ses élèves quels langages, parmi les suivants, ils souhaitent étudier : SQL (noté s), O_2 (noté o), Datalog (noté d). Le premier souhaite étudier SQL ou O_2 mais pas Datalog ($K_1 = \{-d, s \vee o\}$). Le second veut étudier seulement Datalog ou O_2 mais pas les deux ($K_2 = \{\neg s, d \vee o, \neg o \vee \neg d\}$). Le troisième veut étudier les trois ($K_3 = \{s, d, o\}$). Dans cet exemple, le décideur est le professeur, et les trois groupes d'étudiants peuvent être assimilés à trois agents. Le professeur doit prendre une décision en respectant au mieux le choix des étudiants selon

⁶On note $<_{lex}$ l'ordre lexicographique

une stratégie donnée. Soit $E = \{K_1 \sqcup K_2 \sqcup K_3\}$ l'ensemble de croyances correspondant. Dans cette situation, le résultat de la fusion sera :

$$- \mathcal{F}_{Card} \mathcal{R}(E) = \{\{\neg s, d\}, \{s, d\}\} \text{ and } \Delta^{Card}(E) = \{\{\neg d, s \vee o, d \vee o, \neg o \vee \neg d, s, o\}, \{\neg d, s \vee o, \neg s, d \vee o, \neg o \vee \neg d, o\}\};$$

$$- \mathcal{F}_{\Sigma} \mathcal{R}(E) = \mathcal{F}_{Card} \mathcal{R}(E) \text{ and } \Delta^{\Sigma}(E) = \Delta^{Card}(E);$$

$$- \mathcal{F}_{Max} \mathcal{R}(E) = \{\{\neg s, d\}\} \text{ and } \Delta^{Max}(E) = \{\{\neg d, s \vee o, d \vee o, \neg o \vee \neg d, s, o\}\};$$

$$- \mathcal{F}_{Gmax} \mathcal{R}(E) = \{\{\neg s, d\}\} \text{ and } \Delta^{Gmax}(E) = \{\{\neg d, s \vee o, d \vee o, \neg o \vee \neg d, s, o\}\}.$$

Nous présentons maintenant une implantation de l'approche RSF pour les stratégies *Card* et Σ .

4 Mise en œuvre de RSF par la programmation logique avec sémantique des modèles stables

Nous montrons maintenant comment construire un programme logique, noté P_E , tel que les modèles stables préférés de P_E correspondent au R-ensembles de E .

D'abord, nous montrons comment traduire la Fusion par R-ensembles en un programme logique, dans le même esprit que [20], afin d'obtenir une bijection entre les modèles stables de P_E et les R-ensembles Potentiels de E . Puis, nous définissons la notion de modèle stable préféré afin de réaligner la Fusion par R-ensembles.

4.1 La traduction en un programme logique

Soit $E = \{K_1, \dots, K_n\}$ un ensemble de croyances. L'ensemble de tous les littéraux positifs de P_E est noté V^+ . L'ensemble de tous les littéraux négatifs de P_E est noté V^- . L'ensemble de tous les atomes représentant les clauses sont définis par $R^+ = \{r_c^i \mid c \in K_i\}$ et $CL(r_c^i)$ représente les clauses de K_i correspondant à r_c^i dans P_E , autrement dit $\forall r_c^i \in R^+, CL(r_c^i) = c$. À chaque modèle stable S de P_E , nous associons le R-ensemble Potentiel $CL(R^+ \cap S)$.

1. Dans la première étape, nous introduisons des règles permettant de construire une bijection entre les modèles stables de P_E et les interprétations de V^+ . Pour chaque atome, $a \in V^+$ on introduit deux règles : $a \leftarrow \text{not } a'$ et $a' \leftarrow \text{not } a$ où $a' \in V^-$ est l'atome négatif correspondant à a .
2. Dans la seconde étape, nous excluons les modèles stables S qui correspondent aux interprétations qui ne sont pas des modèles de $(K_1 \sqcup \dots \sqcup K_n) \setminus C_i$ avec $C_i = \{c \mid r_c \in S\}$. Pour chaque clause c of K_j telle que $c = \neg b_o \vee \dots \vee \neg b_n \vee b_{n+1} \vee \dots \vee b_m$, on introduit la règle suivante $r_c^j \leftarrow b_o, \dots, b_n, b'_{n+1}, \dots, b'_m$.

Cette traduction permet de générer l'ensemble des modèles possibles ainsi que les ensembles de règles qui leurs sont associés. Grâce au traitement décrit dans la section 4.2, nous pourrions définir quels sont les ensembles préférés selon la stratégie choisie. Elle est différente de celle proposée dans [3] pour RSR car nous considérons uniquement les atomes positifs R^+ représentant les clauses.

Exemple. Soit $E = \{K_1 \sqcup K_2 \sqcup K_3\}$ défini dans l'exemple. Nous avons :

- $V^+ = \{s, d, o\}$;
- $V^- = \{s', d', o'\}$;
- $R^+ = \{r_{-d}^1, r_{s \vee o}^1, r_{-s}^2, r_{d \vee o}^2, r_{-d \vee \neg o}^2, r_s^3, r_d^3, r_{o'}^3\}$;

La traduction en un programme logique P_E correspondant au problème de fusion est la suivante :

$$\begin{array}{lll} s \leftarrow \text{not } s' & s' \leftarrow \text{not } s & d \leftarrow \text{not } d' \\ d' \leftarrow \text{not } d & o \leftarrow \text{not } o' & o' \leftarrow \text{not } o \\ r_{-d}^1 \leftarrow d & r_{s \vee o}^1 \leftarrow s', o' & r_{-s}^2 \leftarrow s \\ r_{d \vee o}^2 \leftarrow d', o' & r_{-d \vee \neg o}^2 \leftarrow d, o & r_s^3 \leftarrow s' \\ r_d^3 \leftarrow d' & r_{o'}^3 \leftarrow o' & \end{array}$$

Soit S un ensemble d'atomes, on définit I_S comme étant $I_S = \{a \mid a \in S\} \cup \{\neg a \mid a' \in S\}$. La proposition suivante établit la correspondance entre les modèles stables de P_E et les interprétations de $(K_1 \sqcup \dots \sqcup K_n) \setminus CL(R^+ \cap S)$.

Proposition 1. *Soit $E = \{K_1, \dots, K_n\}$ un ensemble de croyances. Soit $S \subseteq V$ un ensemble d'atomes. S est un modèle stable de P_E ssi I_S est une interprétation de V^+ qui satisfait $(K_1 \sqcup \dots \sqcup K_n) \setminus CL(R^+ \cap S)$.*

Afin de calculer les modèles stables correspondant aux R-ensembles, nous introduisons la notion de modèles stables préférés d'après une stratégie P .

Définition 7. *Soit P_E un programme logique. Soient S et S' deux ensembles d'atomes de P_E . S est un modèle stable préféré de P_E selon P ssi :*

1. S est un modèle stable de P_E ;
2. Pour chaque modèle stable S' de P_E , S' n'est pas préféré à S selon P .

La correspondance entre les modèles stables préférés et les R-ensembles est donnée par la proposition suivante pour les stratégies $Card$ et Σ .

Proposition 2. *Soit $E = \{K_1, \dots, K_n\}$ un ensemble de croyances. X est un R-ensemble de E selon la stratégie P ssi il existe un modèle stable préféré S de P_E selon P tel que $CL(R^+ \cap S) = X$.*

Preuve. La proposition précédente peut être réexprimée comme $\{CL(S \cap R^+) \mid S \text{ est un modèle stable préféré selon } P\} = \mathcal{F}_P \mathcal{R}(E)$. Pour simplifier, la preuve ne sera donnée que pour $Card$, la trame reste identique pour Σ .

On cherche donc à prouver que $\{CL(S \cap R^+) \mid S \text{ est un modèle stable préféré selon } P\} = \mathcal{F}_{Card} \mathcal{R}(E)$.

Dans un premier temps, on montre que $\{CL(S \cap R^+) \mid S \text{ est un modèle stable préféré selon } P\} \subseteq \mathcal{F}_{Card} \mathcal{R}(E)$. Pour cela, on suppose qu'il existe un ensemble S appartenant à l'ensemble des modèles stables préférés de P_E selon $Card$ et qu'il est impossible que le R-ensemble potentiel correspondant ne fasse pas partie des R-ensembles de E selon $Card$. Ainsi, on pose $X = CL(S \cap R^+)$. Par la proposition 1, $(K_1 \sqcup \dots \sqcup K_n) \setminus X$ est cohérent. Supposons maintenant que X n'appartienne pas à $\mathcal{F}_{Card} \mathcal{R}(E)$, il existe donc X' tel que $(K_1 \sqcup \dots \sqcup K_n) \setminus X'$ est cohérent et $X' <_{Card} X$. On peut en déduire qu'il existe une interprétation $I_{S'}$ sur l'ensemble des atomes de E qui satisfait $(K_1 \sqcup \dots \sqcup K_n) \setminus X'$. S' est le modèle stable associé à X' et $S' = \{a : a \in I_{S'}\} \cup \{a' : \neg a \in I_{S'}\} \cup \{r_c : c \in X'\}$. On sait que $X' = CL(S' \cap R^+)$ et, par la proposition 1, que S' est un modèle stable de P_E . Cela entraîne que S' est préféré à S car $|X| = |S \cap R^+|$ ce qui est contradictoire avec l'hypothèse que S est un modèle stable préféré.

On montre maintenant que $\mathcal{F}_{Card} \mathcal{R}(E) \subseteq \{CL(S \cap R^+) \mid S \text{ est un modèle stable préféré selon } P\}$. X est un R-ensemble de E donc $(K_1 \sqcup \dots \sqcup K_n) \setminus X$ est cohérent. Il existe donc une interprétation I_S sur l'ensemble des atomes de E qui satisfait $(K_1 \sqcup \dots \sqcup K_n) \setminus X$. On note S le modèle stable de P_E correspondant à X . Par la proposition 1, on sait que S est un

modèle stable de P_E . Supposons que S ne soit pas un modèle stable préféré de P_E selon $Card$, cela suppose qu'il existe S' tel que $|CL(S' \cap R^+)| = |S' \cap R^+| < |CL(S \cap R^+)| = |S \cap R^+|$. Mais $X' = CL(S' \cap R^+)$ et $(K_1 \sqcup \dots \sqcup K_n) \setminus X'$ est cohérent. X n'appartiendrait donc pas à $\mathcal{F}_{Card}\mathcal{R}(E)$ ce qui est contraire à l'hypothèse.

Exemple. Soit P_E le programme logique correspondant à l'exemple précédent. La collection des modèles stables préférés de P_E selon les stratégies $Card$ et Σ est : $\{S_1 = \{s, d', o, r_{-s}^2, r_d^3\}, S_2 = \{s', d', o, r_s^3, r_d^3\}\}$.

Etant donné que $R^+ = \{r_{-d}^1, r_{s \vee o}^1, r_{-s}^2, r_{d \vee o}^2, r_{-d \vee -o}^2, r_s^3, r_d^3, r_o^3\}$ les R-ensembles sont $CL(R^+ \cap S_1) = \{-s, d\}$ et $CL(R^+ \cap S_2) = \{s, d\}$.

4.2 Calcul des modèles stables préférés : l'algorithme *rsf*

L'algorithme *rsf* calcule les modèles stables préférés correspondant aux R-ensembles. Cet algorithme est une modification de celui de *Smodels* qui sélectionne les modèles stables préférés selon la stratégie P choisie. Il construit, étape par étape, une collection de modèles stables candidats. À la fin du calcul, cette collection contient tous les modèles stables préférés correspondant aux R-ensembles.

La sélection des modèles stables préférés est réalisée grâce à la fonction $Condition_P(A)$, où A est un ensemble d'atomes. Cette fonction compare le modèle stable candidat A en cours à ceux qui ont déjà été calculés. Les trois comportements possibles de la fonction $Condition_P(A)$ sont :

1. A ne peut plus conduire à aucun modèle stable préféré. Dans ce cas, le calcul est arrêté et l'algorithme revient en arrière ;

2. L'interprétation qui correspond à A est complète et A est aussi préféré que les meilleurs modèles stables déjà calculés. Dans ce cas, A est ajouté à la collection des modèles stables candidats ;
3. L'interprétation qui correspond à A est complète et A est préféré aux modèles stables précédemment calculées. Dans ce cas, la collection constituée par le singleton A remplace la collection des modèles stables candidats.

Une autre adaptation de *Smodels* concerne l'heuristique originale $heuristic(A)$. Si un atome a est choisi, alors l'atome a' ne peut plus être déduit. Les seuls autres atomes pouvant être déduit sont ceux qui représentent les règles r_c^i .

L'utilisation de l'heuristique standard conduit à maximiser le nombre de r_c^i déduit, ce qui est contradictoire avec les objectifs de RSF et ne tend pas à tirer profit de la réduction de l'espace de recherche. Nous modifions cette heuristique afin de sélectionner les atomes qui minimisent le nombre d'atomes déduits. De cette manière, le premier modèle stable aura plus de chances d'être un modèle stable préféré d'après la stratégie choisie. Cette nouvelle fonction est appelée $mheuristic(A)$.

Les adaptations de l'algorithme original de *Smodels* consistent à : (i) ne pas calculer des sous-ensembles de R^+ conduisant à des modèles stables qui retirent plus de clauses que les meilleurs modèles déjà calculés ; (ii) ne pas calculer plusieurs fois les mêmes sous-ensembles de littéraux de R^+ ; (iii) tirer avantage d'élagages dans l'arbre de recherche.

5 Etude expérimentale préliminaire

Nous présentons les résultats d'une étude expérimentale préliminaire sur l'approche RSF. Les tests ont été conduit sur un Cen-

Algorithme 2 $rsf(A)$

```

 $A \leftarrow expand(A)$ 
si  $conflict(A)$  alors
    renvoyer  $Faux$ 
fini
si (1)  $Condition_P(A) = 1$  alors
    renvoyer  $Faux$ 
sinon si  $A$  conduit à un R-ensemble déjà
calculé alors
    renvoyer  $Faux$ 
sinon si  $A$  couvre  $Atome(E)$  alors
    si (2)  $Condition_P(A) = 0$  alors
         $A$  est ajouté à l'ensemble des solu-
tions
        renvoyer  $Vrai$ 
    sinon
        (3)  $\{A\}$  devient l'ensemble des so-
lutions
        renvoyer  $Vrai$ 
    fini
fini
 $x \leftarrow mheuristic(A)$ 
 $rsf(A \cup \{x\})$ 
 $rsf(A \cup \{not\ x\})$ 

```

trino cadencé à 1,73GHz et équipé d'un IGO de RAM.

À notre connaissance, il n'existe pas d'autres implantations réalisant la fusion de bases de croyances propositionnelles, ni de plateforme de tests pour la fusion. Les tests préliminaires suivants ne sont pas assez exhaustifs pour conclure sur l'efficacité de RSF. Néanmoins, ils montrent la viabilité de l'approche. Afin de pouvoir conclure sur l'efficacité de RSF, nous devons développer une plateforme de tests plus complète.

Les tests sont générés aléatoirement d'après plusieurs paramètres : le nombre de bases (nb), le nombre de clauses dans chacune des bases (nc), le nombre de variables dans les bases (nv), la taille des clauses (sc) et un paramètre qui mesure à quel point les bases diffèrent les unes des autres (d).

Les bases de tests sont construites comme suit. Nous construisons une interprétation I. Ensuite, nous générons aléatoirement des clauses qui sont ajoutées si elles satisfont I. D'une base à l'autre, nous changeons l'interprétation d'après le paramètre (d) qui représente le pourcentage de variables changées. Pour chaque ensemble de paramètres, nous avons testé 10 ensembles de bases de tests différents. Un test est considéré comme un échec si, au bout de 300 secondes, il n'a pas abouti. Nous conservons le temps d'exécution moyen des tests réussis. Les tableaux suivants présentent le pourcentage de tests réussis et, en secondes, le temps d'exécution pour le calcul de tous les R-ensembles.

La table 1 montre le comportement de l'algorithme RSF pour 3 bases de clauses ternaires. L'approche RSF réalise la fusion de 3 bases dans un temps raisonnable jusqu'à un total de 3000 clauses pour 8000 variables. En faisant varier nv/nc , la table 2 exhibe un pic de difficulté lorsque nv/nc

<i>nc</i>	<i>nv</i>	<i>Succes</i> (%)	<i>Temps</i> (s)
100	1000	100	2, 1
200	2000	100	7, 2
400	4000	100	37, 6
600	6000	100	105, 2
800	8000	100	221, 4
1200	12000	0	—

TAB. 1 – Resultats pour $nb = 3$, $sc = 3$ et $d = 20\%$.

<i>nc</i>	<i>nv</i>	<i>Succes</i> (%)	<i>Temps</i> (s)
400	200	40	68, 7
400	400	20	13, 5
400	800	70	17, 5
200	100	90	2, 2
200	200	90	11, 1
200	400	90	2, 1

TAB. 2 – Resultats pour $nb = 3$, $sc = 3$ et $d = 20\%$.

se rapproche de 1.

En analysant le temps d'exécution, nous avons observé que l'heuristique permettant de choisir l'atome consomme beaucoup de temps et doit être améliorée encore.

6 Conclusion

Ce papier présente une approche nouvelle pour réaliser la fusion syntaxique de plusieurs bases de croyances propositionnelles et montre que les opérateurs classiques de fusion *Card*, Σ , *Max*, *Gmax*, initialement définis au niveau sémantique peuvent être capturés dans notre cadre syntaxique.

Ce papier montre que RSF peut être traduit de manière efficace en un programme logique avec sémantique des modèles stables pour les stratégies *Card* et Σ et propose une implantation à partir du système Smo-

dels. Une étude expérimentale préliminaire est présentée et les résultats semblent prometteurs pour la réalisation de la fusion de bases de croyances sur des applications réelles. Nous envisageons également d'implanter RSF pour les stratégies *Max* et *Gmax*.

Une expérimentation plus profonde devra être conduite sur des applications réelles afin de pouvoir donner une évaluation plus précise des performances de l'approche RSF. Cette expérimentation sera conduite dans le cadre d'un projet européen pour la fusion d'information spatiale. De plus, le développement d'une plateforme de tests pour la fusion sera utile, non seulement pour tester RSF, mais, plus globalement, pour d'autres futurs travaux portant sur l'implantation des opérateurs de fusion.

La Fusion par R-ensembles rend possible l'implantation efficace des opérateurs classiques de fusion *Card* et Σ , et, il généralise RSR car la révision de bases de croyances peut être considérée comme la fusion de deux bases de croyances l'une étant préférée à l'autre [6]. De ce fait, RSR revient à fusionner deux sources selon la stratégie *Card*.

Notre cadre peut être étendu dans plusieurs directions. Par exemple, permettre de gérer les contraintes que la fusion de bases de croyances $\Delta(E)$ doit satisfaire ou la fusion de bases de croyances avec priorité.

Un travail futur détaillera la caractérisation sémantique de la méthode de Fusion par R-ensembles. Cette caractérisation repose sur l'ensemble de clauses falsifiées de $K_1 \sqcup \dots \sqcup K_n$ par une interprétation. Selon la stratégie *P* de fusion choisie, un préordre peut être défini sur les interprétations à partir d'une relation de préférence, selon *P*, sur l'ensemble des clauses falsifiées de $K_1 \sqcup \dots \sqcup K_n$.

7 Remerciements

Ce travail a été réalisé avec le soutien de Communauté Européenne à travers le projet VENUS (contrat IST034924) du programme " Information Society Technologies (IST) of the 6th FP of RTD". Les auteurs sont seuls responsables du contenu de cet article. Il ne représente pas l'opinion de la Communauté Européenne et la Communauté Européenne n'est pas responsable de l'utilisation qui pourrait être faite des données figurant dans cet article.

Références

- [1] Chitta Baral, Sarit Kraus, Jack Minker, and V. S. Subrahmanian. Combining knowledge bases consisting of first order theories. In *ISMIS*, pages 92–101, 1991.
- [2] S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Possibilistic Merging and Distance-based Fusion of Propositional Information. *AMAI'02*, 34((1-3)) :217–252, 2002.
- [3] J. Bennaïm, S. Benferhat, O. Papini, and E. Würbel. An answer set programming encoding of prioritized removed sets revision : application to gis. In J. Alferes and Springer Verlag J. Leite, editors, *Proc. of JELIA'04*, pages 604–616, Lisbonne, Portugal, Septembre 2004. Lecture notes in Artificial Intelligence. Logics for AI.
- [4] P. Cholewinski, V. Marek, A. Miki-tiuk, and M. Truszczynski. Computing with default logic. *AI*, 112 :105–146, 1999.
- [5] L. Cholvy. Reasoning about merging information. *Handbook of Defeasible Reasoning and Uncertainly Management Systems*, 3 :233–263, 1998.
- [6] J. Delgrande, D. Dubois, and J. Lang. Iterated revision as prioritized merging. In *Proc. of KR'06*, pages 210–220, Windermere, GB, 2006.
- [7] James Delgrande, Didier Dubois, and Jérôme Lang. Iterated revision as prioritized merging. In *International Conference on Principles of Knowledge Representation and Reasoning (KR), Lake District (UK), 02/06/2006-05/06/2006*, pages 210–220, <http://www.aaai.org/Press/press.php>, 2006. AAAI Press.
- [8] D. Dubois, J. Lang, and H. Prade. Possibilistic Logic. in *Handbook of Logic in Artificial Intelligence and Logic Programming*, 3 :439–513, 1994.
- [9] T. Eiter, N. Leone, C. Mateis, G. Pfeifer, and F. Scarcello. the kr system dlv : progress report, comparison and benchmarks. In *Proc. of KR'98*, pages 406–417, 1998.
- [10] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In Robert A. Kowalski and Kenneth Bowen, editors, *Proc. of the Fifth Int. Conference on Logic Programming*, pages 1070–1080, Cambridge, Massachusetts, 1988. The MIT Press.
- [11] S. Konieczny, J. Lang, and Pierre Marquis. Distance-based merging : A general framework and some complexity results. In *Proc. of KR'02*, pages 97–108, 2002.
- [12] S. Konieczny and R. Pino Pérez. On the logic of merging. In *Proc. of KR'98*, pages 488–498, 1998.
- [13] Sébastien Konieczny. On the difference between merging knowledge bases and combining them. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, *KR2000 : Principles of Knowledge Representation and Reasoning*, pages 135–144, San Francisco, 2000. Morgan Kaufmann.
- [14] C. Lafage and J. Lang. Logical representation of preferences for group

- decision making. In *Proc. of KR'00*, pages 457–468, Breckenridge, CO, 2000.
- [15] Jérôme Lang. Some representation and computational issues in social choice. In L. Godo, editor, *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECS-QARU'05) - LNAI 3571, Barcelona, 06/07/05-08/07/05*, pages 15–26, Berlin Heidelberg, juillet 2005. Springer-Verlag. L010.
- [16] J. Lin. Integration of weighted knowledge bases. *AI*, 83 :363–378, 1996.
- [17] J. Lin and A. O. Mendelzon. Merging databases under constraints. *IJ-CIS'98*, 7(1) :55–76, 1998.
- [18] T. Linke. More on nomore. In *Proc. of NMR'02*, 2002.
- [19] T. Meyer, A. Ghose, and S. Chopra. Syntactic representations of semantic merging operations, 2001.
- [20] I. Niemelä and P. Simons. An implementation of stable model and well-founded semantics for normal logic programs. In *Proc. of LPNMR'97*, pages 420–429, 1997.
- [21] O. Papini. A complete revision function in propositional calculus. In B. Neumann, editor, *Proc. of ECAI92*, pages 339–343. John Wiley and Sons. Ltd, 1992.
- [22] P. Rao, K. Sagonas, Swift, D. S. Warren, and J. Friere. Xsb : A system for efficiently computing well-founded semantics. In *Proc. of LPNMR'97*, pages 430–440, 1997.
- [23] P. Z. Revesz. On the semantics of theory change : arbitration between old and new information. *12th ACM SIGACT-SGMIT-SIGART symposium on Principles of Databases*, pages 71–92, 1993.
- [24] P. Z. Revesz. On the semantics of arbitration. *Journal of Algebra and Computation*, 7(2) :133–160, 1997.
- [25] R.Fagin, G.M.Kuper, J.D.Ullman, and M.Y.Vardi. Updating logical databases, 1986.
- [26] Konieczny S. and Pino Pérez R. Propositionnal belief base merging or how to merge belief/goals coming from several sources and some links with social choice theory. *European Journal of Operational Research*, 160(3) :785–802, 2005.
- [27] P. Simons. Extending and implementing the stable model semantics, 2000.
- [28] Meyer T., Ghose A., and Chopra S. Social choice, merging, and elections. In *Proceedings of ECSQARU'01*, volume LNAI 1695, pages 466–477, 2001.
- [29] E. Würbel, R. Jeansoulin, and O. Papini. Revision : An application in the framework of gis. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, *Proc. of KR'00*, pages 505–516, Breckenridge, Colorado, USA, April 2000. KR, inc., Morgan Kaufmann.