Merging belief bases represented by logic programs

Julien Hué[1] and Odile Papini[2] and Eric Würbel[1]

[1] LSIS-CNRS 6168, Université du Sud Toulon -Var. BP 132. 83957 La Garde Cedex France.
hue, wurbel@univ-tln.fr
[2] LSIS-CNRS 6168, ESIL, Université de la Méditerranée. Av de Luminy. Case 13009 Marseille Cedex 9 France. papini@esil.univmed.fr

Abstract. This paper presents a method which allows for merging beliefs expressed thanks to logic programming with stable model semantics. This method is based on the syntactic merging operators described in the framework of propositional logic. The study of these operators leads to a new definition of the consequence relation between logic programs which is based on the logic of Here-and-There brought by Turner. Moreover, the specificity of the non-monotonic framework given by logic programming with stable model semantics allows for describing a weakened version of the merging operation. Once the operators are defined, their behaviour with respect to the Konieczny and Pino-Perez postulates for merging are examined and discussed.

1 Introduction

Nowadays, the computer science field has to deal with distributed sources of knowledge, especially in the context of databases. These sources are rarely synchronized, they generally conflict. Therefore, the interrogation and sharing of those distributed sources are crucial questions for artificial intelligence.

This problem has been widely discussed within the framework of propositional logic [5, 19, 3]. These operators have been defined in a semantic way [10], a syntactic way [16, 8] or based on morphologic properties of beliefs [4]. The last two methods has led to an implementation [9]. The main advantages of propositional logic is both its strong formal background and its simplicity. But, this simplicity can also be a drawback for representing real world situations. Hence, logic programming with stable model semantics [7] provides a belief representation formalism which is more interesting than classical logic for non-monotonic reasoning. Thus, the question of belief bases merging when beliefs are represented by logic programs deserves attention.

This paper presents a method for merging belief bases represented by logic programs. This method is based on the syntactic operators defined in [9]. Moreover, the non-monotonicity of the stable model semantics allows us to define a weakened version of the merging operation in order to save more beliefs than the strong merging operation. A study of properties will be conducted and an implementation of the merging operations will be provided.

The rest of the paper is organized as follows. Section 1 gives a refresher on belief bases merging and logic programming. Section 2 gives the definition of strong and weak version of merging operations. Section 3 presents an implementation of merging operations based on logic programming with stable model semantics before concluding.

2 Preliminaries and notations

In this section, we give the definitions and notations with respect to logic programming with stable model semantics. We then remind the work described in [21] on the logic of Here-and-There which is used to provide an alternative definition of stable models. We also give a reminder on belief merging and on the Konieczny and Pino-Perez postulates for merging operations.

We consider a finite alphabet $\mathcal{P}$ consisted in propositional atoms. Atoms and formulas are denoted by lower case letters. Sets of atoms are denoted by capital letters. An interpretation is a function from $\mathcal{P}$ to $\{0,1\}$ and the set of every interpretation is denoted by $\mathcal{W}$. For every interpretation I and every set of atoms A, we say that I implies A ($I \models A$) iff every atom of A is true for I. If A and B are two sets of formulas, then $A \models B$ iff $I \models A$ implies $I \models B$. $mod(A)$ represents the set of models of A.

2.1 Logic programming with stable model semantics

A normal logic program is a set of rules with the form $c \leftarrow a_1, \ldots, a_n, not\ b_1, \ldots, not\ b_m$ where $c, a_i (1 \leq i \leq n), b_j (1 \leq j \leq m)$ are propositional atoms and the symbol not stands for negation as failure. A basic program is a logic program without negation as failure. Let r be a rule, we introduce $head(r) = c$ and $body(r) = \{a_1, \ldots, a_n, b_1, \ldots, b_m\}$. Moreover, we define $body^+(r) = \{a_1, \cdots, a_n\}$ which represents the set of positive atoms in the body of this rule and $body^-(r) = \{b_1, \ldots, b_m\}$ which represents the set of negative atoms in the body of this rule, hence $body(r) = body^+(r) \cup body^-(r)$. r^+ represents the rule $head(r) \leftarrow body^+(r)$, obtained from r by withdrawing negative elements to the body of r.

A set X of atoms is closed under a logic program Π iff $\forall r \in \Pi$, $head(r) \in X$ when $body(r) \subseteq X$. The smallest set of atoms which is closed under a basic program Π is denoted $CN(\Pi)$. The reduction of Gelfond-Lifschitz [7] of a program Π with respect to a set X of atoms is defined by $\Pi^X = \{r^+ \mid r \in \Pi \text{ and } body^-(r) \cap X = \emptyset\}$. A set X of atoms is a stable model of Π iff $CN(\Pi^X) = X$. A logic program is said to be inconsistent if it does not have any stable model.

Extended logic programs In order to represent more complete information, it is possible to consider classical negation $\neg$ in addition to the negation as failure. Therefore, an extended logic program is a set of rules in the form: $c \leftarrow a_1, \ldots, a_n, not\ b_1, \ldots, not\ b_m$ where $c, a_i (1 \leq i \leq n), b_j (1 \leq j \leq m)$ are literals (atoms or negation of atoms). The previous definition of a stable model remains valid for any set of atoms X which is consistent (does not contain an atom and its negation).

During the last years, logic programming with stable model semantics has been considered as a convenient tool to handle non-monotonic reasoning. It especially led to several efficient systems, called ASP solvers: smodels [17], DLV [6], NoMore [1], ASSAT [13], CLASP [2].

In [21], H.Turner gave an alternative definition of stable models of a logic program based on the logic of Here-and-There. This logic, which is monotonic, represents interpretations of a logic program in the form of pairs of sets of atoms. Intuitively, the

collection of all HT-interpretations can be constructed as follows. Let Π be a logic program and X and Y be sets of atoms from Π. First, Y is a set of atoms which is consistent with the program Π. Then, for every set Y, X is a set of atoms such that $X \subseteq Y$ and which is a set of plausible consequence of Π knowing Y.

Definition 1 (HT-interpretations). *Let Π be a logic program and X and Y be consistent sets of atoms such that $X \subseteq Y$. A pair (X, Y) is a HT-interpretation of Π iff $Y \models \Pi$ and $X \models \Pi^Y$. We denote by $HT(\Pi)$ the set of all HT-interpretations of program Π.*

Example 1. Let us consider the following program $\Pi = \{a \leftarrow not\ b. \quad b \leftarrow not\ a.\}$. The sets of atoms consistent for every rule of Π are: $\{a\}, \{b\}$ and $\{a, b\}$. It is not possible to have an HT-interpretation with $\emptyset$ as a second element because the absence of a entails the deduction of b (by the rule $a \leftarrow not\ b.$) and *v*ice versa. For $Y = \{a\}$, $\Pi^Y = \{a \leftarrow\}$ then $X = \{a\}$ is the only set of atoms which is consistent with Π^Y; Similarly for $Y = \{b\}$ then $X = \{b\}$; for $Y = \{a, b\}$, $\Pi^Y = \emptyset$ then every $X \subseteq Y$ is a set of plausible consequences of Π^Y.

Finally $HT(\Pi) = \{(\{a\}, \{a\}), (\{b\}, \{b\}), (\emptyset, \{a, b\}), (\{a\}, \{a, b\}), (\{b\}, \{a, b\}), (\{a, b\}, \{a, b\})\}$

If there is only one HT-interpretation $(\{Y\}, \{Y\})$ with a given Y as a second element then Y is consistent with Π and every of its atom is justified.

Definition 2 (Stable models). *Let Π be a logic program and Y be a set of atoms. Y is a stable model of Π iff (Y, Y) is the only element of $HT(\Pi)$ where the second element is Y.*

Lemma 1. *Let Π_1 and Π_2 be two logic programs then $HT(\Pi_1 \cup \Pi_2) = HT(\Pi_1) \cap HT(\Pi_2)$. This result is provided by [21].*

Example 2. In the example 1, the only HT-interpretation of Π with a as a second element is $(\{a\}, \{a\})$ then $\{a\}$ is a stable model of Π. $(\{b\}, \{b\})$ is the only HT-interpretation with $\{b\}$ as a second element then $\{b\}$ is a stable model of Π.

2.2 Belief merging and Removed Sets Fusion

First works on belief merging came from the database area [20] and, later, Konieczny [12] focused on belief merging from a semantical point of view. Belief merging aims at associating a consistent interpretation or a consistent belief base to an inconsistent set of belief bases (called belief profile). The interpretation or belief base resulting of this operation has to be as close as possible to the original belief profile. Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile. We denote $\Delta(\Psi)$ the result of the merging operation. There are two straightforward ways to define $\Delta(\Psi)$ depending on if the sources are conflicting or not, the classical conjunctive merging: $\Delta(\Psi) = \bigwedge_{\varphi_i \in \Psi} \varphi_i$ suitable when the sources are not conflicting and the classical disjunctive merging: $\Delta(\Psi) = \bigvee_{\varphi_i \in \Psi} \varphi_i$ appropriate in case of conflicting sources. These two opposite cases are not satisfactory, so several methods have been proposed for fusion depending on if the bases have the same importance or not.

When the solution provided by the merging operation is an interpretation, it is called semantic merging. When the solution is a belief base, it is called syntactic merging, like in [8]. In particular, the following classical fusion operators have been proposed according to various strategies. The *Sum operator*, denoted by Σ, [15, 18] which follows the point of view of the majority of the belief bases of Ψ. The *Cardinality operator*, denoted by $Card$, [3] which is similar to Σ but without taking repetitions into account. The *Max-based* operator, denoted by Max [19], which tries to best satisfy all the belief bases of Ψ. The *Leximax-based* operator, denoted by $GMax$, [11] which is the lexicographic refinement of Max.

Some methods have been proposed within the context of semantic merging [5, 14, 3]. Among the approaches within the syntactic merging framework, Removed Sets Fusion [9] provides a method and an implementation with the central idea to determine maximal consistent subsets of formulas. As an heuristic, we consider the set of formulas to remove in order to restore consistency.

Definition 3 (Potential Removed Set). *Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile constrainted by the belief base μ such that $\varphi_1 \sqcup \ldots \sqcup \varphi_n \sqcup \mu$ is inconsistent. Let X be a subset of formulas of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$. X is a potential Removed Set of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ with μ for constraints iff $((\varphi_1 \sqcup \ldots \sqcup \varphi_n) \backslash X) \sqcup \mu$ is consistent.*

The Removed Sets Fusion framework captures the classical merging strategies thanks to total pre-orders over Potential Removed Sets.

Definition 4 (Pre-order and strategies). *Let $\Psi = \{\varphi_1 \sqcup \ldots \sqcup \varphi_n\}$ be a belief profile constrainted by the belief base μ and X and Y be potential Removed Sets of Ψ constrainted by μ. For every strategy P, a pre-order $\leq_P$ over potential Removed Sets is defined. $X \leq_P Y$ means that X is preferred to Y according to the strategy P. We define $<_P$ as the strict pre-order associated with $\leq_P$ (i.e. $X <_P Y$ iff $X \leq_P Y$ and $Y \not\leq_P X$).*

Therefore, potential Removed Sets of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ with μ for constraints which are minimal according to the chosen strategy will be considered as the solutions of our merging operation. These potential Removed Sets which are minimal for the $<_P$ pre-order are called Removed Sets according to P.

Definition 5 (Removed Sets according to P). *Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile constrainted by the belief base μ such that $\varphi_1 \sqcup \ldots \sqcup \varphi_n \sqcup \mu$ is inconsistent. Let P be a merging strategy. $X \subseteq \varphi_1 \sqcup \ldots \sqcup \varphi_n$ is a Removed Set of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ with μ for constraints according to the strategy P iff (i) X is a potential Removed Set of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ with μ for constraints; (ii) There is no potential Removed Set Y of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ with μ for constraints such that $Y <_P X$.*

The collection of Removed Sets of Ψ with μ for constraints according to the strategy P is denoted by $\mathcal{F}_\mu^P \mathcal{R}(\Psi)$. The Removed Set Fusion operation is defined by:

Definition 6 (Removed Sets Fusion). *Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile constrainted by the belief base μ, the Removed Sets Fusion operation $\Delta_\mu^P(\Psi)$ is defined by:* $\Delta_\mu^P(\Psi) = \bigvee_{X \in \mathcal{F}_\mu^P \mathcal{R}(\Psi)} \{((\varphi_1 \sqcup \ldots \sqcup \varphi_n) \backslash X) \sqcup \mu\}$

Konieczny and Pino-Perez postulates for merging operation In [12], Konieczny and Pino-Perez defined a set of postulates for belief bases merging. Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile and μ be a belief base.

(KP0) $\Delta_\mu(\Psi) \vdash \mu$.
(KP1) If μ is consistent, then $\Delta_\mu(\Psi)$ is consistent.
(KP2) If Ψ is consistent with μ, then $\Delta_\mu(\Psi) = \bigwedge \Psi \wedge \mu$.
(KP3) If $\Psi_1 \equiv \Psi_2$ and $\mu_1 \equiv \mu_2$, then $\Delta_{\mu_1}(\Psi_1) \equiv \Delta_{\mu_2}(\Psi_2)$.
(KP4) If $\varphi_1 \vdash \mu$ and $\varphi_2 \vdash \mu$, then $\Delta_\mu(\varphi_1 \sqcup \varphi_2) \wedge \varphi_1 \nvdash \bot \rightarrow \Delta_\mu(\varphi_1 \sqcup \varphi_2) \wedge \varphi_2 \nvdash \bot$.
(KP5) $\Delta_\mu(\Psi_1) \wedge \Delta_\mu(\Psi_2) \vdash \Delta_\mu(\Psi_1 \sqcup \Psi_2)$.
(KP6) If $\Delta_\mu(\Psi_1) \wedge \Delta_\mu(\Psi_2)$ is consistent, then $\Delta_\mu(\Psi_1 \sqcup \Psi_2) \vdash \Delta_\mu(\Psi_1) \wedge \Delta_\mu(\Psi_2)$.
(KP7) $\Delta_{\mu_1}(\Psi) \wedge \mu_2 \vdash \Delta_{\mu_1 \wedge \mu_2}(\Psi)$.
(KP8) If $\Delta_{\mu_1}(\Psi) \wedge \mu_2$ is consistent, then $\Delta_{\mu_1 \wedge \mu_2}(\Psi) \vdash \Delta_{\mu_1}(\Psi) \wedge \mu_2$.

3 Syntactic merging of belief bases represented by logic programs

We here propose to study Removed Sets Fusion when beliefs are expressed in terms of logic programs with stable model semantics. The principle remains identical to the propositional case: the removal of some formulas in order to restore consistency. For the rest of this section, we consider that the belief bases are expressed in the form of logic progams. We now give the definition of Removed Sets Fusion in this context, this definition deals with constraints.

Definition 7 (Strong Potential Removed Set). *Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile constrainted by the belief base μ such that $\varphi_1 \sqcup \ldots \sqcup \varphi_n \sqcup \mu$ is inconsistent. Let X be a subset of formulas of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$. X is a strong potential Removed Set of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ constrainted by μ iff $((\varphi_1 \sqcup \ldots \sqcup \varphi_n) \backslash X) \sqcup \mu$ is consistent.*

Definition 8 (Strong Removed Sets according to P). *Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile constrainted by the belief base μ such that $\varphi_1 \sqcup \ldots \sqcup \varphi_n \sqcup \mu$ is inconsistent. Let P be a merging strategy. $X \subseteq \varphi_1 \sqcup \ldots \sqcup \varphi_n$ is a Removed Set of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ constrainted by μ according to the strategy P iff (i) X is a strong potential Removed Set of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ constrainted by μ; (ii) There is no strong potential Removed Set Y of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ constrainted by μ such that $Y <_P X$.*

The collection of Strong Removed Sets of Ψ according to the strategy P is denoted by $\mathcal{F}_\mu^P \mathcal{R}(\Psi)$. The Removed Sets Fusion operation is defined by:

Definition 9 (Strong Removed Sets Fusion). *Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile constrainted by the belief base μ , the Strong Removed Sets Fusion operation $\Delta_\mu^P(\Psi)$ is defined by: $\Delta_\mu^P(\Psi) = \bigvee_{X \in \mathcal{F}_\mu^P \mathcal{R}(\Psi)} \{((\varphi_1 \sqcup \ldots \sqcup \varphi_n) \backslash X) \sqcup \mu\}$*

3.1 Consequence relation between logic programs

In propositional logic, the consequence relation between two set of formulas is clearly defined ($A \models B$ iff $\forall I \in \mathcal{W}$, if $I \models A$ then $I \models B$). This definition can hardly be applied to logic programs.

Example 3. Let $\Pi = \{a\}$ be a belief profile constrainted by the belief base $\mu = \{c \leftarrow not\ a. \quad \neg b.\}$. Thus, the consequences of μ are $\{\neg b, c\}$ and the consequences of $\Pi \cup \mu$ are $\{\neg b, a\}$.

One can easily see that the consequences of $\Pi \cup \mu$ are completely different from the consequences of μ and that a definition of a consequence relation given in terms of stable models inclusion will not properly fit the logic programming framework. This problem can be overcome by chosing a definition of the consequence relation in terms of inclusions of HT-interpretations.

Definition 10 (Inference). *Let Π_1 and Π_2 be logic programs, we define that Π_1 implies Π_2, denoted by $\Pi_1 \models \Pi_2$ iff $HT(\Pi_1) \subseteq HT(\Pi_2)$.*

Example 4. Let us consider again the example 3 in order to illustrate the consequence relation between two sets of rules. In this example, we have:
$HT(\mu) = \{(\{\neg b\}, \{\neg b, a\}), (\{\neg b, a\}, \{\neg b, a\}), (\{\neg b, c\}, \{\neg b, c\}),$
$(\{\neg b\}, \{\neg b, a, c\}), (\{\neg b, a\}, \{\neg b, a, c\}), (\{\neg b, c\}, \{\neg b, a, c\}), (\{\neg b, a, c\}, \{\neg b, a, c\})\}$
$HT(\Pi \cup \mu) = \{(\{\neg b, a\}, \{\neg b, a\}), (\{\neg b, a\}, \{\neg b, a, c\}), (\{\neg b, a, c\}, \{\neg b, a, c\})\}$
We have $HT(\Pi \cup \mu) \models HT(\mu)$ and therefore according to the previous definition $(\Pi \cup \mu) \models \mu$

This new definition allows us to study properly the KP postulate for the Strong Removed Sets Fusion operation.

3.2 KP postulates with respect to Removed Sets Fusion for logic programs

The Strong Removed Sets Fusion operation for logic programs verifies the following KP postulates:

Strategies	(KP0)	(KP1)	(KP2)	(KP3)	(KP4)	(KP5)	(KP6)	(KP7)
Σ	yes	yes	yes	no	no	no	no	no
Card	yes	yes	yes	no	no	no	no	no
Max	yes	yes	yes	no	no	no	no	no
Gmax	yes	yes	yes	no	no	no	no	no

Sketch of proofs The counter-examples are explained only for the Σ operator but also make sense for the other operators.

(KP0) Thanks to the theorem 1, we know that $HT(\Pi \cup \Pi') = HT(\Pi) \cap HT(\Pi')$. By contruction, we have that $HT(\Delta_\mu^\Sigma(\Psi)) \subseteq HT(\mu)$ and then $\Delta_\mu^\Sigma(\Psi) \models \mu$.

(KP1) and (KP2) True by construction

(KP3) Consider $\Psi_1 = \{p. \quad q.\}$, $\Psi_2 = \{p. \quad q \leftarrow p.\}$ and $\mu = \{\neg p.\}$.
$\Delta_\mu^\Sigma(\Psi_1) = \{\neg p, q\} \quad HT(\Delta_\mu^\Sigma(\Psi_1)) = \{(\{\neg p, q\}, \{\neg p, q\})\}$
$\Delta_\mu^\Sigma(\Psi_2) = \{\neg p, q \leftarrow p\} \quad HT(\Delta_\mu^\Sigma(\Psi_2)) = \{(\{\neg p\}, \{\neg p\}) \quad (\{\neg p\}, \{\neg p, \neg q\}$
$(\{\neg p, \neg q\}, \{\neg p, \neg q\})\}$

(KP4) and (KP5)

$$\Pi_1 = \left\{ \begin{array}{cc} a \leftarrow not\ \neg h. & c \leftarrow b. \\ c \leftarrow a. & b \leftarrow not\ \neg h. \\ \neg c. & \end{array} \right\} \qquad \Pi_2 = \left\{ \begin{array}{cc} d \leftarrow not\ \neg c. & h \leftarrow e. \\ h \leftarrow d. & e \leftarrow not\ \neg c. \\ \neg h. & \end{array} \right\}$$

$\Delta^{\Sigma}_{\top}(\Pi_1) = \{a \leftarrow not\ \neg h. \quad b \leftarrow not\ \neg h. \quad c \leftarrow a. \quad c \leftarrow b.\}$ and every HT-interpretation has c in the first set of atoms of the pair. $\Delta^{\Sigma}_{\top}(\Pi_2) = \{d \leftarrow not\ \neg c. \quad e \leftarrow not\ \neg c. \quad h \leftarrow d. \quad h \leftarrow e.\}$ and every HT-interpretation has h in the first set of atoms of the pair. Hence, $\Delta^{\Sigma}_{\top}(\Pi_1 \sqcup \Pi_2) = \Pi_1 \cup \Pi_2$ and $\neg c$ and $\neg h$ are in the first set of atoms of the pair.

(KP6) and (KP7)

$$\Pi = \left\{ \begin{array}{ccc} a \leftarrow not\ c. & \neg d. & d \leftarrow a. \\ d \leftarrow b. & b \leftarrow not\ c. & \end{array} \right\}$$

with $\mu_2 = \{c.\}$. $\Delta^{\Sigma}_{\top}(\Pi) = \{a \leftarrow not\ c. \quad b \leftarrow not\ c. \quad d \leftarrow a. \quad d \leftarrow b.\}$. Each HT-interpretation of $\Delta^{\Sigma}_{\top}(\Pi)$ has d in the first set of atoms of the pair and each HT-interpretation of $\Delta^{\Sigma}_{\top \wedge \mu_2}(\Pi)$ has $\neg d$ in the first set of atoms of the pair.

Discussion The KP postulates have been defined in the framework of monotonic propositional logic. It is normal that the operators described in this paper do not fully respect them. For instance, the postulates **(KP4)** and **(KP5)** mean that if two sets of rules agree on some consequences, their union should respect the consensus; which is clearly not the case in logic programming with stable model semantics.

3.3 Weak Removed Sets Fusion for logic programs

Some sets of rules do not have stable models because they imply inconsistent sets of atoms and some others because it is impossible to justify their consequences. For instance, the program $\Pi = \{\neg a. \quad a \leftarrow b. \quad b.\}$ has for immediate consequences the set of atoms $\{a, \neg a, b\}$ which is inconsistent. On the contrary, the program $\Pi' = \{a \leftarrow not\ b. \quad b \leftarrow not\ c. \quad c \leftarrow not\ a.\}$ which does not imply inconsistent sets of atoms but does not have any stable models because it is impossible to find any self-justifying set of atoms. In one hand, in the case of Π, there are no set of rules φ such that $\Pi \cup \varphi$ has stable models. Though, the only way to restore consistency in those beliefs is to remove some rule. On the other hand, the program Π' is not intrinsincally inconsistent. It is possible to restore consistency without losing beliefs, for instance, the union $\Pi' \cup \{a.\}$ has a stable model ($\{a, b\}$).

It seems reasonable to consider an operation which would keep as much rules as possible as long as consistency can still be restored. We can call this operation a weak merging operation.

Formally, a set of formulas which has at least one HT-interpretation can still have its consistency restored. Generally speaking, consider a logic program which has several HT-interpretations where the second element is Y. If a set of facts Y is added, then this new program will have Y as stable model. Actually, a set of atoms Y is a stable model of Π iff the only HT-interpretation of Π where the second element is Y is (Y, Y). Let

$\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile and μ be a belief base such that $\varphi_1 \sqcup \ldots \sqcup \varphi_n \sqcup \mu$ does not have any stable model, a set of rules X such that $((\varphi_1 \sqcup \ldots \sqcup \varphi_n) \backslash X) \sqcup \mu$ has at least one HT-interpretation, it is called weak potential Removed Set. A weak potential Removed Set of Ψ constrainted by μ which is minimal according to the strategy P, is called weak Removed Set of Ψ constrainted by μ according to P.

Definition 11 (Weak potential Removed Set). *Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile constrainted by the belief base μ such that $\varphi_1 \sqcup \ldots \sqcup \varphi_n \sqcup \mu$ does not have any HT-interpretation. Let X be a subset of rules of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$. X is a weak potential Removed Set of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ constrainted by μ iff $((\varphi_1 \sqcup \ldots \sqcup \varphi_n) \backslash X) \sqcup \mu$ has at least one HT-interpretation.*

Definition 12 (Weak Removed Set). *Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile constrainted by the belief base μ such that $\varphi_1 \sqcup \ldots \sqcup \varphi_n \sqcup \mu$ does not have any HT-interpretation. Let P be a merging strategy. $X \subseteq \varphi_1 \sqcup \ldots \sqcup \varphi_n$ is a weak Removed Set of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ constrainted by μ iff (i) X is a weak potential Removed Set of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ constrainted by μ; (ii) There is no Y which is a weak potential Removed Set of $\varphi_1 \sqcup \ldots \sqcup \varphi_n$ constrainted by μ such that $Y <_P X$.*

The collection of Weak Removed Sets of Ψ according to the strategy P is denoted by $\mathcal{F}_\mu^{P,w}\mathcal{R}(\Psi)$. The Removed Sets Fusion operation is defined by:

Definition 13 (Weak Removed Sets Fusion). *Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile constrainted by the belief base μ and P be a merging strategy, the Weak Removed Sets Fusion operation $\Delta_\mu^{P,w}(\Psi)$ is defined by: $\Delta_\mu^{P,w}(\Psi) = \bigvee_{X \in \mathcal{F}_\mu^{P,w}\mathcal{R}(\Psi)} \{((\varphi_1 \sqcup \ldots \sqcup \varphi_n) \backslash X) \sqcup \mu\}$*

4 Implementation of the merging problem

The implementation of Removed Sets Fusion for logic programs stems from an approach similar to the propositional cases described in [8]. Let Ψ be a belief profile and μ be a belief base representing constraints on Ψ. It constructs a logic program $\Pi_{\Psi,\mu}$, such that for any strategy P, the preferred stable models of $\Pi_{\Psi,\mu}$ according to P correspond to the Removed Sets of Ψ constrainted by μ according to P. In the same way, we construct a logic program $\Pi^w_{\Psi,\mu}$ to solve the weak merging operation.

The first part of the program gives the potential Removed Sets of Ψ constrainted by μ and the second part selects Removed Sets amongst them. It is done thanks to the enumeration of possible interpretations which will provide the maximal consistent subsets of logic program. There is however some differences with the propositional case:

- A model for a propositional logic base can contain either a or $\neg a$. A stable model can contain a, $\neg a$ or none of them.
- In Removed Sets Fusion, the subset of formulas generated by an interpretation has to be consistent (the interpretation which generates it being the model), which is not the case for logic programs because an interpretation can satisfy every rule without being a stable model of the program.

Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be a belief profile and μ be a belief base representing constraints. The set of all positive (resp. negative) literals of $\Pi_{\Psi,\mu}$ is denoted by V^+ (resp. V^-). The set of atoms representing rules is defined by $R^+ = \{r_f^i \mid f \in \varphi_i\}$ and $FO(r_f^i)$ denotes the rule of φ_i corresponding to r_f^i in $\Pi_{\Psi,\mu}$. Namely, $\forall r_f^i \in R^+, FO(r_f^i) = f$. To each answer set of $\Pi_{\Psi,\mu}$ we associate the potential Removed Set $FO(R^+ \cap S)$. Considering this, we will describe the logic program which will represent the merging problem. Our program will have four steps:

- The first step generates the set of interpretations of V which can be stable models of a subset of rules. (4.1)
- The second step assures that there exists a rule which allows the atom to be present in the current interpretation. (4.2)
- The third step allows to point out the rules that should be removed. (4.3)
- The last step, finally, is used to encode the strategy. (4.4)

Example 5. We illustrate each part of the translation with the following example. Consider $\Psi = \{\varphi_1, \varphi_2\}$ with $\varphi_1 = \{f_1 \ : \ a \leftarrow not\ b. \quad f_2 \ : \ b \leftarrow not\ c.\}$, $\varphi_2 = \{f_3 \ : \ c \leftarrow not\ a. \quad f_4 \ : \ d \leftarrow a.\}$ and $\mu = \{\leftarrow a.\}$.

4.1 First step: generating interpretations

Generating all the interpretations of V for the set of atoms $\{a_1, \ldots, a_n\}$ is done through the rules $\{a_1, a_1', \ldots, a_n, a_n'\}$ where a_i' represents the negation of a_i. Finally, to avoid the presence of an atom and its negation in the same interpretation, we introduce, for every atom a_i, the contrainst $\leftarrow a_i, a_i'$.

Case of basic program When dealing with basic programs (which do not contain any negation), this part can be reduced to the instruction $\{a_1, \ldots, a_n\}$.

Example 6. Continuing the example 5. Their interpretations are generated thanks to the statement $\{a, b, c, d\}$.

4.2 Second step: rules to remove

It is impossible that a set of atoms S is a stable model of a logic program $\Pi_{\Psi,\mu}$ if there exists a rule f such that S satisfies $body(f)$ and $head(f) \notin S$. Such a rule should therefore be removed in order to allow the interpretation to be a model.

Hence, for every rule $f \ : \ head(f) \leftarrow body(f)$, we introduce the rule $r_f \leftarrow not\ head(f), body(f)$. The presence of the atom r_f means that the rule f should not be considered in the stable model corresponding to S.

Example 7. Consider the example 5. The selection of rules to remove is done thanks to $r_1 \leftarrow not\ a, not\ b.\ r_2 \leftarrow not\ b, not\ c.\ r_3 \leftarrow not\ c, not\ a\ r_4 \leftarrow not\ d, a.$ [3]

[3] For the sake of readability, we will note r_i instead of r_{f_i}.

4.3 Third step: necessity of the presence of an atom

Generally speaking, a stable model represents the set of reasonable consequences of a logic program. It means that an atom only belongs to a stable model if it has been deduced thanks to a rule or a fact. It is necessary, for every set of atoms S, that there exists a very reason for any atom to be true.

For every atom a, we define an atom $auth(a)$ representing the fact that an atom a has been authorized to be deduced. Therefore, for every atom a, we introduce the rule $\leftarrow a, not\ auth(a)$ which implies the impossibility for an atom to be present if its presence is not justified.

Logically, $auth(a)$ is deduced if a rule has not been removed and if $body(f) \subseteq S$. For every rule f, we introduce the rule $auth(head(f)) \leftarrow not\ r_f, body(f)$.

Example 8. Continuing the example 5. The rules allowing to determine if an atom has a reason for being deduced are: $\leftarrow a, not\ auth(a).$ $\leftarrow b, not\ auth(b).$ $\leftarrow c, not\ auth(c).$ $\leftarrow d, not\ auth(d).$ $auth(a) \leftarrow not\ b, not\ r_1.$ $auth(b) \leftarrow not\ c, not\ r_2.$ $auth(c) \leftarrow not\ a, not\ r_3.$ $auth(d) \leftarrow a, not\ r_4.$

The whole $\Pi_{\Psi,\mu}$ program has the following stable models: $\{b, auth(b), r_3\}$ $\{r_1, r_2, r_3\}$ $\{c, auth(c), r_1\}$.

Proposition 1. *Let $\Psi = \{\varphi_1, \ldots, \varphi_n\}$ be an belief profile and μ be a belief base representing constraints. Let $S \subseteq V$ be a set of atoms. S is a stable model of $\Pi_{\Psi,\mu}$ iff I_S is an interpretation of V^+ which satisfies $((\varphi_1 \sqcup \ldots \sqcup \varphi_n) \backslash FO(R^+ \cap S)) \sqcup \mu$.*

4.4 Fourth step: optimization

The optimization statements are similar to the ones presented in [9].

Example 9. Continuing the example 5.

For the Σ strategy, the optimization statement will be:
$minimize\{r_1, r_2, r_3, r_4\}.$

For the Max strategy, the optimization statements will be:
$\#domain\ possible(U).$ $\#domain\ base(V).$ $\#domain\ possible(W).$
$possible(1..2).$ $base(1..2).$ $size(U) \leftarrow U\{r_f^V | F_0(f) \in \varphi_V\}U.$
$negmax(W) \leftarrow size(U), U > W.$ $max(U) \leftarrow size(U), not\ negmax(U).$
$minimize[max(1) = 1, max(2) = 2]$

For both strategies, the preferred stable models of $\Pi_{\Psi,\mu}$ are: $\{b, auth(b), r_3\}$ and $\{c, auth(c), r_1\}$ which correspond to the Strong Removed Sets of Ψ constrainted by μ according to Σ and Max: $\{a \leftarrow not\ b.\}$ and $\{c \leftarrow not\ a.\}$.

The following proposition establishes the one-to-one correspondence between the preferred stable models of $\Pi_{\Psi,\mu}$ according to P and the Strong Removed Sets of $\Delta_\mu^P(\Psi)$

Proposition 2. *The set of Strong Removed Sets of Ψ constrainted by μ according to P is the set of preferred stable models of $\Pi_{\Psi,\mu}$ according to P. This proposition holds for Σ, Max, $Card$ and $Gmax$.*

4.5 Weak merging operation

The main difference between the strong and weak merging operations is that an atom does not need justification to belong to an interpretation. Therefore a program to solve the weak version of merging operator will have the same rules as a strong one except the rules described in 4.3.

Example 10. Consider again the example in 5. The program $\Pi^{w}_{\Psi,\mu}$ is:

$$\begin{array}{c} \{a,b,c,d\}.\ r_1 \leftarrow not\ a, not\ b.\ r_2 \leftarrow not\ b, not\ c. \\ \leftarrow a. \quad r_3 \leftarrow not\ c, not\ a. \quad r_4 \leftarrow not\ d, a. \\ minimize\{r_1, r_2, r_3, r_4\}. \end{array}$$

This program has 20 stable models and the minimal one for every strategy is $\{a, b, c, d\}$ which corresponds to the Weak Removed Set of Ψ constrainted by μ which is the empty set.

The following proposition establishes the one-to-one correspondence between the preferred stable models of $\Pi^{w}_{\Psi,\mu}$ and the Weak Removed Sets of $\Delta^{P,w}_{\mu}(\Psi)$

Proposition 3. *The set of Weak Removed Sets of Ψ constrainted by μ according to P is the set of preferred stable models of $\Pi^{w}_{\Psi,\mu}$ according to P. This proposition holds for Σ, Max, $Card$ and $Gmax$.*

5 Conclusions and perspectives

We presented a first approach for merging logic programs based on Removed Sets Fusion. A study of the properties has been led thanks to the Konieczny and Pino-Perez postulates. This study showed that the Konieczny and Pino-Perez postulates are not suitable in the framework of belief bases merging when beliefs are expressed thanks to logic programs with stable model semantics. We proposed a definition of an inference relation between logic programs. We also defined a weakened version of the merging operation.

Removed Sets Fusion for belief bases represented by logic programs is translated into a logic program with stable model semantics and the one-to-one correspondence between removed sets (both Weak and Strong version) and preferred stable models is shown. Moreover, the paper shows how Removed Sets Fusion can be performed with any ASP solver.

Future works will study the properties of the weak Removed Sets Fusion. A more extensive experimentation of the Removed Sets Fusion for belief represented by logic programs has to be performed. It also can be relevant to study KP postulates in order to allow postulates for dealing with a broader range of frameworks for representing beliefs.

Acknowledgements

Work partially supported by the European Community under project VENUS (Contract IST-034924) of the "Information Society Technology (IST) program of the 6th FP of RTD". The

authors are solely responsible for the contents of this paper. It does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data therein.

References

1. C. Anger, K. Konczak, and T. Linke. Nomore: Non-monotonic reasoning with logic programs. In *JELIA'02*, pages 521–524, 2002.
2. C. Baral, G. Brewka, and J. Schlipf, editors. *Proc. of the Ninth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'07)*, volume 4483 of *Lecture Notes in Artificial Intelligence*, 2007.
3. C. Baral, S. Kraus, J. Minker, and V. S. Subrahmanian. Combining knowledge bases consisting of first order theories. In *ISMIS*, pages 92–101, 1991.
4. Isabelle Bloch and Jérôme Lang. Towards mathematical morpho-logics. In *Technologies for constructing intelligent systems: tools*, pages 367–380, Heidelberg, Germany, Germany, 2002. Physica-Verlag GmbH.
5. L. Cholvy. Reasoning about merging information. *Handbook of DRUMS*, 3:233–263, 1998.
6. 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.
7. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of the 5th Int. Conf. on Log. Prog.*, pages 1070–1080, 1988.
8. J. Hue, O. Papini, and Eric Würbel. Syntactic propositional belief bases fusion with removed sets. *Proc. of ECSQARU'07*, 2007.
9. J.Hue, O.Papini, and E.Würbel. Removed sets fusion: Performing off the shelf. In *Proc. of ECAI08*, 2008.
10. S. Konieczny. On the difference between merging knowledge bases and combining them. In *Proc of KR'00*, pages 135–144. Morgan Kaufmann, 2000.
11. S. Konieczny and R. Pino Pérez. On the logic of merging. In *Proc. of KR'98*, pages 488–498, 1998.
12. S. Konieczny and R. Pino Pérez. Merging with integrity constraints. *Lecture Notes in Computer Science*, 1638:233–??, 1999.
13. F. Lin and Y. Zhao. Assat: computing answer sets of a logic program by sat solvers. *Artif. Intell.*, 157(1-2):115–137, 2004.
14. J. Lin. Integration of weighted knowledge bases. *AI*, 83:363–378, 1996.
15. J. Lin and A. O. Mendelzon. Merging databases under constraints. *Int. Journal of Cooperative Information Systems*, 7(1):55–76, 1998.
16. T. Meyer, A. Ghose, and S. Chopra. Syntactic representations of semantic merging operations. In *Proc. of IJCAI'01*, 2001.
17. 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.
18. P. Z. Revesz. On the semantics of theory change: arbitration between old and new information. 12^{th} *ACM SIGACT-SGMIT-SIGART symposium on Principes of Databases*, pages 71–92, 1993.
19. P. Z. Revesz. On the semantics of arbitration. *Journal of Algebra and Computation*, 7(2):133–160, 1997.
20. R.Fagin, G.M.Kuper, J.D.Ullman, and M.Y.Vardi. Updating logical databases. In *Advances in computing research*, pages 1–18, 1986.
21. Hudson Turner. Strong equivalence made easy: nested expressions and weight constraints. *Theory Pract. Log. Program.*, 3(4):609–622, 2003.