Merging Rules: Preliminary Version

Richard Booth
Faculty of Informatics
Mahasarakham University
Mahasarakham 44150, Thailand
richard.b@msu.ac.th

Souhila Kaci
CRIL
Rue de l'Université SP 16
62307 Lens Cedex, France
kaci@cril.univ-artois.fr

Leendert van der Torre
ILIAS
University of Luxembourg
Luxembourg
leon.vandertorre@uni.lu

Abstract

In this paper we consider the merging of rules or conditionals. In contrast to other approaches, we do not invent a new approach from scratch, for one particular kind of rule, but we are interested in ways to generalize existing revision and merging operators from belief merging to rule merging. First, we study ways to merge rules based on only a notion of consistency of a set of rules, and illustrate this approach using a consolidation operator of Booth and Richter. Second, we consider ways to merge rules based on a notion of implication among rules, and we illustrate this approach using so-called $\min$ and $\max$ merging operators defined using possibilistic logic.

Introduction

We are interested in the merging or fusion of rules or conditionals. When there are several sources of rules that are in some sense conflicting, incoherent or inconsistent, then a rule merging operator returns a weaker non-conflicting set of rules. Such merging operators can be used in many areas of artificial intelligence, for example when merging regulations in electronic institutions, merging conditional default rules, or merging conditional goals in social agent theory.

In general, there are two approaches to develop operators and algorithms to merge rules. One approach starts from a particular kind of rule, and develops a merging operator for a particular application domain. The other approach tries to generalize existing operators from belief merging, which have been developed as a generalization of belief revision operators. In this paper we follow in the latter approach, and we address the following research questions:

1. How to define a general framework to study rule merging and develop rule merging operators?
2. Given a merging operator for belief merging, how can we use it for rule merging?
3. Defeasible rules can be stratified into a prioritized rule base. How can we use this stratification in rule merging?

Though many notions of rules have been defined, they are typically represented as syntactic objects $\phi \rightarrow \psi$ in a meta-language, expressing a conditional statement "if ϕ then ψ", where ϕ and ψ are formulas of an object language, for example propositional or first-order logic. Given a set of such rules R expressed as pairs of formulas of a language L, we can apply the rules to a context S, consisting of formulas of L, which results again in a set of sentences of L. In this paper, following conventions from input/output logic (Makinson & van der Torre 2000), we write $out(R, S)$ for the result of applying the rules of R to S.

A crucial ingredient of belief merging operators is a notion of inconsistency. However, since rules are typically represented in the meta-language, there is no obvious choice of rule inconsistency which can be used. To use a merging operator for rule merging, we have to define when a set of rules is conflicting or contradictory. We discuss various ways to define the inconsistency of a set of rules, and illustrate how a merging operator for belief merging can be used for rule merging using a generalization of the so-called AGM partial meet approach (Alchourrón, Gärdenfors, & Makinson 1985; Hansson 1999) due to (Booth & Richter 2005).

Moreover, a notion of consistency is sufficient to define selection operators, but in general we need also a notion of implication among rules. For example, when we interpret the arrow $\rightarrow$ as an implication in conditional logic, or as the material implication of propositional logic, then two rules $\phi \rightarrow \psi$ and $\xi \rightarrow \varphi$ imply the rule $\phi \wedge \xi \rightarrow \psi \vee \varphi$. Moreover, if we merge the former two rules, the latter one may be the result. We illustrate this using merging operators from possibilistic logic (Dubois, Lang, & Prade 1994), a logic that associates with a formula a numerical value between 0 and 1. One interpretation of this value is that it represents a stratification of the formulas in the knowledge base in formulas with higher and lower priority. A particular kind of conditionals has been defined, and these conditionals have been stratified using a stratification algorithm.

The layout of this paper is as follows. We first discuss the inconsistency of a set of rules, and illustrate it on the merging operator of Booth and Richter. Then we discuss rule implication, and illustrate it on merging operators defined using possibilistic logic.

Preliminaries: Unless otherwise indicated, our background logic in this paper will be a propositional logic L containing the usual propositional connectives, including material implication which we denote by $\supset$. For any set of formulas S, $Cn(S)$ is the set of logical consequences of S. We will say S is Cn-consistent if S is classically consistent. Ω is the set of all propositional interpretations relative to L.

Rules, alias conditionals, will be of the form $\phi \to \psi$ where $\phi, \psi \in L$. Thus L^2 is the set of all rules.

Rule consistency

Applying rules

In this paper we make only minimal assumptions on $out(R, S)$ in general. We assume $out(R, S)$ is a logically closed set of formulas of L. We also assume that a rule can be applied when the context is precisely the body of the rule, and that a set of rules cannot imply more than the materialization of the rules, that is, then assuming that the set of rules are formulas of L by interpreting the condition as a material implication. There are many additional properties one may want to impose on the application of rules.

Let R be a set of pairs from a logic L, let S be a set of formulas of L, $out(R, S) \subseteq L$ is assumed to satisfy the following conditions:

1. $out(R, S) = Cn(out(R, S))$
2. If $\phi \to \psi \in R$, then $\psi \in out(R, \{\phi\})$;
3. $out(R, S) \subseteq Cn(S \cup \{\phi \supset \psi \mid \phi \to \psi \in R\})$

Seven kinds of such rules have been studied in the input/output logic framework (Makinson & van der Torre 2000). One example, called *simple-minded output* in (Makinson & van der Torre 2000), is

$$out_1(R, S) = Cn(\{\psi \in L \mid (\phi \to \psi) \in R \text{ and } \phi \in Cn(S)\}).$$

We will refer to this operation again later in this section. Many other examples can be defined. They satisfy additional properties, such as the monotonicity property that the output $out(R, S)$ increases if either R or S increases.

Consistency of output

Since rules are defined as pairs of formulas of L, we can define the consistency of a set of rules in terms of Cn-consistency. In input/output logic, the following two notions of consistency have been defined for a set of rules relative to a given context S (Makinson & van der Torre 2001):

Output constraint A set of rules R satisfies the output constraint when $out(R, S)$ is Cn-consistent.

Input/output constraint A set of rules R satisfies the input/output constraint when $S \cup out(R, S)$ is Cn-consistent.

When the application of a set of rules $out(R, S)$ always contains the input S, then these two kinds of constraints obviously coincide. However, there are several intuitive notions of rules, such as norms, which do not have this property, and where the two constraints are distinct.

Rule consistency

We consider a weak and a strong notion of consistency of a set of rules. A set is weakly consistent when it does not lead to Cn-inconsistent output for the inputs of the available rules, and it is strongly consistent when it does not lead to Cn-inconsistency for any consistent context. Strong consistency is sometimes used, for example, when developing institutional regulations.

Weak consistency For all $\phi \to \psi \in R$, we have $out(R, \{\phi\})$ is Cn-consistent.

Strong consistency For any Cn-consistent $S \subseteq L$, we have that $out(R, S)$ is Cn-consistent.

Example: A consolidation operator

(Booth & Richter 2005) assume a very abstract framework based on the abstract framework for fuzzy logic due to (Gerla 2001). They just need three ingredients:

(i) a set L_0 of formulas,

(ii) a set W of abstract *degrees* which can be assigned to the formulas in L_0 to create *fuzzy belief bases*, and

(iii) a special subset $\mathbf{Con}$ of these fuzzy belief bases which specifies those fuzzy bases which are meant to be *consistent*, in whatever sense.[1]

For the set L_0 they assume no particular structure – it is just an arbitrary set, while the only thing assumed about W in general is that it is a complete distributive lattice. Formally, a fuzzy belief base is a function $u : L_0 \to W$. $\mathcal{F}(L_0)$ denotes the set of all fuzzy bases. If $u(\varphi) = a$, then this is interpreted as the information that the degree of φ is *at least* a, i.e., it *could* in actual fact be bigger than a, but the information contained in u doesn't allow us to be more precise. The partial order over W is denoted by $\leq_W$. The "fuzzy" subset relation $\sqsubseteq$ between fuzzy bases is defined by

$$u \sqsubseteq v \text{ iff } u(\varphi) \leq_W v(\varphi) \text{ for all } \varphi \in L_0.$$

So $\sqsubseteq$ is an "information ordering": $u \sqsubseteq v$ iff the information contained in v is more "precise" than u. Under these definitions $(\mathcal{F}(L_0), \sqsubseteq)$ itself forms a complete distributive lattice. Given any set $X \subseteq \mathcal{F}(L_0)$ of fuzzy bases the infimum and supremum of X under $\sqsubseteq$ are denoted by $\bigsqcap X$ and $\bigsqcup X$ respectively, with $u \sqcup v$ being written rather than $\bigsqcup\{u, v\}$, etc. In the simplest case we can take $W = \{0, 1\}$ with $0, 1$ standing for true and false respectively. In this case the fuzzy bases just reduce to (characteristic functions of) crisp belief bases and we can write $\varphi \in u$ rather than $u(\varphi) = 1$, while $\sqsubseteq, \bigsqcup, \bigsqcap$ reduce to the usual $\subseteq, \bigcup, \bigcap$.

The set $\mathbf{Con} \subseteq \mathcal{F}(L_0)$ is required to satisfy two conditions. First, it is assumed to be *downwards closed* in the lattice $\mathcal{F}(L_0)$:

If $v \in \mathbf{Con}$ and $u \sqsubseteq v$ then $u \in \mathbf{Con}$.

The second condition is slightly more involved, and corresponds to a type of compactness condition:

Definition 1 $\mathbf{Con}$ *is* logically compact *iff* $\bigsqcup X \in \mathbf{Con}$ *for any* $X \subseteq \mathbf{Con}$ *such that* $u, v \in X$ *implies there exists* $w \in X$ *such that* $u \sqcup v \sqsubseteq w$.

[1] Actually for *(iii)* they start off assuming a *deduction* operator D which for each fuzzy base returns a new fuzzy base denoting its fuzzy *consequences*. However, as they point out, only the plain notion of consistency is required for their formal results. (See Section 7 of (Booth & Richter 2005).)

In other words, the supremum of every directed family of consistent fuzzy bases is itself consistent.

Given all this, we can make the following definitions, assuming some fixed $u \in \mathcal{F}(L_0)$:

Definition 2 *$u\perp$ is defined to be the set of maximally consistent fuzzy subsets of u, i.e., $v \in u\perp$ iff*
(i). $v \sqsubseteq u$.
(ii). $v \in \mathbf{Con}$.
(iii). *If $v \sqsubset w \sqsubseteq u$ then $w \notin \mathbf{Con}$.*
A selection function (for u) *is a function γ such that $\emptyset \neq \gamma(u\perp) \subseteq u\perp$.*

From a selection function γ we define a *consolidation* operator $!_\gamma$ for u by setting

$$u!_\gamma = \bigsqcap \gamma(u\perp).$$

Definition 3 *! is a* partial meet fuzzy base consolidation *operator (for u) if $! = !_\gamma$ for some selection function γ for u.*

Partial meet fuzzy base consolidation can be thought of as a special case of a more general operation of partial meet fuzzy base *revision*. In fact consolidation amounts to a revision by a "vacuous" revision input $(\varphi/0_W)$ representing the new information that the degree of φ is at least 0_W, where 0_W is the minimal element of the lattice W. This more general operation was studied and axiomatically characterized in (Booth & Richter 2005). The following characterization of partial meet fuzzy consolidation does not appear in (Booth & Richter 2005), though it can be proved using similar methods.

Theorem 1 *! is a partial meet fuzzy consolidation operator iff ! satisfies the following three conditions:*

1. $u! \in \mathbf{Con}$.
2. $u! \sqsubseteq u$
3. *For all $\phi \in L_0$, $b \in W$, if $b \not\leq_W u!(\phi)$ and $b \leq_W u(\phi)$ then there exists $u' \in \mathbf{Con}$ such that $u! \sqsubseteq u' \sqsubseteq u$ and $u' \sqcup (\phi/b) \notin \mathbf{Con}$.*

In 3, $u!(\phi)$ is the degree assigned to ϕ by the fuzzy base $u!$, while (ϕ/b) denotes that fuzzy base which assigns b to ϕ and 0_W to every other formula.

Application

Given an arbitrary set $R \subseteq L^2$ (possibly infinite) of rules, we need to formally define when R is consistent. For now, we use the earlier-defined notion of strong consistency for out_1, which we will refer to as consistent$_1$.

Definition 4 *Let $R \subseteq L^2$ be a set of rules. We say R is* consistent$_1$ *iff $out_1(R, \phi)$ is Cn-consistent for all Cn-consistent $\phi \in L$.*

Using results of (Makinson & van der Torre 2000), we get an alternative characterization of consistent$_1$:

Proposition 1 *The following are equivalent:*
(i). *R is consistent$_1$.*
(ii). *For all Cn-consistent $\phi \in L$, $\phi \to \perp$* cannot *be derived from R using the rule-set* Rules$_1$ *that contains SI: derive $(\phi \wedge \xi) \to \psi$ from $\phi \to \psi$, WO: derive $\phi \to (\psi \vee \varphi)$ from $\phi \to \psi$, AND: derive $\phi \to (\psi \wedge \varphi)$ from $\phi \to \psi$ and $\phi \to \varphi$.*

To help define merging operators for rules, our aim now is to define an operation which takes an arbitrary set of rules R and returns a new rule set $R!$ which is consistent$_1$. We set up the following definitions:

Definition 5 *$R\perp$ is defined to be the set of maximally consistent$_1$ subsets of R, i.e., $X \in R\perp$ iff*
(i). *$X \subseteq R$.*
(ii). *X is consistent$_1$.*
(iii). *If $X \subset Y \subseteq R$ then Y is inconsistent$_1$.*
A selection function (for R) *is a function γ such that $\emptyset \neq \gamma(R\perp) \subseteq R\perp$.*

From a given selection function γ we then define a consolidation operator for R by setting

$$R!_\gamma = \bigcap \gamma(R\perp).$$

Definition 6 *! is a* partial meet consolidation *operator (for R) if $! = !_\gamma$ for some selection function γ for R.*

What are the properties of this family of consolidation operators? It turns out the following is a sound and complete set of properties for partial meet consolidation.

1. $R!$ is consistent$_1$.
2. $R! \subseteq R$.
3. If $\phi \to \psi \in R \setminus R!$ then there exists X such that $R! \subseteq X \subseteq R$, X is consistent$_1$, and $X \cup \{\phi \to \psi\}$ is inconsistent$_1$.

Theorem 2 *! is a partial meet consolidation operator iff ! satisfies 1–3 above.*

Now, by considering the special case $L_0 = L^2, W = \{0, 1\}$, and $\mathbf{Con}$ = consistent$_1$ we obtain Theorem 2 as just an instance of Theorem 1. However, to be able to do this we need to check that consistent$_1$ satisfies the conditions required of it:

Theorem 3 *consistent$_1$ is downwards closed and logically compact.*

Proof: The easiest way to show these is by considering the proof-theoretical characterization of consistent$_1$ from Proposition 1*(ii)*.

To show consistent$_1$ is downwards closed in this case means to show that if R is consistent$_1$ and $R' \subseteq R$ then R' is consistent$_1$. But if R' is inconsistent$_1$ then $\phi \to \perp$ is derivable from R' using *Rules*$_1$, for some Cn-consistent ϕ. If $R' \subseteq R$ then obviously any derivation from R' is a derivation from R. Hence this implies R is inconsistent$_1$.

To show consistent$_1$ is logically compact means to show that $\bigcup X$ is consistent$_1$ for any set X of consistent$_1$ rule bases such that $R, R' \in X$ implies there exists $R'' \in X$ such that $R \cup R' \subseteq R''$. But suppose for contradiction consistent$_1$ was *not* logically compact. Then there is some set X of consistent$_1$ rule bases satisfying the above condition and such that $\bigcup X$ is inconsistent$_1$. This means for some Cn-consistent ϕ there is a derivation of $\phi \to \perp$ from $\bigcup X$ using *Rules*$_1$. Let $A_1, \ldots, A_n$ be the elements of $\bigcup X$ used in this derivation, and let $R_1, \ldots, R_n$ be rule bases in X such that $A_i \in R_i$. By repeated application of the above condition on X we know there exists $R'' \in X$ such that $R_1 \cup \ldots \cup R_n \subseteq R''$. Hence our derivation of $\phi \to \bot$ is also a derivation from R''. Thus we have found an element of X (namely R'') which is inconsistent$_1$ – contradicting the assumption that X contains only consistent$_1$ rule bases. Thus consistent$_1$ is logically compact. ■

Remark
The proof above clearly goes through independently of the actual rules which belong to *Rules*$_1$. We could just as easily substitute *Rules*$_2$ = *Rules*$_1 \cup \{$OR$\}$: derive $\phi \vee \xi \to \psi$ from $\phi \to \psi$ and $\xi \to \psi$, or *Rules*$_3$ = *Rules*$_1 \cup \{$CT$\}$: derive $\phi \to \varphi$ from $\phi \to \psi$ and $\phi \wedge \psi \to \varphi$, or *Rules*$_4$ = *Rules*$_1 \cup \{$OR, CT$\}$. This means Theorem 3 also holds if we replace consistent$_1$ with consistent$_i$ for any $i \in \{1, 2, 3, 4\}$, where we define R is consistent$_i$ iff $out_i(R, \phi)$ is Cn-consistent for all Cn-consistent $\phi \in L$. This follows from results in (Makinson & van der Torre 2000) which state $\psi \in out_i(R, \phi)$ iff $\phi \to \psi$ is derivable from R using *Rules*$_i$.

Rule implication

Merging operators may merge $\phi \wedge \psi$ and $\neg\phi$ into ψ, which illustrates that merging operators not necessarily select a subset of the formulas from the conflicting sources, like the consolidation operators discussed in the previous section, but they may also contain a formula implied by one of the formulas of the sources. When we want to adapt such an operator for rule merging, we have to define not only the consistency of a set of rules, but also when rules imply other rules.

There are many notions of rule implication in the literature. For example, consider the material implication in propositional logic, thus $\phi \to \psi = \phi \supset \psi$. We have for example that $\phi \supset \psi$ implies $(\phi \wedge \xi) \supset \psi$ and $\phi \supset (\psi \vee \xi)$, or that $\phi \supset \psi$ together with $\psi \supset \xi$ implies $\phi \supset \xi$. Such properties have been studied more systematically traditionally in conditional logic, or more recently in input/output logic (Makinson & van der Torre 2000; Bochman 2005). But these are just examples, and do not directly provide a general solution. In particular, it does not solve the question how to use merging operators for rules defined in a meta-language, in which case we only have an operation $out(R, S)$ defining how to apply a set of rules.

For the general case we propose the following definition of implication among rules. Following the convention in input/output logic, we overload the operator 'out' to refer to this operation too (the two kinds of operations are distinguished by the number of their arguments).

$$out(R) = \{\phi \to \psi \mid \phi = \wedge S, \psi \in out(R, S)\}.$$

Example: merging in possibilistic logic

Prioritized information is represented in possibilistic logic (Dubois, Lang, & Prade 1994) by means of a set of weighted formulas of the form $B = \{(\phi_i, a_i) : i = 1, \cdots, n\}$. The pair (ϕ_i, a_i) means that the certainty (or priority) degree of ϕ_i is at least a_i which belongs to the unit interval $[0, 1]$. A possibility distribution is associated to a possibilistic base as follows: $\forall\omega \in \Omega$,

$$\pi_B(\omega) = \begin{cases} 1 & \text{if } \forall(\phi_i, a_i) \in B, \omega \models \phi_i \\ 1 - max\{a_i : (\phi_i, a_i) \in B \text{ and } \omega \not\models \phi_i\} & \text{otherwise.} \end{cases}$$

When $\pi(\omega) > \pi(\omega')$ we say that ω is preferred to (or more satisfactory than) ω'. For the rest of this section we simplify by assuming our language L is generated by only *finitely* many propositional variables.

A possibilistic base $B = \{(\phi_i, a_i) : i = 1, \cdots, n\}$ is consistent iff the set of propositional formulas $\{\phi_i : (\phi_i, a_i) \in B\}$ associated with B is Cn-consistent.

$\oplus : [0, 1]^k \to [0, 1]$ is a k merging operator when it satisfies the following two conditions. The first condition says that if an alternative is fully satisfactory for all agents then it will be also fully satisfactory w.r.t. the result of merging. The second condition is the monotonicity property.

(i). $\oplus(1, \cdots, 1) = 1$
(ii). $\oplus(a_1, \cdots, a_n) \geq \oplus(b_1, \cdots, b_n)$ if $a_i \geq b_i$ for all $i = 1, \cdots, n$.

For example, let $B_1 = \{(\phi_i, a_i) : i = 1, \cdots, n\}$ and $B_2 = \{(\psi_j, b_j) : j = 1, \cdots, m\}$ be two possibilistic bases. Using $\oplus$, the result of merging B_1 and B_2, written as $\mathcal{B}_\oplus$, is defined as follows (Benferhat *et al.* 1999):

$$\begin{aligned} \mathcal{B}_\oplus = \; & \{(\phi_i, 1 - \oplus(1 - a_i, 1)) : (\phi_i, a_i) \in B_1\} \\ & \cup\{(\psi_j, 1 - \oplus(1, 1 - b_j)) : (\psi_j, b_j) \in B_2\} \\ & \cup\{(\phi_i \vee \psi_j, 1 - \oplus(1 - a_i, 1 - b_j))\}. \end{aligned} \quad (1)$$

We suppose that the bases (possibilistic bases in this section and rule bases in the following sections) are *individually consistent*. Inconsistency occurs after their merging.

Given $\mathcal{B}_\oplus$, the *useful* result of merging – from which inferences are drawn – is defined as a subset of $\mathcal{B}_\oplus$ composed of the consistent most prioritized formulas of $\mathcal{B}_\oplus$, as far as possible. Formally we have:

Definition 7 (Useful result of merging) *Let $\mathcal{B}_\oplus$ be the result of merging $B_1, \cdots, B_n$ using $\oplus$. Let $\mathcal{B}_{\oplus_{\geq a}} = \{\phi_i : (\phi_i, a_i) \in \mathcal{B}_\oplus, a_i \geq a\}$ and $Inc(\mathcal{B}_\oplus) = max\{a_i : (\phi_i, a_i) \in \mathcal{B}_\oplus, \mathcal{B}_{\oplus_{\geq a_i}}$ is Cn-inconsistent$\}$. The useful part of $\mathcal{B}_\oplus$ is:*

$$\rho(\mathcal{B}_\oplus) = \{(\phi_i, a_i) : (\phi_i, a_i) \in \mathcal{B}_\oplus, a_i > Inc(\mathcal{B}_\oplus)\}.$$

Another more qualitative representation of a possibilistic base has been studied in possibilistic logic, based on a well ordered partition of formulas (so without explicit weights!) $\mathcal{B} = B_1; \cdots; B_n$, where formulas of B_i are prioritized over formulas of B_j for $i < j$.

Let $\mathcal{B} = B_1; \cdots; B_n$ and $\mathcal{B}' = B'_1; \cdots; B'_m$. The useful result of merging $\mathcal{B}$ and $\mathcal{B}'$ using the $\min$ operator, written as $\mathcal{B}_{\min}$, is $B_{\min,1}; \cdots; B_{\min,max(n,m)}$, where $B_{\min,i} = (B_i \cup B'_i)$ if $\bigcup_{1 \leq j \leq i}(B_j \cup B'_j)$ is Cn-consistent, empty otherwise.

The useful result of merging $\mathcal{B}$ and $\mathcal{B}'$ using the $\max$ operator is $B_{\max,1}; \cdots; B_{\max,max(n,m)}$, where $B_{\max,i} =$

$(\cup_{\phi \in B_i, \psi \in B'_1 \cup \cdots \cup B'_i}(\phi \vee \psi)) \cup (\cup_{\phi \in B_1 \cup \cdots \cup B_i, \psi \in B'_i}(\phi \vee \psi))$ with B_i (resp. B'_i) is composed of tautology when $i > n$ (resp. $i > m$).

A possibility distribution π can also be written under a well ordered partition, of the set of all possible worlds Ω, of the form $(E_1, \cdots, E_n)$ such that

- $E_1 \cup \cdots \cup E_n = \Omega$,
- $E_i \cap E_j = \emptyset$ for $i \neq j$,
- $\forall \omega, \omega' \in \Omega$, $\omega \in E_i, \omega' \in E_j$ with $i < j$ iff $\pi(\omega) > \pi(\omega')$.

Rules in possibilistic logic

The qualitative representation of a possibilistic base is a particular kind of rules, using Algorithm 1 to compute the possibility distribution associated with a set of rules (Pearl 1990; Benferhat, Dubois, & Prade 1992). Let $R = \{\phi_i \to \psi_i : i = 1, \cdots, n\}$, and let $\mathcal{C} = \{(L(C_i), R(C_i)) : L(C_i) = Mod(\phi_i \wedge \psi_i), R(C_i) = Mod(\phi_i \wedge \neg\psi_i), \phi_i \to \psi_i \in R\}$, where $Mod(\xi)$ is the set of worlds satisfying ξ.

Algorithm 1: Possibility distribution associated with a rule base.

begin
  $l \leftarrow 0$;
  while $\Omega \neq \emptyset$ **do**
    – $l \leftarrow l + 1$;
    – $E_l = \{\omega : \forall (L(C_i), R(C_i)) \in \mathcal{C}, \omega \notin R(C_i)\}$;
    if $E_l = \emptyset$ **then**
      Stop (inconsistent rules);
      $E_l = \Omega$;
    – $\Omega = \Omega - E_l$;
    – $\mathcal{C} = \mathcal{C} \backslash \{(L(C_i), R(C_i)) : L(C_i) \cap E_l \neq \emptyset\}$;
  return $(E_1, \cdots, E_l)$
end

Algorithms have been defined to translate one representation into another. For example, Algorithm 2 translates a set of rules into a possibilitic base given in a well ordered partition (Benferhat, Dubois, & Prade 2001).

Algorithm 2: Translating R into B.

begin
  $m \leftarrow 0$;
  while $R \neq \emptyset$ **do**
    – $m \leftarrow m + 1$;
    – $A = \{\phi_k \supset \psi_k : \phi_k \to \psi_k \in R\}$;
    – $H_m = \{\phi_k \supset \psi_k : \phi_k \to \psi_k \in R, A \cup \{\phi_k\}$ is Cn-consistent$\}$;
    – **if** $H_m = \emptyset$ **then** Stop (inconsistent rules);
    – $R = R \backslash \{\phi_k \to \psi_k : \phi_k \supset \psi_k \in H_m\}$;
  return $\Sigma = \Sigma_1; \cdots; \Sigma_n$ *s.t.* $\Sigma_i = H_{m-i+1}$.
end

Let R be a set of rules and $\Sigma = \Sigma_1; \cdots; \Sigma_n$ be its associated possibilistic base using Algorithm 2. We define a stratification of R as $R = R_1; \cdots; R_n$ with $R_i = \{\phi_k \to \psi_k : \phi_k \supset \psi_k \in \Sigma_i\}$. This stratification of R will be used in the next section.

Moreover, the associated rule base of $\Sigma = \Sigma_1; \cdots; \Sigma_n$ is (Benferhat *et al.* 2001):

$$R = \{\top \to \Sigma_n,$$
$$\neg\Sigma_{n-1} \vee \neg\Sigma_n \to \Sigma_{n-1},$$
$$\cdots,$$
$$\neg\Sigma_1 \vee \neg\Sigma_2 \to \Sigma_1\},$$

where $\Sigma_i = \bigwedge_{\phi \in \Sigma_i} \phi$.

Merging rules in possibilistic logic

For the particular kind of rules defined in possibilistic logic, we can thus define a merging operator as follows. Given a set of rules, transform the set of rules to a possibilistic base. Then apply a merging operator from possibilistic logic. Finally, transform the useful part of the merged base back into a set of rules.

Definition 8 *Let $\mathcal{R}$ and $\mathcal{R}'$ be two rule bases. The result of merging $\mathcal{R}$ and $\mathcal{R}'$ using the* $\min$ *operator, written as $R_{\min}$, is obtained by translating R and R' to B and B' using Algorithm 2, merging B and B' according to the* $\min$ *operator, and translating the useful result of merging back into a set of rules. The result of merging $\mathcal{R}$ and $\mathcal{R}'$ using the* $\max$ *operator is defined analogously.*

Instead of this indirect way, we also define the merger directly. We consider again the min and max mergers.

Definition 9 *Let R and R' be two rule bases, and let $R_1; \ldots; R_n$ and $R'_1; \ldots; R'_m$ be their stratifications according to Algorithm 2. Let $R[k]$ be the set of rules in the first k equivalence classes, $\bigcup_{i=1\ldots k}(R_i \cup R'_i)$. The merger of R and R' according to* $\min$*, written as $R_{\min}$, is $R[k]$ such that $\{\phi_l \supset \psi_l : \phi_l \to \psi_l \in R[k]\}$ is Cn-consistent, and k is maximal.*

Definition 10 *Let R and R' be two rule bases. The merger of R and R' according to* $\max$*, written as $R_{\max}$, is $\{(\phi \wedge \xi) \to (\psi \vee \varphi) \mid \phi \to \psi \in R, \xi \to \varphi \in R'\}$.*

The direct merging approach is twofold interest. It avoids the different translations and also provides more intuitive results at the syntactic level.

Example 1 *Assume there is only a single rule $\phi \to \psi$ which is merged with the empty base. The indirect approach leads to $\{\top \to (\phi \supset \psi)\}$ and the direct approach leads to $\{\phi \to \psi\}$. The two sets are equivalent in the sense that they lead to the same possibility distribution using Algorithm 1.*

The indirect and direct approaches are in this sense equivalent.

Proposition 2 *Let $\mathcal{R}$ and $\mathcal{R}'$ be two rule bases. Let R_1 (resp. R_2) be the result of merging $\mathcal{R}$ and $\mathcal{R}'$ using the* $\min$ *operator following Definition 8 (resp. Definition 9). Then R_1 and R_2 are equivalent in the sense that they induce the same possibility distribution.*
This result holds for the $\max$ *operator as well.*

The latter example illustrates that the kind of rules studied in possibilistic logic are of a particular kind, and it raises the question how the merging operation can be generalized for arbitrary rules. This is studied in the following section.

Application

We now consider the generalization of this approach for an arbitrary notion of rules. We first introduce the following generalization of the stratification Algorithm 2.

Algorithm 3: Stratification of a rule base R.

begin
  $m \leftarrow 0$;
  while $R \neq \emptyset$ **do**
    – $m \leftarrow m + 1$;
    – $H_m = \{\phi_k \rightarrow \psi_k : \phi_k \rightarrow \psi_k \in R, out(R, \phi_k) \cup \{\phi_k\}$ is Cn-consistent$\}$;
    – **if** $H_m = \emptyset$ **then** Stop (inconsistent base);
    – remove H_m from R;
  return $R = R_1; \cdots ; R_n$ s.t. $R_i = H_{m-i+1}$.
end

That Algorithm 3 is a generalization of Algorithm 2 can be seen by setting $out(R) = \{\phi \supset \psi \mid \phi \rightarrow \psi \in R\}$ in the above.

The following example illustrates two distinct kinds of examples of rule sets.

Example 2 *Consider the following two:*

1. $\top \rightarrow f, d \rightarrow \neg f$
2. $\top \rightarrow \neg f, f \rightarrow w$

Both examples will be stratified in two equivalence classes using the algorithm above, but for completely different reasons. In the first example, "d" causes an inconsistency, and in the second example, "f" is an inconsistency. (the first is the usual kind of specificity in the Tweety example, the second is the contrary-to-duty studied in deontic logic; the first is an exception, the second a violation). The first base is stratified into $\{d \rightarrow \neg f\}; \{\top \rightarrow f\}$ *and the second base is stratified into* $\{f \rightarrow w\}; \{\top \rightarrow \neg f\}$.

Proposition 3 *A set of rules is inconsistent, according to Algorithm 3, when for all rules* $(\phi, \psi) \in R$*, we have that* $out(R, \{\phi\})$ *is* Cn*-inconsistent.*

We can use the proposition to define a merging operator according to the min operator, which is again a selection operator. We therefore can use the same definition as above; clearly it is again a generalization.

Definition 11 *Let* R *and* R' *be two rule bases, and let* $R_1; \ldots ; R_n$ *and* $R'_1; \ldots ; R'_m$ *be their stratifications according to Algo. 3. Let* $R[k]$ *be the set of rules in the first* k *equivalence classes,* $\bigcup_{i=1\ldots k}(R_i \cup R'_i)$*. The merger of* R *and* R' *according to* $\min$*, written as* $R_{\min}$*, is* $R[k]$ *such that* $R[k]$ *is consistent according to Algo. 3, and* k *is maximal.*

For the merging operator based on max, we have to use the notion of implication in out. We simply use the same operator as above.

Definition 12 *Let* R *and* R' *be two rule bases. The merger of* R *and* R' *according to* $\max$*, written as* $R_{\max}$*, is* $\{(\phi \wedge \xi) \rightarrow (\psi \vee \varphi) \mid \phi \rightarrow \psi \in R, \xi \rightarrow \varphi \in R'\}$.

Variations

The product operator in possibilistic logic may be seen as a combination of the min and the max operator, in the sense that the merger contains both selections and disjunction. We conjecture that it can be defined analogously in our generalized setting. There are other ways to generalize Algorithm 2 for an arbitrary notion of rules. If we write $R[\phi] = \{(\xi \rightarrow \psi) \in R | (\phi \leftrightarrow \xi) \in Cn(\emptyset)\}$, we can find alternatives for the relative line of the algorithm, for example:

– $H_m = \{\phi_k \rightarrow \psi_k : \phi_k \rightarrow \psi_k \in R, out(R, \phi_k)\}$ is Cn-consistent$\}$;

– $H_m = \{\phi_k \rightarrow \psi_k : \phi_k \rightarrow \psi_k \in R, out(R \setminus R[\phi_k], \{\phi_k\})$ is Cn-consistent$\}$;

Summary

In this paper we introduce a general framework to study rule merging and develop rule merging operators as a generalization of belief merging operators. We use simple rules defined as pairs of formulas of a base logic, i.e., as conditionals. We distinguish weak consistency of rules only in contexts of the given rules, and strong consistency for all possible consistent contexts. We define a notion of implication among rules based on implication in the base language:

$$out(R) = \{\phi \rightarrow \psi \mid \phi = \wedge S, \psi \in out(R, S)\}.$$

We use the framework to study two examples.

Booth and Richter introduce a merging operator based on a notion of consistency. Using our strong notion of consistency of a set of rules, we define a rule merging operator. For the proof of completeness we use a proof-theoretical characterization. This illustrates a general point: to use belief merging operators for rule merging, we may need to prove some additional properties of the rule system, such as a notion of compactness.

In possibilistic logic, a framework has been proposed to study a variety of merging operators. Since also a kind of rules have been studied in the framework of possibilistic logic, these merging operators can also be used for this particular kind of rules. When generalizing the operators for other kinds of rules, several new issues arise.

Since we considered only two examples of generalizing belief merging operators to rule merging, there are many possible studies for further research. We expect that a study of such examples will lead to a further refinement of our general framework.

Acknowledgements

Thanks are due to the reviewers for some helpful comments.

References

Alchourrón, C.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic* 50:510–530.

Benferhat, S.; Dubois, D.; Prade, H.; and Williams, M. 1999. A practical approach to fusing and revising prioritized belief bases. In *Proc. of EPIA 99*, 222–236.

Benferhat, S.; Dubois, D.; Kaci, S.; and Prade, H. 2001. Bridging logical, comparative and graphical possibilistic representation frameworks. In *Conf. on Symbolic and Quantitative Approaches to Reas. and Uncert.*, 422–431.

Benferhat, S.; Dubois, D.; and Prade, H. 1992. Representing default rules in possibilistic logic. In *Int. Conf. of Principles of Knowledge Rep. and Reas. (KR'92)*, 673–684.

Benferhat, S.; Dubois, D.; and Prade, H. 2001. Towards a possibilistic logic handling of preferences. *Applied Intelligence* 14(3):303–317.

Bochman, A. 2005. *Explanatory non-monotonic reasoning*. World scientific.

Booth, R., and Richter, E. 2005. On revising fuzzy belief bases. *Studia Logica* 80(1):29–61.

Dubois, D.; Lang, J.; and Prade, H. 1994. Possibilistic logic. *In Handbook of Logic in Artificial Intelligence and Logic Programming* pages 439–513.

Gerla, G. 2001. *Fuzzy Logic: Mathematical Tools for Approximate Reasoning*. Kluwer Academic Publishers.

Hansson, S. O. 1999. *A Textbook of Belief Dynamics*. Kluwer Academic Publishers.

Makinson, D., and van der Torre, L. 2000. Input-output logics. *Journal of Philosophical Logic* 29:383–408.

Makinson, D., and van der Torre, L. 2001. Constraints for input-output logics. *Journal of Philosophical Logic* 30(2):155–185.

Pearl, J. 1990. System Z: A natural ordering of defaults with tractable applications to default reasoning. In Eds, R. P., ed., *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge (TARK'90)*, 121–135. Morgan Kaufmann.