Syntactic representations of semantic merging operations

Author names here.

1 Introduction

Intelligent agents have to be able to take differing points of view into account to produce a coherent view of the world. In the artificial intelligence literature this is *merging* or *fusion* [8, 10, 2, 1, 11, 3]. A promising approach to merging is to work on the level of *epistemic states*, where an epistemic state contains the beliefs of an agent, as well as an indication of the strength of these beliefs. In this paper we discuss syntactic representations of epistemic states in terms of two types of structures: *ranked knowledge bases* and *partitions*. Ranked knowledge bases, or R-bases, are discussed in section 2 and partitions in section 3. Section 4 is the heart of the paper. First, it provides a uniform method for obtaining syntactic representations of a large class of semantic merging operations in terms of partitions which is linear in n times the product of the sizes of the n partitions used in the merging process. Next, some semantic merging constructions proposed in the literature are considered [2, 10]. It is shown that for some of these we can obtain syntactic representations directly in terms of R-bases which are all linear in the sum of the sizes of the n R-bases to be merged. For some, however, it is shown that we can do no better than the general method in terms of computational complexity. Section 5 looks at related work and concludes.

For the rest of this paper we assume a finitely generated propositional language L closed under the usual propositional connectives and with a classical model-theoretic semantics. Classical entailment is denoted by $\models$ and U is the set of interpretations of L. For $X \subseteq L$, $M(X)$ is the set of models of X. For $\alpha \in L$ we write $M(\alpha)$ instead of $M(\{\alpha\})$. For a non-empty subset X of L we let $\bigwedge X = \bigwedge_{\alpha \in X} \alpha$ and $\bigvee X = \bigvee_{\alpha \in X} \alpha$. Examples involving interpretations are phrased in the language with two atoms, p and q. Interpretations are represented as sequences of 0s and 1s representing falsity and truth respectively. The convention is that the first digit represents the truth value of p and the second one the truth value of q. We denote the successor function for a finite subset of the natural numbers by *suc* and the predecessor function by *pred*.

2 Ranked knowledge bases

In the spirit of [10] we think of an epistemic state as a plausibility ranking of natural numbers on U; the lower the number assigned to an interpretation, the more plausible it is deemed to be. This is along the line of work initially proposed in [14].

Definition 2.1 An *epistemic state* Φ is a function from U to $\mathbb{N}$. ■

For an epistemic state Φ, let $M^i(\Phi) = \{u \mid \Phi(u) = i\}$ and let $\min(\Phi) = \min\{\Phi(u) \mid u \in U\}$.

Definition 2.2 The *knowledge base extracted from* Φ is some wff $B(\Phi)$ such that $M(B(\Phi)) = M^{\min(\Phi)}(\Phi)$. ■

By extracting knowledge from an epistemic state in this way we ensure that $B(\Phi)$ is always satisfiable, thereby circumventing the problems associated with inconsistent knowledge bases. It also shows that what matters is not the particular plausibility ranks assigned to interpretations, but rather the *difference* between plausibility ranks.

One way to represent epistemic states syntactically is in terms of ranked knowledge bases.

Definition 2.3 A *ranked sentence* is a pair (α, r) where $\alpha \in L$ and $r \in \mathbb{N}$. A *ranked knowledge base* or *R-base* RB is a finite set of ranked sentences such that $r > 0$ for every $(\alpha, r) \in RB$. ■

Intuitively (α, r) asserts that α has an entrenchment rank of at least r (where a higher rank is better). The choice of the phrase "entrenchment ranking" is not a coincidence. Associated with every R-base is an epistemic entrenchment ordering in the sense of [6], with an appropriate rank assigned to each wff in L. Space limitations prevent us from providing more details here.

R-bases are similar to the possibilistic bases used in possibilistic logic [5] and the relationship between R-bases and epistemic states are similar to the relationship between possibilistic bases and possibility distributions [2]. It is not difficult to perform translations between the two frameworks.

For a set of ranked sentences R, let $B^i(R) = \{\alpha \mid (\alpha, i) \in R\}$ and let $\min(R) = \min\{i \mid \bigcup_{j \geq i} B^j(R) \not\models \bot\}$. Observe that the notation $B^i(R)$ can be used with any set of ranked sentences, not just with R-bases. In section 3 it will be applied to *partitions* as well.

Definition 2.4 The *knowledge base extracted from* an R-base RB is
$B(RB) = \bigcup_{j \geq \min(RB)} B^j(RB)$. ■

Observe that the knowledge base extracted from an R-base will always be consistent. As is the case with epistemic states, the notion of extraction shows that it is the *difference* in the entrenchment ranks of wffs that are important, not the entrenchment ranks themselves.

Every R-base is associated with a unique epistemic state.

Definition 2.5 The *epistemic state extracted from* an R-base RB is defined as follows,

$$\Phi_{RB}(u) = \begin{cases} 0 \textit{ if } u \in M(\alpha)\ \forall (\alpha, r) \in RB, \\ \max\{r \mid (\alpha, r) \in RB \textit{ and } u \in M(\neg\alpha)\} \\ \qquad \textit{otherwise} \end{cases}$$

■

Example 2.1 Let $RB = \{(p, 3), (q, 2)\}$. Then $\Phi_{RB}(00) = \Phi_{RB}(01) = 3$, $\Phi_{RB}(10) = 2$ and $\Phi_{RB}(11) = 0$. ■

It can be shown that the knowledge base extracted from an R-base and its extracted epistemic state are logically equivalent. That is, for every R-base RB, $B(RB) \equiv B(\Phi_{RB})$.

Example 2.2 Let $RB = \{(p, 3), (q, 2), (\neg p, 1)\}$. Then it follows that $\Phi_{RB}(00) = \Phi_{RB}(01) = 3$, $\Phi_{RB}(10) = 2$ and $\Phi_{RB}(11) = 1$. Furthermore, $B(RB) = \{p, q\}$ and $M(B(\Phi_{RB})) = \{11\}$. ■

2.1 Normal forms for R-bases

Every epistemic state can be extracted from some R-base, but the same epistemic state may be extracted from different R-bases. It is thus useful to have normal form representations of R-bases.

Definition 2.6 An R-base RB is in *normal form* iff for every $i \in \mathbb{N}$ it contains at most one ranked sentence with rank i, and for every $(\alpha, i), (\beta, j) \in RB$ such that $i < j$, $\alpha \vDash \beta$. ∎

Example 2.3 The R-base $\{(p \vee q, 2), (p, 1), (q, 2)\}$ is not in normal form, but the R-base $\{(q, 2), (p \wedge q, 1)\}$ is. ∎

Every epistemic state can be extracted from an R-base in normal form. For an epistemic state Φ let $\mathcal{I}(\Phi) = \{\Phi(u) \mid u \in U\}$ and let $M^i_<(\Phi) = \bigcup_{j=1}^{i-1} M^j(\Phi)$.

Proposition 2.1 *For an epistemic state Φ, let $RB = \bigcup_{i \in \mathcal{I}(\Phi) \setminus \{0\}} \{(\beta_i, i)\}$ where $M(\beta_i) = M^i_<(\Phi)$. Then RB is in normal form and $\Phi_{RB} = \Phi$.*

Example 2.4 Let $\Phi(11) = 0$, $\Phi(10) = \Phi(01) = 1$ and let $\Phi(00) = 3$. Then Φ can be extracted from $RB = \{(p \vee q, 3), (p \wedge q, 1)\}$. ∎

Every R-base can be converted into normal form.

Definition 2.7 An R-base RB' is a *normal form for* an R-base RB iff RB' is in normal form and $\Phi_{RB} = \Phi_{RB'}$. ∎

For a set of ranked sentences R, let $\mathcal{I}(R) = \{i \mid (\alpha, i) \in R\}$. Now, for an R-base RB, let *suc* be the successor function for $\mathcal{I}(RB)$. Then $N(RB) = \bigcup_{i \in \mathcal{I}(RB)} \{(\beta_i, i)\}$, where

$$\beta_i = \begin{cases} \bigwedge B^i(RB) \text{ if } i = \max(\mathcal{I}(RB)) \\ \bigwedge B^i(RB) \wedge \beta_{suc(i)} \text{ otherwise.} \end{cases}$$

It is easy to see that $N(RB)$ is a normal form for RB.

Proposition 2.2 $\Phi_{RB} = \Phi_{N(RB)}$ *for every R-base RB. Determining $N(RB)$ is linear in the size of RB.*

Example 2.5 Let $RB = \{(p, 1), (q, 1), (r, 4)\}$. Then $N(RB) = \{(r, 4), (r \wedge p \wedge q, 1)\}$. ∎

An R-base in normal form can contain redundant information. For example, it can be verified that the same epistemic state is extracted from the R-bases $\{(\top, 4), (p, 3), (p, 2)\}$ and $\{(p, 3)\}$. It is useful to have a normal form that is a compact representation of the information contained in an R-base.

Definition 2.8 An R-base is in *pure normal form* iff its ranked sentences do not contain any tautologies or logically equivalent wffs. An R-base RB' is a *pure normal form for* an R-base RB iff RB' is in pure normal form and it is a normal form for RB. ∎

An R-base in pure normal form has a cardinality that is no bigger than that of any R-base from which the same epistemic state can be extracted.

Every R-base can be converted to one in pure normal form. If RB is in normal form, let $PN(RB)$ be obtained from RB as follows: First remove all ranked sentences containing tautologies. Then, for any set of ranked sentences containing logically equivalent wffs, retain only the one with the highest entrenchment rank. It is easy to see that $PN(RB)$ is a pure normal form for RB whenever RB is in normal form.

Proposition 2.3 $\Phi_{RB} = \Phi_{PN(RB)}$ *for every R-base RB in normal form.*

Example 2.6 Consider the R-base $RB = \{(\top, 5), (p \to q, 4), (\neg p \vee q, 2), (q, 1)\}$. Observe that RB is in normal form. Then $PN(RB) = \{(p \to q, 4), (q, 1)\}$. It can be verified that the epistemic state Φ extracted from RB and $PN(RB)$ is such that $\Phi(11) = \Phi(01) = 0, \Phi(00) = 1$ and $\Phi(10) = 4$. ■

A pure normal form for an R-base is unique up to logical equivalence.

Proposition 2.4 *Let RB' and RB'' be two pure normal forms for RB. Then $\mathcal{I}(RB') = \mathcal{I}(RB'')$ and $\alpha \equiv \beta$ for every $(\alpha, i) \in RB'$ and $(\beta, i) \in RB''$. Converting an R-base in normal form into pure normal form involves, at worst, $|N(RB)| - 1$ checks for logical entailment.*

3 Partitioning

From a semantic point of view even R-bases in pure normal form will always contain redundant information because of the requirement of logical entailment between the wffs contained in it. In this section we provide a representation of R-bases which eliminates such redundancy.

Definition 3.1 A *partition* P is a finite set of ranked sentences with no entrenchment rank occurring more than once. P is a *partition for an epistemic state* Φ iff for every $i \in \mathcal{I}(P)$, $M^i(\Phi) = M(B^i(P))$. (Recall that $B^i(P) = \{\alpha \mid (\alpha, i) \in P\}$.) P is a *partition for an R-base* RB iff it is a partition for Φ_{RB}, and P is a *pure partition* iff none of the ranked sentences in P contains a contradiction. ■

The models of a ranked sentence (α, i) in a partition for an epistemic state Φ are exactly those in the equivalence class of interpretations assigned the value i by Φ; hence the use of the word "partition".

Observe that both partitions and R-bases are sets of ranked sentences. For the rest of the paper we shall use RB and P, possibly subscripted, to denote R-bases and partitions respectively.

Every R-base can be converted into a partition for itself. Let $RB = \bigcup_{i \in \mathcal{I}(RB)} \{(\beta_i, i)\}$ be an R-base in normal form and let suc be the successor function for $\mathcal{I}(RB)$. Without loss of generality we assume that $\beta_i \vDash \beta_{suc(i)}$ for every $i \in \mathcal{I}(RB) \setminus \{\max(\mathcal{I}(RB))\}$. Let $P(RB) = \bigcup_{i \in \mathcal{I}(RB) \cup \{0\}} \{(\gamma_i, i)\}$ where:

$$\gamma_i = \begin{cases} \beta_{\min(\mathcal{I}(RB))} \textit{ if } i = 0, \\ \neg\beta_i \textit{ if } i = \max(\mathcal{I}(RB)), \\ \neg\beta_i \wedge \beta_{suc(i)} \textit{ otherwise.} \end{cases}$$

Proposition 3.1 *If RB is an R-base in normal form then $P(RB)$ is a partition for RB. Determining $P(RB)$ is linear in the size of RB.*

Example 3.1 Let $RB = \{(p \vee q, 5), (p, 4), (p, 2)\}$. Observe that RB is in normal form. Then $P(RB) = \{(p, 0), (\neg p \wedge p, 2), (\neg p \wedge (p \vee q), 4), (\neg(p \vee q), 5)\}$. It can be verified that $\Phi_{RB}(00) = 5$, $\Phi_{RB}(01) = 4$, and $\Phi_{RB}(10) = \Phi_{RB}(11) = 0$. Furthermore, it is easy to see that $M(p) = M^0(\Phi_{RB}) = \{10, 11\}$, $M(\neg p \wedge p) = M^2(\Phi_{RB}) = \emptyset$, $M(\neg p \wedge (p \vee q)) = M^4(\Phi_{RB}) = \{01\}$ and $M(\neg(p \vee q)) = M^5(\Phi_{RB}) = \{00\}$. ■

Once an R-base is in pure normal form, obtaining a pure partition for it is a matter of obtaining a partition for it and then doing one check for logical entailment.

Proposition 3.2 *Let RB be an R-base in pure normal form. If $\alpha \vDash \bot$ for some $(\alpha, i) \in RB$, then $P(RB) \setminus \{(\alpha, 0)\}$ is a pure partition for RB. Otherwise $P(RB)$ is a pure partition for RB.*

Example 3.2 Let $RB = \{(p \vee q, 5), (p, 4), (\bot, 2)\}$. Observe that RB is in pure normal form. It can be verified that $P(RB) \setminus \{(\bot, 0)\} = \{(\neg\bot \wedge p, 2), \neg p \wedge (p \vee q), 4), (\neg(p \vee q), 5)\}$ is a pure partition for RB. ■

An R-base can thus easily and efficiently be converted into a partition for itself. The converse is also true. F or a partition P, let *pred* and *suc* be the predecessor and successor functions for $\mathcal{I}(P)$ respectively, and let

$$RB(P) = \begin{cases} \bigcup_{i \in \mathcal{I}(P) \setminus \{\max(\mathcal{I}(P))\}} \{(\gamma_i, suc(i))\} \\ \quad \textit{if } (\alpha, 0) \in P \textit{ for some } \alpha, \\ \bigcup_{i \in \mathcal{I}(P) \setminus \{\max(\mathcal{I}(P))\}} \{(\gamma_i, suc(i))\} \cup \\ \{(\bot, \min(\mathcal{I}(P))\} \textit{ otherwise,} \end{cases}$$

where

$$\gamma_i = \begin{cases} B^i(P) \textit{ if } i = \min(\mathcal{I}(P)), \\ B^i(P) \vee \gamma_{pred(i)} \textit{ otherwise.} \end{cases}$$

In the definition of the γ_is above we abuse notation by treating the singleton set $B^i(P)$ as a wff. We shall do so again.

Proposition 3.3 *If P is a partition for an R-base RB then $RB(P)$ is a normal form for RB. Determining $RB(P)$ is linear in the size of P.*

Example 3.3 Let $RB = \{(p \vee q, 5), (p, 4), (\bot, 2)\}$. $P = \{(\neg p \wedge \neg q, 5), (\neg p \wedge q, 4), (p, 2)\}$ is a partition for RB and $RB(P) = \{(\bot, 2), (p, 4), (p \vee (\neg p \wedge q), 5)\}$ is a normal form for RB. ■

4 Merging R-bases

[10] provides a semantic framework for the merging of epistemic states and proposes a number of semantic merging operations. In his framework a semantic merging operation maps an *epistemic list* (a finite non-empty list of epistemic states) to an epistemic state. Intuitively, the epistemic states represent the information provided by various sources to an agent which merges them into a single epistemic state.

Definition 4.1 A semantic merging operation Δ is a function from the set of finite sequences of natural numbers to $\mathbb{N}$. For an epistemic list $E = [\Phi_1, \ldots, \Phi_n]$ and, for every $u \in U$, let $s^E(u)$ be the sequence $\Phi_1(u), \ldots, \Phi_n(u)$. $\Delta(s^E(u))$ represents the plausibility rank assigned to the interpretation u by the epistemic state obtained when merging is applied to E. ■

By adopting the position set out in definition 4.1 we commit ourselves to the study of semantic merging operations which satisfy a property referred to as (Unit) in [10]. Indeed, all the operations considered in the remainder of the paper satisfy (Unit). With Δ viewed as a function from epistemic lists to epistemic states, (Unit) can be phrased as follows:

(Unit) If $\Phi_i(u) = \Phi_i(v)$ for every $i \in \{1, \ldots, n\}$ then $\Delta(E)(u) = \Delta(E)(v)$

So, if every epistemic state in an epistemic list assigns the same plausibility rank to two interpretations u and v, (Unit) requires of the epistemic state obtained after merging to do the same. Observe that (Unit) is a very weak condition to place on merging operations and therefore does not exclude *any* reasonable merging operation. We conjecture that it is unlikely that any sensible semantic merging-like operation exists which does *not* satisfy (Unit). For the remainder of this paper, then, we operate under the implicit asumption that semantic merging operations satisfy (Unit). It turns out that there is a uniform syntactic description, in terms of partitions, of *all* these semantic merging operations.

4.1 Merging via partitions

Consider a list of partitions $PL = [P_1, \ldots, P_n]$ and its *associated* epistemic list $E = [\Phi_1, \ldots, \Phi_n]$ where P_i is a partition for Φ_i for every $i \in \{1, \ldots, n\}$.

Definition 4.2 An operation δ which maps partition lists to partitions is called a *syntactic version of* a semantic merging operation Δ iff for every partition list PL and its associated epistemic list E, $\Phi(u) = \Delta(s^E(u))$, where $\delta(PL)$ is a partition for Φ. ■

In other words, the epistemic state obtained when applying Δ to E can also be obtained by applying δ to PL and then converting the resulting partition to an epistemic state Φ.

For any finite sequence of natural numbers s, let $s(i)$ denote the ith number in s, and for a partition list PL of length n, let $seq(PL)$ be the set containing every sequence s of natural numbers of length n such that $s(i) \in \mathcal{I}(P_i)$ for every $i \in \{1, \ldots, n\}$. For $s \in seq(PL)$, let $PL(s) = \bigwedge_{i=1}^{i \leq n} B^{s(i)}(P_i)$. (Recall again that $B^{s(i)}(P_i) = \{\alpha \mid (\alpha, s(i)) \in P_i\}$, and observe that we treat the singleton set $B^{s(i)}(P_i)$ as a wff.)

Proposition 4.1 *For every $s \in seq(PL)$, $M(PL(s)) = \{u \mid \Phi_i(u) = s_i$ for every $i \in \{1, \ldots, n\}\}$.*

Proposition 4.1 shows that $PL(s)$ is a syntactic description of the set of interpretations assigned a value of $s(i)$ by Φ_i for every $i \in \{1, \ldots, n\}$.

For a semantic merging operation Δ and for any partition list PL, let

$$PL^{\Delta} = \bigcup_{s \in seq(PL)} \{(PL(s), \Delta(s))\}.$$

So PL^{Δ} assigns to every wff $PL(s)$ the value that Δ assigns to the models of $PL(s)$. Obtaining a syntactic version of Δ is then a matter of turning PL^{Δ} into a partition. This is achieved by conjoining those wffs occurring in PL^{Δ} with the same rank assigned to them. That is, we let $\delta_{\Delta}(PL) = \bigcup_{i \in \mathcal{I}(P^{\Delta})} \{(\bigwedge B^i(P^{\Delta}), i)\}$. The heart of the paper, then, is the next result.

Theorem 4.1 *For every semantic merging operation Δ, δ_{Δ} is a syntactic version of Δ. Determining $\delta_{\Delta}(PL)$ for a partition list PL is linear in n times the product of the sizes of the n partitions (assuming that $\Delta(s)$ can be calculated in constant time).*

Theorem 4.1 allows us to perform merging on R-bases as follows. Given a list of R-bases RL and a semantic merging operation Δ, we find a partition for each R-base in RL and we apply δ_{Δ} to this list of partitions, obtaining a partition from which the appropriate R-base, the merged version of RL, can be obtained.

Example 4.1 Let $RL = [RB_1, RB_2]$ where $RB_1 = \{(p \wedge q, 1), (p, 3)\}$ and $RB_2 = \{(p \wedge \neg q, 2)\}$, and let Δ be a semantic merging operation. It can be verified that $P_1 = \{(p \wedge q, 0), (p \wedge \neg q, 1), (\neg p, 3)\}$ is a partition for RB_1 and $P_2 = \{(p \wedge \neg q, 0), (p \rightarrow q, 2)\}$ is a partition for RB_2. Then $seq([P_1, P_2]) = \{00, 02, 10, 12, 30, 32\}$. Also, $PL(00) = \bigwedge\{p \wedge q, p \wedge \neg q\}$, $PL(02) = \bigwedge\{p \wedge q, p \rightarrow q\}$, $PL(10) = \bigwedge\{p \wedge \neg q\}$, $PL(12) = \bigwedge\{p \wedge \neg q, p \rightarrow q\}$, $PL(30) = \bigwedge\{\neg p, p \wedge \neg q\}$, and $PL(32) = \bigwedge\{\neg p, p \rightarrow q\}$. Therefore $\delta_{\Delta}([P_1, P_2]) = \{((p \wedge q) \wedge (p \wedge \neg q), \Delta(00)), ((p \wedge q) \wedge (p \rightarrow q), \Delta(02)), (p \wedge \neg q, \Delta(10)), ((p \wedge \neg q) \wedge (p \rightarrow q), \Delta(12)), (\neg p \wedge (p \wedge \neg q), \Delta(30)), (\neg p \wedge (p \rightarrow q), \Delta(32))\}$. ■

Theorem 4.1 is significant for two reasons. Firstly, it applies to a very large class of semantic merging operations. And secondly, as we shall see in section 4.3, for at least some semantic merging operations we cannot do any better, in terms of efficiency, than the method provided by theorem 4.1.

4.2 Improving efficiency

One of the ways in which the application of δ_{Δ} might be made more efficient is by working towards the use of pure partitions in the partition list PL. Every ranked sentence which can be removed from a partition in PL before applying δ_{Δ} results in a saving that is linear in n times the product of the remaining $n - 1$ partitions. In cases where the size of the language (the number of atoms) is fairly small relative to the number of sources and the number of ranked sentences in each partition, the benefits of obtaining pure partitions might even outweigh the costs involved. In general, though, obtaining a pure partition is expensive since it involves as many as $|P|$ checks for logical entailment for a partition P. A useful strategy would be to employ an *anytime algorithm*, identifying and removing redundant ranked sentences in a partition only when computing resources are idle.

Another way to increase the efficiency of performing merging is to exploit the particular structure of specific merging constructions. Theorem 4.1 applies to a large class of semantic merging operations and the application of δ_Δ may therefore not always be very efficient. In the rest of this section we consider specific semantic merging operations proposed in the literature and try to determine whether they have syntactic descriptions which are more efficient than δ_Δ.

For a sequence of numbers s, we denote by $|s|$ the number of elements in s. Now, let $\min(s) = \min\{s(i) \mid 1 \leq i \leq |s|\}$, and let $\max(s) = \max\{s(i) \mid 1 \leq i \leq |s|\}$. Consider the following semantic merging operations:

$$\Delta_{\max}(s) = \max(s),$$

$$\Delta_{\min}(s) = \min(s),$$

$$\Delta_{\min1}(s) = \begin{cases} 2\min(s) \textit{ if } s(i) = s(j)\ \forall i, j \in \\ \quad \{1, \ldots, |s|\}, \\ 2\min(s) + 1 \textit{ otherwise}, \end{cases}$$

$$\Delta_{\min2}(s) = \begin{cases} s(i) \textit{ if } s(i) = s(j)\ \forall i, j \in \{1, \ldots, |s|\} \\ \min(s) + 1 \textit{ otherwise}. \end{cases}$$

$\Delta_{\max}$ was proposed in [10] as a generalisation of a proposal in [8]. In [2], which employs a possibilistic logic setting, $\Delta_{\max}$ is referred to as *idempotent conjunction* and $\Delta_{\min}$ as *disjunctive aggregation.* $\Delta_{\min1}$ and $\Delta_{\min2}$ are variations of $\Delta_{\min}$. $\Delta_{\min1}$ was proposed in [10] where it is referred to as $\Delta_{\min}$. As far as we know $\Delta_{\min2}$ has not been proposed elsewhere. With the exception of $\Delta_{\min}$, all these merging operations satisfy the basic postulates for merging proposed in [10].

For the above group of semantic merging operations it is possible to find more efficient syntactic versions directly in terms of R-bases. Consider a list of R-bases $RL = [RB_1, \ldots, RB_n]$ and its *extracted* epistemic list $E = [\Phi_{RB_1}, \ldots, \Phi_{RB_n}]$.

Definition 4.3 An operation ρ which maps R-base lists to R-bases is called a *syntactic version of* a semantic merging operation Δ iff for every R-base list RL and its extracted epistemic list E, $\Phi_{\rho(RL)}(u) = \Delta(s^E(u))$. (Recall that $s^E(u)$ is the sequence $\Phi_{RB_1}(u), \ldots, \Phi_{RB_n}(u)$.) ■

In other words, the epistemic state obtained when applying Δ to E can also be obtained by applying ρ to RL and then extracting an epistemic state from the resulting R-base.

For an R-base list $RL = [RB_1, \ldots, RB_n]$, let $RL^i = \{\alpha \mid (\alpha, i) \in RB_j$ for some RB_j in $RL\}$, let $\mathcal{I}(RL) = \bigcup_{i=1}^{i \leq n} \mathcal{I}(RB_i)$, and let *pred* and *suc* be the predecessor and successor functions for $\mathcal{I}$(RL) respectively. Now, let

$\rho_{\max}(RL) = \bigcup_{i \in \mathcal{I}(PL)} \{(\gamma_i, i)\}$ where

$$\gamma_i = \begin{cases} \bigwedge RL^i \textit{ if } i = \max(\mathcal{I}(RL)), \\ \bigwedge RL^i \wedge \gamma_{suc(i)} \textit{ otherwise}, \end{cases}$$

$\rho_{\min}(RL) = \bigcup_{i \in \mathcal{I}(PL)} \{(\gamma_i, i)\}$ where

$$\gamma_i = \begin{cases} \bigvee RL^i \ \textit{if} \ i = \min(\mathcal{I}(RL)), \\ \bigvee RL^i \vee \gamma_{pred(i)} \ \textit{otherwise}, \end{cases}$$

Proposition 4.2 *$\rho_{\max}$ and $\rho_{\min}$ are syntactic versions of $\Delta_{\max}$ and $\Delta_{\min}$, respectively, when applied to R-base lists in normal form. The determination of both $\rho_{\max}$ and $\rho_{\min}$ is linear in the sum of the sizes of the n R-bases.*

Syntactic versions of $\Delta_{\min 1}$ and $\Delta_{\min 2}$ are described most elegantly in terms of $\rho_{\min}$ and $\rho_{\max}$. First, recall that for any set of ranked sentences R, $B^i(R) = \{\alpha \mid (\alpha, i) \in R\}$. Now, let

$\rho_{\min 1}(RL) = \bigcup_{i \in \mathcal{I}(RL)} \{(\gamma_{2i-1}, 2i-1), (\gamma_{2i}, 2i)\}$ where

$$\gamma_{2i-1} = \begin{cases} B^i(\rho_{\max}(RL)) \ \textit{if} \ i = \min(\mathcal{I}(RL)), \\ B^i(\rho_{\max}(RL)) \vee B^{pred(i)}(\rho_{\min}(RL)) \\ \qquad \textit{otherwise}, \end{cases}$$

and $\gamma_{2i} = B^i(\rho_{\min}(RL))$,

$\rho_{\min 2}(RL) = \bigcup_{i \in \mathcal{I}(RL)} \{(\gamma_i, i)\}$ where

$$\gamma_i = \begin{cases} B^i(\rho_{\max}(RL)) \ \textit{if} \ i = \min(\mathcal{I}(RL)), \\ B^i(\rho_{\max}(RL)) \vee B^{pred(i)}(\rho_{\min}(RL)) \\ \qquad \textit{otherwise}. \end{cases}$$

Proposition 4.3 *ρ_{min1} and ρ_{min2} are syntactic versions of Δ_{min1} and Δ_{min2}, respectively, when applied to R-base lists in normal form. The determination of both ρ_{min1} and ρ_{min2} is linear in the sum of the sizes of the n R-bases.*

4.3 Bounds on efficiency

The next group of semantic merging operations we consider all assume the existence of a *worst* level of plausibility, say Pl, that may be assigned to any interpretation. In practice this is necessary anyway to ensure *commensurability*, the assumption that all sources of information use the same *scale* when deciding on the plausibility of interpretations. An easy way to achieve this is to require of each source s to specify its own personal worst level of plausibility Pl_s. Such a commitment does not mean that s *has* to rank at least one interpretation at Pl_s; it simply means that this is the worst level of plausibility that s would ever consider attributing to any interpretation. Allowing for personal levels of worst plausibility ensures that a source is able to decide on its own preferred level of granularity. A commonly agreed upon worst level of plausibility Pl is then easily obtained. For example, simply take Pl to be the product of the personal worst levels of plausibility of all sources and scale the rankings supplied by the different sources accordingly. For any finite set sq of sequences of natural numbers we define the function $\Omega^{sq} : sq \rightarrow \{0, \ldots, |sq|-1\}$ by assigning consecutive natural numbers to the elements of sq in the order imposed by the *lexicographic* *ordering* on it, starting by assigning 0 to the element lowest down in the lexicographic ordering.

For any positive integer n, we let $sq(n)$ be the set of all sequences of length n of natural numbers ranging from 0 to Pl. We let $sq_{\leq}(n)$ be the subset of elements of $sq(n)$ whose numbers are ordered non-decreasingly, and we let $sq_{\geq}(n)$ be the subset of elements of $sq(n)$ whose numbers are ordered non-increasingly. For any sequence of numbers s, we let $s_{\leq}$ be the sequence obtained by ordering the numbers occurring in s non-decreasingly, and we let $s_{\geq}$ be the sequence obtained by ordering the numbers occurring in s non-inecreasingly. Now consider the following semantic merging operations defined as functions from $\bigcup_{n>0} sq(n)$ to $\mathbb{N}$:

$$\Delta_{\text{lex}}(s) = \Omega^{sq(|s|)}(s),$$

$$\Delta_{\text{Rmin}}(s) = \Omega^{sq_{\leq}(|s|)}(s_{\leq}),$$

$$\Delta_{\text{Rmax}}(s) = \Omega^{sq_{\geq}(|s|)}(s_{\geq}), \text{ and}$$

$$\Delta_{\Sigma}(s) = \Sigma_{i=1}^{i\leq n} s_i.$$

These merging operations were proposed in [10] and they all satisfy his basic postulates for semantic merging. Constructions similar to Δ_{lex}, Δ_{Rmax} and Δ_{Σ} are also proposed and discussed in [2, 1].

Δ_{lex} assumes a total order of reliability on epistemic states in an epistemic list, with epistemic states occurring earlier in the list having a higher reliability. It ensures that an epistemic state with a high reliability completely dominates those with lower reliability. This is a generalisation of a proposal in [12]. In the literature on social choice theory it is known as a *dictator* operation [7]. It can also be described directly as follows:

$$\Delta_{\text{lex}}(s) = \Sigma_{i=1}^{i\leq|s|} \left(s_i \cdot (Pl+1)^{|s|-i}\right)$$

It turns out that the computational complexity of any syntactic version of Δ_{lex} will be, at best, of the same order as $\delta_{\Delta_{\text{lex}}}$, the operation provided by theorem 4.1. In essence, this is due to the fact that, when restricted to sequences of a fixed length, Δ_{lex} is an injective function. Space limitations prevent us from providing a more detailed explanation here.

Δ_{Rmax} is a generalisation of a merging operation proposed in [8] and is a refinement of $\Delta_{\max}$. Δ_{Rmin} can be seen as a refinement of any one of $\Delta_{\min}$, $\Delta_{\min 1}$ and $\Delta_{\min 2}$. At present we do not know of any syntactic versions of these two semantic merging operations that are more efficient than $\delta_{\Delta_{\text{Rmax}}}$ and $\delta_{\Delta_{\text{Rmin}}}$, the operations provided by theorem 4.1.

Δ_{Σ} is an appropriate generalisation of an example in [9] which was independently proposed in [13]. [10] provides a syntactic version of Δ_{Σ} in terms of lists of wffs known as *infobases* and shows that it is linear in the sum of the sizes of the infobases. It can be shown that converting a partition into an infobase is linear in the size of the partition. Unfortunately the conversion of an infobase back into a partition is of the order $\sum_{i=1}^{i\leq m} \binom{m}{m-i}$, where m is the size of the infobase.

5 Related work and conclusion

[2] deals with syntactic versions of semantic merging operations (or *information fusion* operations, as it is referred to in their paper) on possibility distributions. They represent possibility distributions syntactically in terms of possibilistic bases. We can make fairly direct comparisons between their work and ours because of the relative ease of translation between the two frameworks. Their results only hold for binary merging operations, although it can easily be extended to the more general case for those merging operations that satisfy associativity. Like us, they present a general uniform syntactic method for representing a large class of semantic merging operations. If their method is generalised to deal with n-ary merging operations, it will be linear in n times the product of the sizes of the n possibilistic bases. However, their approach differs substantially from ours and the result holds for a strict subclass of the semantic merging operations we consider.

Like us, [2] also considers specific semantic merging constructions and provides appropriate syntactic versions. But, with the exception of $\Delta_{\max}$, for which a syntactic description is provided which is linear in the sum of the sizes of the possibilistic bases, these syntactic versions are instances of their general method.

[1] is also concerned with syntactic encodings of information fusion operations in the possibilistic setup, but the focus is on a restricted set of possibility distributions; the ones obtained using Dalal's measure of distance between interpretations [4].

In conclusion, then, we have provided a general method for obtaining syntactic versions of a very large class of semantic merging operations. We showed that for some semantic constructions it is possible to obtain syntactic versions which are more efficient than the general method. For at least one specific semantic merging operation, though, it is not possible to find a more efficient syntactic version than the general method provided. As a general tool this method is therefore optimal in terms of computational complexity.

References

[1] Salem Benferhat, Didier Dubois, Souhila Kaci, and Henri Prade. Encoding information in possibilistic logic: A general framework for rational syntactic merging. In Werner Horn, editor, *ECAI 2000. 14th European Conference on Artificial Intelligence*, Amsterdam, 2000. IOS Press.

[2] Salem Benferhat, Didier Dubois, Henri Prade, and Mary-Anne Williams. A Practical Approach to Fusing Prioritized Knowledge Bases. In *Progress in Artificial Intelligence*, volume 1695 of *Lecture Notes In Artificial Intelligence*, pages 222–236, Berlin, 1999. Springer-Verlag.

[3] Salem Benferhat, Souhila Kaci, Daniel Le Berre, and Mary-Anne Williams. Weakening Conflicting Information for Iterated Revision and Knowledge Integration. In Bernhard Nebel, editor, *IJCAI-01. Seventeenth International Joint Conference on Artificial Intelligence*, pages 109–118, San Mateo, CA, 2001. Morgan Kaufmann.

[4] Mukesh Dalal. Investigations into a theory of knowledge base revision. In *Proceedings of the 7th National Conference of the American Association for Artificial Intelligence, Saint Paul, Minnesota*, pages 475–479, 1988.

[5] Didier Dubois, Jerome Lang, and Henri Prade. Possibilistic logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 439–513. 1994.

[6] Peter Gärdenfors. *Knowledge in Flux : Modeling the Dynamics of Epistemic States.* The MIT Press, Cambridge, Massachusetts, 1988.

[7] J. S. Kelly. *Arrow impossibility theorems.* Series in economic theory and mathematical economics. Academic Press, New York, 1978.

[8] Sébastien Konieczny and Ramón Pino-Pérez. On the logic of merging. In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR '98)*, pages 488–498, San Francisco, California, 1998. Morgan Kaufmann.

[9] J. Lin and A. O. Mendelzon. Knowledge base merging by majority. In R. Pareschi and B. Fronhoefer, editors, *Dynamic Worlds: From the Frame Problem to Knowledge Management.* Kluwer Academic Publishers, 1999.

[10] Thomas Meyer. Merging Epistemic States. In Riichiro Mizoguchi and John Slaney, editors, *PRICAI 2000: Topics in Artificial Intelligence*, volume 1886 of *Lecture Notes In Artificial Intelligence*, pages 286–296, Berlin, 2000. Springer-Verlag.

[11] Thomas Meyer, Aditya Ghose, and Samir Chopra. Social Choice, Merging, and Elections. In Salem Benferhat and Philippe Besnard, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 2143 of *Lecture Notes In Artificial Intelligence*, pages 466–477, Berlin, 2001. Springer.

[12] Abhaya C. Nayak. Iterated belief change based on epistemic entrenchment. *Erkenntnis*, 41:353–390, 1994.

[13] P. Z. Revesz. On the Semantics of Theory Change: Arbitration between Old and New Information. In *Proceedings PODS '93, 12th ACM SIGACT SIGMOD SIGART Symposium on the Principles of Database Systems*, pages 71–82, 1993.

[14] Wolfgang Spohn. Ordinal conditional functions: A dynamic theory of epistemic states. In William L. Harper and Brian Skyrms, editors, *Causation in Decision: Belief, Change and Statistics: Proceedings of the Irvine Conference on Probability and Causation: Volume II*, volume 42 of *The University of Western Ontario Series in Philosophy of Science*, pages 105–134, Dordrecht, 1988. Kluwer Academic Publishers.