On the Use of Complex Matrices for Belief Revision

George Flouris and Dimitris Plexousakis

{fgeo,dp}@ics.forth.gr
Institute of Computer Science
Foundation for Research and Technology - Hellas
Science and Technology Park of Crete P.O.Box 1385
GR 711 10 Heraklion, Crete, Greece
Tel: +30(81)39 16 00
Fax: +30(81)39 16 01

Abstract. A most crucial problem in knowledge representation is the revision of knowledge when new, possibly contradictory, information is obtained (belief revision). In this paper, we address this problem for propositional knowledge bases. We introduce a new representation of propositional expressions using 2-dimensional matrices and describe the intuitive processes leading to this representation. We present some theoretical results regarding matrices and their similarity with logic and show how the increased expressiveness of this representation can be exploited for devising a solution to the problem of belief revision in propositional knowledge bases. We also propose a simple method to perform update operations and address the new problems and opportunities for query answering arising under this new representation.

1 Introduction

The problem of revising beliefs is the problem of adapting a given piece of knowledge to accommodate a new piece of information regarding the world being modeled. It is a most crucial problem in knowledge representation, as we generally have to deal with dynamic worlds, where changes are quite frequent. Moreover, we must base our knowledge on information that could be incomplete, or even faulty. Therefore, new updates could contradict the given knowledge, and we should be able to track and remove the contradictions created by each update. This problem is greatly interwoven with the representation of our knowledge; before deciding on a proper belief revision method, we should choose a proper knowledge representation scheme.

The updating methods of knowledge are by no means obvious, even when we are concerned with the intuitive processes only. Let us consider the simple example of knowing a fact α, as well as the proposition $\alpha \rightarrow b$. One obvious implication of the above is the fact b (by modus ponens), which could be inserted into our knowledge base (KB) as a new fact. Let us now consider the negation of b ($\neg b$) entering into the base as a new piece of knowledge (i.e. an update). This contradicts our previous assumption that b is true, so we will have to give up some (or all) of our previous beliefs or we would result in an inconsistent KB. Alternatively, we could reject the update as non-valid. Even in this trivial example, it is not clear which approach should be taken. Extra-logical factors should be taken into account, like the source and reliability of each piece of information or some kind of bias towards or against updates.

This and similar problems have been addressed by several scientists, including philosophers, computer scientists, logicians and others, in an effort to provide us with an intuitively correct method of belief updating. An excellent introductory survey of such efforts by Gärdenfors may be found in [10]. In our paper, we introduce an innovative representation of propositional expressions based on matrices of complex numbers. We devise an elegant solution to the problem of belief revision based on this representation and explore some of its properties as well as the connection of this method with proposed algorithms from the literature. For a short, but partial description of our technique, see [8]; for a more recent and complete one, see [7, 9].

2 Previous Work

One of the first attempts to solve the belief revision problem is due to Fagin, Ullman and Vardi [6]. However, as proven in the same paper, their method forces us to completely abandon the old knowledge whenever the KB is inconsistent with the update. This may be unacceptable in most applications.

Dalal in [2, 3] proposed another, more promising method of revising beliefs. He provided a specific algorithm for updating propositional databases which was based on four basic principles, namely:

1. *Irrelevance of Syntax*: Updating logically equivalent KBs with logically equivalent updates should give logically equivalent update results.
2. *Primacy of New Information*: New knowledge is always assumed more reliable than old knowledge.
3. *Persistence of Prior Knowledge*: As much as possible of the old data should be retained in the KB; we should only retract the minimum knowledge needed to keep the KB consistent.
4. *Fairness*: All things being equal, when we have more than one choice for the result of the update, none of the choices should be made arbitrarily, in order to preserve determinism.

Dalal also formalized the notion of minimal change (third principle) and proved ([3]) that his method retained more knowledge in cases of inconsistent updates than any of the up-to-then proposed algorithms.

An alternative approach was presented by Alchourron, Gärdenfors and Makinson in a series of papers ([1, 10, 16]). Their idea was to recede from the search of any specific algorithm and attempt to formalize the notion of update. As a result, a set of widely accepted properties of belief revision algorithms was introduced, in the form of postulates expressed as a set of logical propositions (named AGM postulates after the authors' initials). Using these postulates, a series of important theoretical results could be proved.

The AGM postulates inspired a series of other works, like [13] by Katsuno and Mendelzon who proposed a different theoretical foundation of update functions by reformulating the AGM postulates in terms of formulas and provided an elegant representation based on orderings of belief sets and [17] by Nebel, who investigated generalizations of the postulates into the knowledge level. Most of the above works deal with the theoretical foundation of the belief revision problem, so they are mostly of theoretical interest. Williams in [18, 19] followed a more practical approach by providing implementations of algorithms based on the AGM paradigm.

3 Properties of Belief Revision

In order to address the problem of belief revision we must first examine some of its properties. One primary consideration is the concurrence of the results with human intuition. This consideration is formally expressed by the principles of Dalal and the AGM postulates. However, it is not absolutely clear how humans revise their beliefs, despite several efforts in the area.

One example of disagreement is the representation of knowledge in the human brain. There are two general types of theories concerning this representation: *foundation* and *coherence* theories ([11]). Foundational theorists argue that knowledge should consist of a set of reasons. According to this theory, knowledge has the form of a pyramid, where only some beliefs (called *foundational*, or *reasons*) can stand by themselves; the rest being derived by the most basic (foundational) beliefs. On the other hand, coherence theorists believe that each piece of knowledge has an independent standing and needs no justification, as long as it does not contradict with other beliefs. Surprisingly, experiments have shown that the human brain actually uses the coherence paradigm ([11]). However, there has been considerable debate on the explanation of the experiments' results. The experiments showed that people tend to ignore causality relationships once a belief has been accepted as a fact, even if this belief has been accepted solely by deduction from other beliefs. The followers of the coherence approach argue that what actually happens is that humans do not actually *ignore* the causality relationships, but *forget* them. The subject will be very willing to reject any beliefs whose logical support no longer exists, but only if he is reminded of this fact.

The approach (foundational or coherence) chosen greatly influences the algorithms considered. Foundational KBs need to store the reasons for beliefs, along with the beliefs themselves, whereas KBs based on the coherence paradigm need to store the set of beliefs only. Reasons should be taken into account when revising a KB only if the foundational approach is selected. The coherence paradigm practically considers all beliefs equal and ignores any causality relationships.

The set of beliefs of any KB includes the derived beliefs. It is generally the case that the derived beliefs are too many, or even infinitely many. This is a serious drawback, so it has been proposed that instead of the whole set of beliefs (*belief set*), a small number of propositions could be stored (*belief base*), enough to reproduce the whole set via deduction. Belief sets are useful theoretic constructions, but cannot be directly used in implementations, due to their size. Belief bases are more useful when it comes to applications. The use of belief bases does not necessarily force us to use the foundational paradigm; the causality relationships possibly implied by the use of the theorem prover that performs the deduction could or could not be used, depending on the approach. The use of belief bases gives rise to another problem which is the selection of the belief base. In general, a given belief set can be derived from several bases. Different selections of bases may give different reasons (deductions) and this difference is crucial under the foundational approach, but irrelevant under the coherence approach.

Another important consideration is the problem of *iterated revisions*. All the algorithms described so far are concerned with just one update. There are cases when this is not entirely correct. It can be shown that there are sequences of revisions which give counter-intuitive results if we process each one individually. A solution to this problem is to process the sequence of revisions as a whole ([4, 15]). The main problem regarding the one-update algorithms is the fact that the belief base is not properly selected after each update, because the algorithms are only concerned with the result of the update and not with how this result occurred. This can cause the loss of valuable information as far as future updates are concerned (see section 9 for an example). The proposed solution is based on the principle that two KBs should be considered equivalent if, in addition to the logical equivalence of the bases themselves, they will give equivalent results to all possible updates as well. This is the basic principle governing the algorithms of iterated belief revision ([15]). It is also pointed out in [4], where the difference between *belief sets* (knowledge only) and *epistemic states* (knowledge including information on how to revise it) is discussed.

An additional difficulty of the problem of belief revision is the fact that the result of an update may also depend on the *source* of the data. Let us suppose that there are two lamps, A and B, in a room and we know that exactly one of them is on. Our knowledge can be represented by the proposition: $(\alpha\wedge\neg b)\vee(\neg\alpha\wedge b)$. If we make the observation that lamp A is on, we should update our KB with the proposition α and the intuitively correct result for the update is the proposition $\alpha\wedge\neg b$, as we know now that B is off. On the other hand, if a robot is sent into the room in order to turn lamp A on, then we would again have the update α. The proper intuitive result of the update is the proposition α in this case, as we know nothing about the state of lamp B; it could have been on or off before sending the robot in the room (and stayed so). This example shows that even identical (not just equivalent) databases can give different intuitively correct results with identical updates!

In order to overcome the problem, two different types of updates have been defined in [13], namely *revision* and *update*. Revision is used when new information about a static world is obtained. This is the first case of our example where the observation did not change the state of A and B. The AGM postulates and the algorithms presented deal with revisions. A revision is performed when the source of the data is an observation regarding the world. Update is used when the world dynamically changes and we have to record that change. In the second case of our example, the robot changed the state of lamp A, and consequently the state of the world being modeled. Therefore, the result of the update must be different. An update is performed when the reason of change is an action, instead of an observation. An excellent study on the problem may be found in [13], where a new set of postulates, adequate for update, is presented.

Update and revision are used to add knowledge to a KB. Essentially, new data enhances our knowledge about the domain of interest. In some cases though, we could have information that removes knowledge from the KB. This could happen, for example, when we find out that a measuring instrument produces wrong output. This should force us to remove knowledge previously added by that source, as unreliable. This operation is called *contraction* ([10, 13, 16]), and it is dual to revision. It has been argued ([10, 13, 16]) that it is intuitively simpler to deal with contraction instead of revision. Similarly, we could define the operation of *erasure* which is dual to update. For the class of revision schemes that satisfy the AGM postulates, it has been proven that revision and contraction can be defined in terms of each other. Similar results apply for the update/erasure schemes that satisfy the postulates for update defined in [13]. As we will see, our approach will deal with all such kinds of updates using the same operation, so throughout this paper the terms update and revision will be used interchangeably; their meaning should be understandable by the context.

4 Driving Considerations

Before describing how our approach deals with the above problems, we will make a short description of the logical framework used. As already stated, we restrict ourselves in KBs whose knowledge can be expressed using a finite number of propositional expressions. The term "knowledge" or "belief" will refer to any propositional formula. The underlying propositional language will contain the usual operators ($\neg,\wedge,\vee,\rightarrow,\leftrightarrow$), parentheses, the logical constants (T,F) and a finite number of atoms ($\alpha_1,\alpha_2,\ldots,\alpha_n$). The language will be denoted by L and the set of all formulas resulting from L will be denoted by L^*. Interpretations are assignments of logical values (T,F) to the atoms of the language. In most cases, it will be more convenient to use the constant 0 instead of F and the constant 1 instead of T. Under this notion, an interpretation is an ordered finite sequence of size n (equal to the number of atoms in the propositional language), consisting of elements from the set {0,1}.

Regarding the representation of the base, we adopt Nebel's proposition ([17]) where the contents of the KB are the individual updates, each of them expressing an observation, experiment, rule etc regarding a domain of interest. As in any belief base, the knowledge of the KB consists of these updates as well as their logical consequences, but the selection of the belief base just described implies that this approach follows the foundational paradigm. The propositions in the KB are the foundational beliefs and the rest are implied directly or indirectly by them. We believe that the foundational approach is more compatible with common sense, and that, in principle, knowledge is actually derived from the observations we make about the world. Therefore, the storage of the observations themselves is the best way to describe our knowledge ([17]). The derived facts may be used for faster deduction and query answering, but they are of no value as far as the actual knowledge is concerned. This deals with the problem of iterated revisions as well, because each observation is actually one update and is explicitly stored, allowing us to process the whole sequence, if this proves necessary.

In practice, however, some updates may contradict previous ones; in such a case, keeping all the updates will result in an inconsistent KB. Whenever a new piece of knowledge is introduced to the KB, we should check whether it contradicts any of the previous updates or their consequences. In such a case, under Nebel's approach, at least one of the previous updates (knowledge) or the new revision itself must be modified or discarded altogether in order for the KB to remain consistent. Under our scheme, a different approach is taken: the contradictions are resolved at query time, but are otherwise kept in the KB. Moreover, instead of storing one structure representing the knowledge of each update individually, we keep one single structure for the whole KB in which all the previous updates are contained in an encoded fashion. This is helpful for a homogeneous approach of all updates, regardless of the number of updates already given as input to our KB.

Given a formula expressing some knowledge regarding the world, the set of interpretations that it satisfies is exactly the set of possible worlds that it describes. At any given point in time, the real world can be represented by a unique assignment of truth-values to the atoms describing it (interpretation). This interpretation may change through time and our goal is to find it. Usually, an update will only give partial knowledge regarding the world, by specifying the truth-values of some, but not all, atoms in the real world. Of course, this update may contain disjunctions, which means that each disjunction describes a different possible world. Moreover, we cannot be sure in advance that the update is correct. We may only assume its correctness and if our assumption is wrong, this may lead us away from the real world.

The above remark leads us to another consideration, regarding the approval or rejection of each new update. Whenever a contradictory update is performed, the contradiction may be resolved in two ways; either we consider the update correct and try to change the KB to accommodate the update, or we consider the KB correct and reject or try to change the update in order to be accommodated in the KB. Under most updating schemes, updates are considered more reliable than the old data, as expressed by Dalal's Principle of Primacy of New Information for example. This is generally a good practice, as updates are usually regarded as the latest information about the world and can be assumed correct. However, this may not be true in several cases, as the new data may come from a noisy or otherwise unreliable source. In order to overcome the problem, we will assign a non-negative real number to each belief, which will represent its reliability. We will call this number the *Reliability Factor* of the belief and we will use the abbreviation *RF*. When a piece of contradictory information is used to revise our knowledge, at least some of the existing data (or the update itself) must be rejected. The use of the RF implies that the piece(s) of data to be rejected should be the one(s) with the lowest RF. Notice however that there are cases where small changes in existing pieces of data (or the update) are enough to accommodate the update without introducing any inconsistencies. No data rejection is necessary in this case. The selection of the data to reject or change is not an easy one, because there may be several ways to accommodate an update. Finding all possible ways to do so requires evaluating all possible subsets of our belief base and comparing all the ways of removal of the inconsistency in terms of "knowledge loss", which is some function of the RFs involved. This is a computationally expensive operation.

5 Matrices and Knowledge Representation

Considerations such as the above, led us to the search of an algorithm that would give us the contradictions, the possible ways of removal and the cost of removal per case at the same time, where the cost is measured under an intuitively proper metric of "lost knowledge". A knowledge representation scheme that would enable the above is based on the *Table Transformation*, a function which transforms an expression of any finite propositional language into a matrix of complex numbers.

As mentioned in the previous section, we need to assign a level of confidence in each belief, which we will call Reliability Factor (RF). The simplest possible belief is an atom, so we will first describe how this assignment is made on atoms. By applying the RF in all atoms of a propositional expression, we implicitly apply it to the expression itself.

In any propositional expression, an atom of the language, say α, can occur either as a positive atom (α) or as a negative atom (¬α); alternatively, it may not occur at all. Let us take one occurrence of atom α in some expression p. Its occurrence indicates belief in the atom itself (α) or its negation (¬α), or, equivalently, our confidence that, in the real world, α is true or false respectively. RF indicates the level of this belief. It is a real number, attached to all atoms' truth and all atoms' falsehood and may be positive, negative or zero. If positive, it indicates the level of confidence in the truth (or falsehood) of the atom; if negative, it indicates distrust and the level of this distrust; if zero, it indicates lack of knowledge. It must be emphasized that distrust in one atom (or proposition, in general) is not generally equivalent to believing the atom's (or proposition's) negation. Therefore, a negative RF attached to the truth of an atom is not equivalent to a positive RF attached to its falsehood (and vice-versa).

This analysis implies that all knowledge related to one occurrence of a given atom in a given proposition can be expressed using a pair of real numbers (x,y), representing the RFs of the atom's truth and falsehood respectively. For an atom appearing as a positive atom (α), number y should be 0 (no knowledge regarding the atom's falsehood), whereas for a negative atom (¬α), number x should be 0 (no knowledge regarding the atom's truth). By using the well known isomorphism of the set $\mathbb{R}^2$ to the set of complex numbers $\mathbb{C}$, we can equivalently express an atom's knowledge using the complex number $z=x+yi$. This number will be referred to as the atom's knowledge, or the number attached to the atom, and we will use atoms and complex numbers interchangeably.

The complex number so defined can express a variety of information for each atom, depending on the real and imaginary part of $z=x+yi\in\mathbb{C}$. The sign of number x indicates whether we believe (x>0) or don't believe (x<0) that the respective atom is true in the real world. The absolute value of x indicates the level of trust or distrust in the above information. Number y gives us similar information about the falsehood of the atom. By setting x=0 or y=0, we indicate lack of knowledge regarding the atom's truth or falsehood respectively. By combining different values (sign and absolute value) of x and y we can express different types of knowledge regarding the atom. By setting both to zero (x=y=0), we believe in neither the atom's affirmation or negation, which implies complete lack of knowledge. In such a case, we must accept both the atom's affirmation or negation (if asked), as there is no reason to disallow any of the two. By setting only one of them equal to 0, we know nothing about the truth (if x=0) or falsehood (if y=0) of the atom, so our knowledge must rely on the non-zero element. On the other hand, if both x>0 and y>0, then we believe in both the affirmation and the negation of the atom. This implies a contradiction; the intensity of the contradiction can be estimated by the values of x and y. Moreover, the prevailing belief can be determined, by a simple comparison between x and y. If both x and y are negative the same comments apply, with the only difference being that we now disbelieve both the affirmation and the negation of the atom. Once again, the intensity of the contradiction and the prevailing belief can be easily determined. Finally, if $x>0$ and $y<0$, we believe in the truth of the atom and don't believe in its falsehood. This information actually enhances our initial impression that the atom must be true in the real world. The same comments apply if $x<0$ and $y>0$, but for the atom's falsehood. All the above apply in different degrees depending on the absolute values of x and y. All these imply a significant ability regarding our power of expressiveness and a great improvement over the classical method of representation using propositional expressions.

In order to expand these thoughts to include arbitrary propositions we need to integrate knowledge from different atoms. A simple way to do that is to set out the knowledge regarding the atoms in a 2-dimensional matrix of complex numbers. We will call this procedure *Table Transformation* (*TT*). This function closely relates propositional logic with 2-dimensional matrices and it has some very interesting properties.

The transformation can be applied to any proposition; however, for technical reasons it is better to use the proposition's disjunctive normal form (DNF). Any well-formed formula in propositional logic has an equivalent DNF expression, so this is not a restriction. We will expand the transformation to include arbitrary propositions at a later point. The transformation returns a matrix of complex numbers. In [8] we used ordered pairs of non-negative numbers instead of complex numbers, but the idea is pretty much the same, as ordered pairs can be assigned to complex numbers and vice-versa. Moreover, negative numbers are now allowed, representing our "stiffness" in the acceptance of an atom and/or its negation, so this is a direct expansion of the method appeared at [8].

In short, each atom of the language is assigned to one column of the matrix, and there are as many lines in the matrix as the number of disjuncts in the DNF of the propositional expression. In the figure below, we show the transformation of the expression $P=(\alpha\wedge b\wedge\neg c)\vee(\alpha\wedge\neg b\wedge d)\vee(c\wedge e)$ into its respective matrix. We suppose that the language consists of 5 atoms, namely α, b, c, d and e, so the matrix has 5 columns. The expression is already in DNF and it has 3 disjuncts, so the number of lines in the matrix is 3. As far as the contents of the matrix are concerned (called *elements* hereof), the procedure is the following: an element has the value of 1 if and only if the respective atom in the respective disjunct appears as a positive atom; if negative the

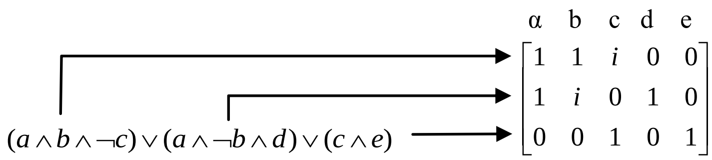

Fig. 1. Example of a table transformation

value is i; if the atom does not appear at all, then the value of the element is 0. The application of these rules on the expression P results in the matrix below:

The above matrix only contains information regarding the affirmation or negation of each atom per clause of the DNF of P. We could have assigned weights in each element, if such information was available, thus enhancing the knowledge carried by the resulting matrix. In such a case, we could have for example the matrix:

Each element in the matrix represents the type of membership of one atom in one disjunct of the proposition. As already noted, a complex number may encapsulate a whole variety of information, including the type of belief (affirmation, negation, lack of knowledge, contradiction etc) and the level of this belief. Therefore, to transform the matrix P back to a logical expression we simply “decode” the information contained in each element of the matrix. This inverse procedure is called *Inverse Table Transformation* and denoted by *TTI*. Due to the inherent inability of propositional logic to express disbelief, elements implying disbelief in a certain fact

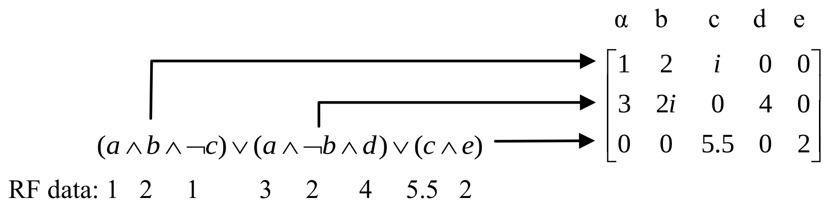

Fig. 2. Example of a table transformation with weights (RFs)

(negative RF) are mapped into the negation of the fact, even though their semantics is not the same, as already noted. Furthermore, any RF data cannot be expressed in logic, so it is ignored during the transformation.

The use of the TT function allows us to relate any propositional expression p to a matrix P. If information on the atoms’ reliability (RF) is available, this additional knowledge is provided by the matrix representation in a very compact form. In any case, TT causes no loss of information, because the TTI function can be applied at any time to transform the matrix back to the proposition p. These properties make matrices very attractive for knowledge representation. Each line of P represents a different clause in p, thus a different possible state of the world represented by p. For this reason, we will usually refer to lines as *possible worlds*.

The above simple definition of the TT captures the intuition that will be used throughout this paper to describe our belief revision scheme. In fact, this is not the only way to define TT, but a special easy case. In our attempt to expand this representation to propositions not in DNF and to include complex numbers with negative real or imaginary part, we will encounter some problems which will force us to change the definition somewhat. However, these issues are irrelevant as far as the knowledge representation and belief revision problems are concerned, so we postpone their discussion until a later point.

6 Matrices and Belief Revision

Our initial goal on defining the table transformation was to use this transformation in belief revision. In order to solve this problem, we will assume that both the knowledge and the update are represented by matrices. In other words, we assume that we already have a general method of transforming the abstract notion of "real world knowledge" to a specific weighted matrix that represents it. Intuitively, this is easy to do, using the method described in the previous section.

For the moment, we will additionally assume that both the base K and the update M have only one line, so they both represent only one possible world. In this special case, the update will be defined as the *addition* of the two matrices, because the inclusion of the update M in our knowledge increases our trust in all the information that M carries, effectively increasing our reliance in each atom and/or its negation. This adjustment may or may not be enough to force us to change our beliefs. In general, when we believe something with reliability x and a new piece of knowledge confirms this belief, coming from a source with reliability y, then the logical result of the revision should be the same belief with reliability x+y. This is the notion behind defining the revision as the addition of the two matrices. One can verify that the operation of addition is intuitively correct even in cases where x and/or y are negative numbers, or when they refer to contradicting data (one implying α and the other ¬α for some atom α). Let us see one example:

$$K = \begin{bmatrix} i & 3 & 0 \end{bmatrix}, M = \begin{bmatrix} 3 & 2 & 1 \end{bmatrix}$$

Using the TTI function we can easily verify that the matrices represent the expressions K=¬α∧b (base) and M=α∧b∧c (update). The matrices additionally show the reliability per atom in each proposition. In this case, the negation of atom α (¬α) is believed with an RF of 1 in K, but M should force us to abandon this belief as atom α is believed with an RF of 3 in M. In atom b, there is no contradiction between K and M; however this revision should increase our confidence in the truth of b. Finally, in atom c, we know now that c is true, with a reliance of 1; we had no knowledge regarding c before. The resulting matrix is K′ below where the symbol "•" stands for the operation of update:

$$K' = K \bullet M = \begin{bmatrix} i & 3 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 3+i & 5 & 1 \end{bmatrix}$$

The proposition related (via the TTI function) to the resulting matrix is: F∧b∧c≅F, because the first element (3+i) implies a contradiction. Note that the resulting matrix K′ is a contradictory matrix (and the related proposition is a contradiction too). This is not generally acceptable, as a contradictory KB actually contains no information. The result should have been α∧b∧c, as showed by the previous analysis. We will deal with this problem later (which is in fact not a problem at all!).

In the general case where one (or both) of the matrices contain more than one line, each line represents one possible world. In order to be sure that the correct world will be represented by a line in the resulting matrix, we must add (revise) each line of K with each line of M, creating one line per pair in the resulting matrix K′. Let us see

$$K = \begin{bmatrix} 1 & 2 \\ 0 & 5i \end{bmatrix}, M = \begin{bmatrix} i & 3 \\ 2i & 3i \end{bmatrix}, K \bullet M = \begin{bmatrix} 1+i & 5 \\ 1+2i & 2+3i \\ i & 3+5i \\ 2i & 8i \end{bmatrix}$$

one example:

7 Queries and Contradictions

Any given matrix has no intuitive meaning for a user, because finding the knowledge contained in a large KB matrix is not a trivial operation. More importantly, the user is normally not allowed nor interested to see the whole KB, but wishes to execute queries for finding specific information. To reply to a user query, we must transform the KB matrix back into a proposition expressing the knowledge contained in it. TTI is one function that transforms a matrix into a logical expression. However, as we saw, there are cases when the result of a revision is a matrix corresponding via TTI to the logical constant F; equivalently, the KB related to the matrix via TTI is inconsistent. This property is undesirable, so the TTI function is inadequate for our purposes. Instead of trying to find another function, we will perform some pre-processing on the KB matrix before applying TTI. This pre-processing must be defined in such a way as to remove the parts of the matrix that cause the contradictions and that contain data not properly describing the real world.

In our search for an adequate such function, some comments are in order. First, we believe that contradictory lines and elements may contain useful information that cannot be discarded without some evaluation. One may argue that contradictory lines represent contradictory possible worlds, so they contain false information and they could as well be deleted. This is the policy followed by the TTI function. However, this is not entirely true. A single faulty revision may create a contradiction in an otherwise correct possible world (line), so we must be careful before discarding a contradictory line as non-true. On the other hand, even if a line contains no contradictions at all, this is by no means a guarantee that it is entirely true. It could be too far from the real world; we just don't know it yet. Therefore, our policy is to keep the matrix as-is and to let inconsistency stay in the KB, even if some (or all) lines are contradictory. Secondly, the transformation we will define cannot be based on all the lines, because some of them might indeed be wrong (away from the real world).

These two points show that a kind of evaluation upon the possible worlds must be performed to find the "most possible" ones. The criterion of correctness of a possible world (line) should be a function that maps each line to a number representing our estimation on the line's correctness or proximity to the real world. Only the most correct ones will be selected. Because of the fact that some (or all!) of the selected lines may contain contradictions, those contradictions must be subsequently removed from the selected submatrix.

One possible way to materialize the above requirements is described in [8, 9]. In order to estimate the proximity of a possible world to the real world, a function named *Line Reliability* (*RL*) is used, which depends on the *Element Reliability* (*RE*) function. RE is a function that calculates our reliance in the truth of the information that one element carries. This quantity has nothing to do with the truth or falsity of the respective atom; all it expresses is our estimation on the proximity of the element's information to the real world. Similarly, RL is a function that uses the estimations of the RE function in order to estimate the proximity of the line's (possible world's) information to the real world.

Given the estimation provided by the RL function, we should select the lines that are sufficiently close to the real world (according to our estimations, as usual). This selection is made by a function named *Line Selection* function (*LS*), which returns a set of indices, corresponding to the selected lines of the original matrix. This set is used by the *Submatrix Selection* function (*MS*) to return a submatrix of the original matrix, according to our selection.

In order to remove the contradictions possibly appearing in the selected lines the *Matrix Normalization* function (*MN*) is used. The contradictions can be removed in a relatively simple manner; a contradictory element denotes belief (or disbelief) to both an atom and its negation. By comparing the absolute value of the real and imaginary part of the contradictory element we can easily determine whether to accept the affirmation or the negation of the atom. In effect, we select the information with the largest RF value, so we decide whether to believe in the truth or falsity of the atom (instead of both or neither, which is the source of the contradiction).

This procedure ensures that there will be no contradictions in the matrix that resulted by the selection implied by the LS function. This final matrix expresses the (consistent) KB that will be used in queries. It contains the most reliable possible worlds of our KB, which have been normalized in order to extract the information contained in the contradictory elements. At this point, we can eventually apply the TTI function to get the logical expression related to this matrix. This expression corresponds to the knowledge stored in the KB at this point.

The whole process (function) of transforming a matrix to a proposition for the needs of queries will be denoted by *QT* (*Query Transformation*). It is clear by the analysis above that QT is in fact a composite function. Notice that the QT operation does not actually change the matrix; it temporarily transforms it to a logical expression for the needs of queries. The next revision will be executed upon the original matrix and the whole process of query transformation should be repeated after the revision to calculate the new related proposition.

The QT function is composed of six subsequent steps, each of which has a specific role, as described above. Out of these steps, the first three (element reliability, line reliability and line selection functions) are used to extract the "most reliable" information of the matrix. This notion of reliability is not something that can be intersubjectively defined, because it depends on the application. For this reason, these three functions have the role of a parameter regarding the QT function. Arguments for this fact, as well as its importance, will be set forth in section 9. The final three (submatrix selection, matrix normalization and inverse table transformation functions), are clearly defined and express the procedure of extracting information out of (possibly contradictory) elements.

8 Formal Framework

With the introduction of the QT function, we completed the description of the framework we propose for representing and updating knowledge. Summarizing our method, we can say that it consists of the following three parts:

1. *Knowledge Representation*: express the knowledge represented by the logical propositions of the input into matrices and encode the reliability information, if available, into the matrix elements. This procedure transforms our knowledge into the more expressive form of the matrix representation.

2. *Knowledge Revision*: whenever any new data is introduced, apply the belief revision algorithm to accommodate the new knowledge (revision).
3. *Knowledge Query*: apply the QT function upon the matrix representing the KB to extract the KB's information, in order to reply to user queries.

In this section we provide a formal framework describing the procedures informally described in previous sections. Proofs are omitted due to lack of space, but they can be found in [9]. At first, some notational conventions should be introduced. As usual, we denote by $\mathbb{R}^{(+)}$ the set of non-negative real numbers. Analogously, we denote by $\mathbb{C}^{(+)}$ the set: $\mathbb{C}^{(+)}=\{x+yi\in\mathbb{C}|x,y\in\mathbb{R}^{(+)}\}$, i.e. the complex numbers whose real and imaginary part are both non-negative. For matrices, we define $\mathbb{C}^{m\times n}$ to be the set of matrices with m lines and n columns whose elements are complex numbers, and $\mathbb{C}^{*\times n}$ the set: $\mathbb{C}^{*\times n}=\{A\in\mathbb{C}^{m\times n}$ for some $m\in\mathbb{N}^*\}$, i.e. the union of $\mathbb{C}^{m\times n}$ for all $m\in\mathbb{N}^*$. In other words, $\mathbb{C}^{*\times n}$ is the set of matrices with n columns, whose elements are complex numbers. Analogously, we define the sets $\mathbb{C}^{(+)m\times n}$ and $\mathbb{C}^{(+)*\times n}$, as well as the sets $\mathbb{R}^{m\times n}$, $\mathbb{R}^{*\times n}$, $\mathbb{R}^{(+)m\times n}$ and $\mathbb{R}^{(+)*\times n}$, for matrices of real numbers.

We will use the usual notation for addition and multiplication of matrices, as well as for the multiplication of a number with a matrix. Moreover, we define the operation of *juxtaposition* as follows:

Definition 1 Let A, $B\in\mathbb{C}^{*\times n}$. The *juxtaposition* of A and B, denoted by A|B, is the matrix that results by placing the lines of A followed by the lines of B, i.e.:

$$A\,|\,B=\begin{bmatrix} A \\ --- \\ B \end{bmatrix}.$$

We will also define a partitioning on $\mathbb{C}$:

Definition 2 Let $z\in\mathbb{C}$ and x=Re(z), y=Im(z) its real and imaginary part, respectively. Then we define:

- If $x\geq0$, $y\leq0$, then z is called *positive*.
- If $x\leq0$, $y\geq0$, then z is called *negative*.
- If $x\cdot y>0$, then z is called *contradictory* (when $x>0$ and $y>0$ or $x<0$ and $y<0$).
- If z is both positive and negative, then z is called *zero* (when x=y=0).

We will denote by $\mathbb{C}_0$ the set of zero complex numbers, $\mathbb{C}_+$ the set of positive complex numbers, $\mathbb{C}_-$ the set of negative complex numbers and $\mathbb{C}_*$ the set of contradictory complex numbers.

It follows that $\mathbb{C}_0=\{0\}$, $\mathbb{C}_+\cap\mathbb{C}_-=\mathbb{C}_0$, $\mathbb{C}_+\cap\mathbb{C}_*=\varnothing$, $\mathbb{C}_-\cap\mathbb{C}_*=\varnothing$, $\mathbb{C}_-\cup\mathbb{C}_+\cup\mathbb{C}_*=\mathbb{C}$. It is also interesting to note the intuitive meaning of such sets:

- Numbers belonging to the set $\mathbb{C}_+$ denote trust in the truth of the atom ($Re(z)\geq0$) and distrust in the falsehood of the atom ($Im(z)\leq0$). This implies that our current knowledge indicates that the atom is true (positive literal).

- In a similar fashion, numbers belonging to the set $\mathbb{C}_-$ denote distrust in the truth of the atom (Re(z)≤0) and trust in the falsehood of the atom (Im(z)≥0). This implies that our current knowledge indicates that the atom is false (negative literal).
- Numbers belonging to the set $\mathbb{C}_*$ denote trust in both the truth and falsehood of the atom (Re(z)>0, Im(z)>0) or distrust in both (Re(z)<0, Im(z)<0). Both these cases indicate a contradiction, because an atom can be either true or false. It cannot be both (Re(z)>0, Im(z)>0) and it cannot be neither (Re(z)<0, Im(z)<0), so both cases are invalid (contradictory).
- Finally, number 0 – belonging to the set $\mathbb{C}_0$ – denotes lack of knowledge in both the atom and its negation; we have no reason to accept or discard the truth or falsehood of the atom, so we have to accept both. Besides, $\mathbb{C}_+ \cap \mathbb{C}_- = \mathbb{C}_0$, therefore 0 indicates the fact that the atom can be either true or false.

Definition 3 We define the following matrices:

- Let $A \in \mathbb{C}^{1\times n}$ be a matrix of the form [0 … 0 1 0 … 0], where the element 1 is in the k-th column. Matrix A will be called a *k-atom* or a *positive k-atom* and we will use the notation A_k.
- Let $A = z \cdot A_k \in \mathbb{C}^{1\times n}$, for some $z \in \mathbb{C}$. Matrix A will be called a *generalized k-atom*, and we will use the notation $A_k(z)$. In the special case where z=i, matrix A will be called an *inverse k-atom* or a *negative k-atom* and we will use the notation A'_k or $A_k(i)$.
- The matrix A=[0 0 … 0]$\in \mathbb{C}^{1\times n}$ is called the *n-true matrix* and we will use the notation T_n.
- The matrix A=[1+i 1+i … 1+i]$\in \mathbb{C}^{1\times n}$ is called the *n-false matrix* and we will use the notation F_n.

The informal analysis made in previous sections shows that our goal is to assign the generalized k-atoms to the atoms of our propositional language. T_n will be assigned to the constant T and F_n will be assigned to the constant F. Moreover, notice that the matrices T_n, F_n and $A_k(z)$, for $z \in \mathbb{C}^{(+)}$ also belong to set $\mathbb{C}^{(+)1\times n}$.

The following proposition is immediate by selecting z_{kj} to be the elements of A:

Proposition 1 For any $m \in \mathbb{N}^*$ and any matrix $A \in \mathbb{C}^{m\times n}$, there exist unique $z_{kj} \in \mathbb{C}$, k=1,2,…,m, j=1,2,…,n, such that: $A = (\sum_{j=1}^{n} A_j(z_{1j})) \mid (\sum_{j=1}^{n} A_j(z_{2j})) \mid \ldots \mid (\sum_{j=1}^{n} A_j(z_{mj}))$.

The above form (Proposition 1) of a matrix A will be called the *extended normal form* of A, and denoted by ENF(A).

Definition 4 We define the truth constants $F=0 \in \mathbb{C}$ and $T=1 \in \mathbb{C}$. Any ordered n-sized sequence of numbers in the set {0,1}={F,T} is called an *interpretation* of space $\mathbb{C}^{*\times n}$. The set of all interpretations of space $\mathbb{C}^{*\times n}$ will be denoted by I(n).

Notice that matrix interpretations can be directly mapped to logical interpretations and vice-versa, as they both are ordered finite sequences consisting of elements from the set {0,1}={F,T}.

Definition 5 Let $I=(\alpha_1,\alpha_2,\ldots,\alpha_n)\in I(n)$ an interpretation and $A\in\mathbb{C}^{1\times n}$ a matrix such that: $A=\sum_{j=1}^{n} A_j(a_j+(1-a_j)\cdot i)$. The matrix A is called an *interpretation matrix* of space $\mathbb{C}^{*\times n}$.

Notice that there is a direct 1–1 and onto relationship between interpretations and interpretation matrices. We will use the term interpretation for the interpretation matrices whenever there is no risk of confusion. Moreover, the number $z=\alpha_j+(1-\alpha_j)\cdot i$ can be either z=1 (for $\alpha_j=1$) or z=i (for $\alpha_j=0$). Therefore, an interpretation matrix is a matrix of the set $\mathbb{C}^{1\times n}$, whose elements are from the set {1,i}.

Definition 6 Let $I=(\alpha_1,\alpha_2,\ldots,\alpha_n)\in I(n)$ an interpretation and $A\in\mathbb{C}^{1\times n}$. We say that A is *satisfied by* I iff the following condition holds:

$\exists z_1,z_2,\ldots,z_n\in C^{(+)}: A=\sum_{j=1}^{n}[A_j((2\cdot a_j-1)\cdot\overline{z_j})]$, where $\overline{z_j}$ is the conjugate complex of $z_j\in\mathbb{C}^{(+)}$.

In general, if $A\in\mathbb{C}^{m\times n}$ such that $A=A^{(1)}|A^{(2)}|\ldots|A^{(m)}$, $A^{(j)}\in\mathbb{C}^{1\times n}$, j=1,2,…,m, then we say that A is *satisfied by* I iff there exists $j\in\{1,2,\ldots,m\}$ such that $A^{(j)}$ is satisfied by I.
The set of interpretations that satisfies A will be denoted by mod(A) and called the *set of models* of A.

Notice that the quantity $(2\cdot\alpha_j-1)$ can only take the values ±1. Therefore, the quantity $x_j=(2\cdot\alpha_j-1)\cdot\overline{z_j}$ has the property that either $Re(x_j)\geq 0$ and $Im(x_j)\leq 0$ (for $\alpha_j=1$) or $Re(x_j)\leq 0$ and $Im(x_j)\geq 0$ (for $\alpha_j=0$). This concurs with our previous definition on positive and negative complex numbers. Numbers z_j have been introduced to support the notion of RF; interpretation matrices have no "intensity" information, because their elements are from the set {1,i}, whereas, in general, a matrix may be weighted, carrying RF information. We can prove two useful, constructive methods to calculate the models of any given matrix, which are given by the propositions:

Proposition 2 Let $A\in\mathbb{C}^{m\times n}$, such that:

$$A=\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix}.$$

Then $\mathrm{mod}(A)=\bigcup_{j=1}^{m}(I_{j1}\times I_{j2}\times\ldots\times I_{jn})$, where for any $j\in\{1,2,\ldots,m\}$, $k\in\{1,2,\ldots,n\}$:

- $I_{jk}=\{0\}$ iff $w_{jk}\in\mathbb{C}_{-}\backslash\mathbb{C}_{+}=\mathbb{C}_{*}\backslash\mathbb{C}_{0}$
- $I_{jk}=\{1\}$ iff $w_{jk}\in\mathbb{C}_{+}\backslash\mathbb{C}_{-}=\mathbb{C}_{-}\backslash\mathbb{C}_{0}$
- $I_{jk}=\{0,1\}$ iff $w_{jk}\in\mathbb{C}_{0}$
- $I_{jk}=\varnothing$ iff $w_{jk}\in\mathbb{C}_{*}$

<u>*Proposition 3*</u> Let $m\in\mathbb{N}^{*}$ and $A\in\mathbb{C}^{m\times n}$. Moreover, let: $A=(\sum_{j=1}^{n}A_j(z_{1j}))\,|\,(\sum_{j=1}^{n}A_j(z_{2j}))\,|\ldots|\,(\sum_{j=1}^{n}A_j(z_{mj}))$, be its ENF. Then: $\mathrm{mod}(A)=\bigcup_{k=1}^{m}\bigcap_{j=1}^{n}\mathrm{mod}(A_j(z_{kj}))$.

Proposition 3 is interesting because it indicates a close connection between the operations of juxtaposition and addition between matrices to the operations of union and intersection between models, respectively. Juxtaposition is closely related to the union of models:

<u>*Proposition 4*</u> Let $A, B\in\mathbb{C}^{*\times n}$. Then $\mathrm{mod}(A|B)=\mathrm{mod}(A)\cup\mathrm{mod}(B)$.

Unfortunately, the connection between addition and intersection is not so straightforward. We will explore their relation at a later point (see Definition 10 and Proposition 6). In order to define the TT at the general case, we need some operations on matrices that will emulate the usual operations of disjunction, conjunction and negation in classical propositional logic:

<u>*Definition 7*</u> We define three classes of functions in $\mathbb{C}^{*\times n}$, denoted by $\mathcal{F}_{\vee}$, $\mathcal{F}_{\wedge}$ and $\mathcal{F}_{\neg}$, called the *class of disjunction functions*, the *class of conjunction functions* and the *class of negation functions* respectively.

- A function $f_{\vee}:\mathbb{C}^{*\times n}\times\mathbb{C}^{*\times n}\to\mathbb{C}^{*\times n}$ is said to belong in the class $\mathcal{F}_{\vee}$ iff for any $A,B\in\mathbb{C}^{*\times n}$, $\mathrm{mod}(f_{\vee}(A,B))=\mathrm{mod}(A)\cup\mathrm{mod}(B)$.
  We will also use the notation $A\vee B$ for the result of $f_{\vee}(A,B)$.
- A function $f_{\wedge}:\mathbb{C}^{*\times n}\times\mathbb{C}^{*\times n}\to\mathbb{C}^{*\times n}$ is said to belong in the class $\mathcal{F}_{\wedge}$ iff for any $A,B\in\mathbb{C}^{*\times n}$, $\mathrm{mod}(f_{\wedge}(A,B))=\mathrm{mod}(A)\cap\mathrm{mod}(B)$.
  We will also use the notation $A\wedge B$ for the result of $f_{\wedge}(A,B)$.
- A function $f_{\neg}:\mathbb{C}^{*\times n}\to\mathbb{C}^{*\times n}$ is said to belong in the class $\mathcal{F}_{\neg}$ iff for any $A\in\mathbb{C}^{*\times n}$, $\mathrm{mod}(f_{\neg}(A))=I(n)\backslash\mathrm{mod}(A)$.
  We will also use the notation $\neg A$ for the result of $f_{\neg}(A)$.

<u>*Definition 8*</u> The space $\mathbb{C}^{*\times n}$, equipped with three functions denoted by $\vee$, $\wedge$, $\neg$ belonging to the classes $\mathcal{F}_{\vee}$, $\mathcal{F}_{\wedge}$ and $\mathcal{F}_{\neg}$ respectively ($\vee\in\mathcal{F}_{\vee}$, $\wedge\in\mathcal{F}_{\wedge}$, $\neg\in\mathcal{F}_{\neg}$), is called

a *logically complete matrix space of dimension n*. We will denote such spaces with the quadruple: $(\mathbb{C}^{*\times n},\vee,\wedge,\neg)$.

Notice that we do not set any restrictions on the selection of the three operations, as long as they satisfy the conditions of the definition of the three classes of functions. Obviously, there is more than one possible selection for these operators. Different selections for the operators $\vee$, $\wedge$, $\neg$ result in different logically complete spaces. The definition of additional operators, such as the implication ($\rightarrow$) can be done using a similar discipline; however in this work we assume that such operations are not needed and will not be defined. The above definitions and propositions are enough to support the definition of the *Table Transformation function*:

Definition 9 Let $(\mathbb{C}^{*\times n},\vee,\wedge,\neg)$ be a logically complete matrix space of dimension n and $L=\{T,F,(,),\vee,\wedge,\neg,\alpha_1,\alpha_2,\ldots,\alpha_n\}$ a finite propositional language. As usual, we denote by α_j the atoms of the propositional language and by A_j the atoms of $\mathbb{C}^{*\times n}$.

We define the *Table Transformation function*, $TT:L^*\rightarrow\mathbb{C}^{*\times n}$, recursively as follows:

- $TT(T)=T_n$
- $TT(F)=F_n$
- For any $j\in\{1,2,\ldots,n\}$: $TT(\alpha_j)=A_j$
- $TT(p\vee q)=TT(p)\vee TT(q)$, for any $p,q\in L^*$
- $TT(p\wedge q)=TT(p)\wedge TT(q)$, for any $p,q\in L^*$
- $TT(\neg p)=\neg TT(p)$, for any $p\in L^*$

Similarly, we define the *Inverse Table Transformation function*, $TTI:\mathbb{C}^{*\times n}\rightarrow L^*$, recursively, as follows:

For any $j\in\{1,2,\ldots,n\}$, $z\in\mathbb{C}$, we define:

- $TTI(A_j(z))=\alpha_j$, iff $z\in\mathbb{C}_+\backslash\mathbb{C}_-=\mathbb{C}_+\backslash\mathbb{C}_0$
- $TTI(A_j(z))=\neg\alpha_j$, iff $z\in\mathbb{C}_-\backslash\mathbb{C}_+=\mathbb{C}_-\backslash\mathbb{C}_0$
- $TTI(A_j(z))=T$, iff $z\in\mathbb{C}_0$
- $TTI(A_j(z))=F$, iff $z\in\mathbb{C}_*$

In general, for any $m\in\mathbb{N}^*$ and any matrix $A\in\mathbb{C}^{m\times n}$, whose ENF is $A=(\sum_{j=1}^{n}A_j(z_{1j}))\,|\,(\sum_{j=1}^{n}A_j(z_{2j}))\,|\ldots|\,(\sum_{j=1}^{n}A_j(z_{mj}))$, we define $TTI(A)=\bigvee_{k=1}^{m}(\bigwedge_{j=1}^{n}TTI(A_j(z_{kj})))$.

The transformations above have a very important property:

Proposition 5 Let $(\mathbb{C}^{*\times n},\vee,\wedge,\neg)$ be a logically complete matrix space of dimension n and $L=\{T,F,(,),\vee,\wedge,\neg,\alpha_1,\alpha_2,\ldots,\alpha_n\}$ a finite propositional language. Then:

- For any proposition $p\in L^*$ we have: $mod(p)=mod(TT(p))$.
- For any matrix $P\in\mathbb{C}^{*\times n}$ we have: $mod(P)=mod(TTI(P))$.

The above proposition shows that the transformation of a matrix to a logical expression and vice-versa does not cause any loss of (logical) information. This is true because any interpretation that satisfies a given matrix satisfies its respective logical expression (via the TTI function) and any interpretation that satisfies a given logical expression satisfies its respective matrix (via the TT function). This is true as far as logic is concerned; for knowledge representation things change, because, if RF information is available, TTI may cause information loss. Moreover, notice that the above proposition holds regardless of the selection of the operations $\vee$, $\wedge$, $\neg$.

We have already stressed the fact that matrices' elements can have negative real and/or imaginary parts. Such numbers indicate lack of confidence to a given literal and/or its negation. This means that they do not give direct information on the truth or falsity of a literal; instead, they indirectly imply its truth or falsity by specifying distrust in its falsity or truth respectively. Such kind of knowledge will be denoted by the term *negative knowledge* contrary to elements with non-negative parts (real and imaginary), which will be referred to as *positive knowledge*. The distinction is justified by the fact that logic can only express positive knowledge. Negative knowledge is only useful as far as knowledge representation is concerned, and its importance will be set forth in the next section.

It is not easy to define the operations $\wedge$, $\neg$ in the general case. In [9] we provide an elegant way to do so for matrices in $\mathbb{C}^{(+)*\times n}$, i.e. matrices expressing positive knowledge only. In fact, it has no real intuitive meaning to define such operations in the general case, though possible; remember that negative knowledge cannot be expressed in propositional logic. That is why we don't bother to define these operations explicitly; all we need is their existence in order to define the TT in the general case.

The above definition of the TT function will not be used in practice. TT cannot express the additional reliability information that we may have, nor can it express negative knowledge. This seriously reduces our expressive abilities. This is the reason we described belief revision in section 6 without bothering to define the transformation explicitly. TT, as defined in Definition 9 is only important as a theoretical construction, showing the relation between matrices and logic; it will also allow us to prove some results (see for example Proposition 8 in the next section). By restricting ourselves to positive knowledge some interesting results can be proved:

Definition 10 Let $A,B\in\mathbb{C}^{(+)*\times n}$, where $A=A^{(1)}|A^{(2)}|\ldots|A^{(k)}$, $B=B^{(1)}|B^{(2)}|\ldots|B^{(m)}$, for some $k,m\in\mathbb{N}^*$, $A^{(j)}\in\mathbb{C}^{(+)1\times n}$, $j\in\{1,2,\ldots,k\}$ and $B^{(j)}\in\mathbb{C}^{(+)1\times n}$, $j\in\{1,2,\ldots,m\}$. We define the operation $\wedge$ as follows:

$$A \wedge B = \mathop{|}_{h=1,j=1}^{h=k,j=m} (A^{(h)} + B^{(j)}) .$$

Proposition 6 Let $A,B\in\mathbb{C}^{(+)*\times n}$. Then: $\mathrm{mod}(A\wedge B)=\mathrm{mod}(A)\cap\mathrm{mod}(B)$.

Proposition 7 Let $A\in\mathbb{C}^{(+)*\times n}$. We define the operation $\neg$ recursively as follows:

- If $A=A_j(z)$ for some $j\in\{1,2,\ldots,n\}$, $z\in\mathbb{C}^{(+)}\cap(\mathbb{C}_+\setminus\mathbb{C}_-)$, then $\neg A=A_j(z\cdot i)$
- If $A=A_j(z)$ for some $j\in\{1,2,\ldots,n\}$, $z\in\mathbb{C}^{(+)}\cap(\mathbb{C}_-\setminus\mathbb{C}_+)$, then $\neg A=A_j(-z\cdot i)$

- If A=A_j(z) for some j∈{1,2,…,n}, z∈$\mathbb{C}^{(+)}\cap\mathbb{C}_0$, then ¬A=$F_n$
- If A=A_j(z) for some j∈{1,2,…,n}, z∈$\mathbb{C}^{(+)}\cap\mathbb{C}_*$, then ¬A=$A_j$(Re(z)·i)∨$A_j$(Im(z))

In the general case, let A∈$\mathbb{C}^{(+)m\times n}$, for some m∈$\mathbb{N}^*$ and:

$A=(\sum_{j=1}^{n}A_j(z_{1j}))\,|\,(\sum_{j=1}^{n}A_j(z_{2j}))\,|\ldots|\,(\sum_{j=1}^{n}A_j(z_{mj}))$ be its ENF. Then:

$$\neg A=\mathop{\wedge}_{h=1}^{m}\mathop{\vee}_{j=1}^{n}(\neg A_j(z_{hj})).$$

Under this definition, mod(¬A)=I(n)\mod(A).

With the three operations defined in Proposition 4, Proposition 6 and Proposition 7 at hand, we have defined a logically complete matrix space of dimension n (for positive knowledge only). It can be easily shown that under these three operations and for propositions in DNF, the simple, informal definition of TT given in section 5 coincides with the operation given by Definition 9.

Having completed the formalization of our knowledge representation scheme and its relation with propositional logic, we can now proceed to the definition of our revision scheme:

Definition 11 Let A,B∈$\mathbb{C}^{*\times n}$, where A=$A^{(1)}|A^{(2)}|\ldots|A^{(k)}$, B=$B^{(1)}|B^{(2)}|\ldots|B^{(m)}$, for some k,m∈$\mathbb{N}^*$, $A^{(j)}\in\mathbb{C}^{1\times n}$, j∈{1,2,…,k} and $B^{(j)}\in\mathbb{C}^{1\times n}$, j∈{1,2,…,m}. We define the operation of revision, denoted by •, between those two matrices as follows:

$$A\bullet B=\mathop{|}_{h=1,j=1}^{h=k,j=m}(A^{(h)}+B^{(j)}).$$

It is easy to verify that this definition concurs with the informal description of the revision operator done in section 6. Also, notice that by Definition 10, Definition 11 and Proposition 6, if we restrict ourselves to positive knowledge, it follows that: mod(A•B)=mod(A)∩mod(B).

Regarding the third part of our method, namely the querying of the knowledge stored in the KB, we already described in section 7 the general method followed in order to extract the knowledge contained in a KB matrix. The informal analysis made in the previous section revealed that the method followed is composed of six steps. More specifically, given a matrix A∈$\mathbb{C}^{m\times n}$ representing the knowledge contained in our base, we perform the following transformations upon A:

1. Apply the RE function on matrix A. The result is a new matrix B, in which all elements have been replaced by their reliability.
2. Apply the RL function on matrix B. The result is a column-matrix C in which each element represents the reliability of the respective line.
3. Apply the LS function on matrix C to get the set of indices S of the lines selected for further evaluation.
4. Select the matrix: D=MS(A,S)=MS(A,LS(C)), which is a submatrix of the original matrix A, containing only the selected lines.
5. Apply the MN function on matrix D to get matrix E, in which all elements of D have been replaced by their non-contradictory, normalized counterparts.

6. Apply the TTI function on the result to get $p=TTI(E)\in L^*$. Expression p is a satisfiable logical proposition, which represents our estimation on the knowledge currently stored in our KB. Expression p is the respective logical proposition of the original matrix A, as given by the query transformation function; in other words: $p=QT(A)$.

As the reader may notice, the first three steps of the above transformation are not clearly set. We claim that the operations represented by these functions are application-specific and cannot be universally defined. Arguments for this claim will be set forth in the following section. The definitions of MS and MN functions are straightforward:

Definition 12 We define the *submatrix selection* function $MS:\mathbb{C}^{*\times n}\times P(\mathbb{N}^*)\to\mathbb{C}^{*\times n}$. For any $k\in\mathbb{N}^*$, $A\in\mathbb{C}^{k\times n}$, $S\subseteq\mathbb{N}^*$, such that $A=A^{(1)}|A^{(2)}|\ldots|A^{(k)}$, $A^{(j)}\in\mathbb{C}^{1\times n}$, $j\in\{1,2,\ldots,k\}$, we define:

- $MS(A,S)=A$, iff $S=\varnothing$ or there exists $m\in S$ such that $m>k$,
- $MS(A,S)=\underset{j\in S}{|}A^{(j)}$, otherwise.

Definition 13 We define the *matrix normalization* function $MN:\mathbb{C}^{*\times n}\to\mathbb{C}^{*\times n}$. For any $k\in\mathbb{N}^*$, $A\in\mathbb{C}^{k\times n}$, such that $A=[a_{ij}]$, $i\in\{1,2,\ldots,k\}$, $j\in\{1,2,\ldots,n\}$, we define:

$MN(A)=B\in\mathbb{C}^{k\times n}$, where $B=[b_{ij}]$ and for all $i\in\{1,2,\ldots,k\}$, $j\in\{1,2,\ldots,n\}$: $b_{ij}=Re(a_{ij})-Im(a_{ij})$

In effect, the MS function returns a submatrix of A which consists of some of the lines of A, those whose indexes belong to the set S. In the abnormal cases where $S=\varnothing$ or S contains faulty indexes (out of range), MS returns the matrix A. Such cases will not appear in this application, but are included for completeness. The MN function is used to to remove any contradictions from the matrix, by comparing (subtracting) the real and imaginary part of a_{ij}. Finally, TTI has already been defined in Definition 9 and it is the final step to transforming a matrix to its respective logical expression.

9 Comments on the Method, Properties and Results

Some comments regarding this method are in order. First, the KB consists of a single matrix, thus constituting a simple and compact knowledge representation scheme. Second, the Principle of Primacy of New Information is overridden by the use of the RF and the definition of the MN function. In queries, given that the respective possible world is selected (by the LS function), the information with the largest RF will be kept, regardless of whether it resulted from the last revision or the old data. This is concurrent with the human intuition, as our beliefs, eventually, are unaffected by the order information is received. This fact additionally allows us to perform database merging in a straightforward way. The important problem of data rejection in contradicting revisions is also solved in the QT function. Rejection is only temporary, for the needs of queries, and depends on the RF and the functions RE and

RL, being minimal with respect to these quantities. The notion of minimality depends on the LS function selection.

Some of the most important results of our method originate from the inexplicit definition of the RE, RL and LS functions. These functions determine the lines to be used for queries, because they select the "important" lines of a matrix. The best selection is application-dependent. To see this, one could try to determine which of the (contradictory) elements 1+3i and 100+300i is more reliable (less contradictory). Many would argue that both elements are equally reliable, as the belief ratio of truth to falsehood of the atom is the same. Others would disagree, on the argument that the removal of the contradiction in 100+300i requires greater changes in the KB than in 1+3i.

In any case, the RE function will determine that. In an application where data is generally obtained through noisy channels, the contradiction in 1+3i is most likely to have occurred due to some random noise (error) of the input; on the other hand, 100+300i is less likely so, statistically, therefore it could be safely assumed that this element implies a real contradiction. In an environment where the world often changes dynamically, the contradiction in both elements may be due to information received in a previous state of the world; thus, they can be assumed equally reliable as they have the same belief ratio. In some other application where decisions are based on subjective opinions, instead of facts, the fact that the number 100+300i implies a bigger sample may force us to consider it more reliable than 1+3i.

The effect of the element reliability on the overall reliability of a line (RL parameter), as well as our tolerance in contradictory lines and the number of lines selected for use in a query (LS parameter) depends mainly on the reliability of the input devices. In a medical application, where the input devices are most reliable, even small contradictions should be "fatal" for the possible world that they belong; contradictions in such a sensitive application should be unacceptable, unless there is no other choice. This is not the case in applications with often dynamic changes of the world's state or with noisy input channels.

The ability to freely define RE, RL and LS functions provides a considerable flexibility in the extraction of the knowledge in a matrix. This flexibility allows relating the result of any given revision (and any given matrix) to several different propositions; as far as the user is concerned, this is equivalent to supplying different belief revision algorithms. Consequently, our framework provides a whole class of belief revision algorithms and the problem of finding a good such algorithm is reduced to the problem of finding a good way to extract the information from a matrix. The search for some interesting members of this class of algorithms is an ongoing work, but it has already been proven ([9]) that Dalal's algorithm ([2, 3]) gives the same results as our method for a specific parameter selection, as shown by the following proposition:

Proposition 8 Let $p,q \in L^*$ be two satisfiable propositional expressions in DNF and let r be the revision of p with q under Dalal's algorithm ($r=p\bullet^D q$). Moreover, let $P \in \mathbb{C}^{(+)^* \times n}$ the matrix related to p via the TT function, using an RF of 1 for all atoms ($P=TT(p)$), $Q \in \mathbb{C}^{(+)^* \times n}$ the matrix related to q via the TT function, using an RF of 2 for all atoms ($Q=2\cdot TT(q)$) and $R \in \mathbb{C}^{(+)^* \times n}$ the matrix resulting by the update of P with Q under our framework ($R=P\bullet Q$). Under these RF selections, there exist selections for the functions RE, RL and LS such that the resulting propositional expression (to be used in queries) is logically equivalent to the expression r as defined above, that is: QT(R)≅r.

The proof of this proposition, as well as the definition of one possible set of RE, RL and LS functions that satisfy it is given in [9]. Moreover, it has been proven in [13] that Dalal's algorithm satisfies all 8 AGM postulates; this implies that there exist parameters under which our method satisfies the AGM postulates for revision. The general conditions under which this is true is an ongoing work. The phrasing of the above proposition also implies the important fact that the selection of the RFs of a matrix's elements has great effects on the result of a revision. An important property of our scheme, closely related to the selection of the RF, is the introduction of negative knowledge. The disbelief expressed by such type of knowledge cannot be expressed in propositional logic, so this is a direct improvement of our expressive abilities over the conventional knowledge representation schemes and may be used to express operations like *contraction*, *update* and *erasure* [13].

As already noted, our scheme deals with all these operations interchangeably. In order to emulate such operations all we have to do is change the revision matrix' RF in such a way as to signify their different nature. For example, contraction indicates loss of knowledge in certain facts. We can signify this by revising with matrix –M whenever we want to contract M. Update is a bit more difficult; this operation indicates a dynamic change in the world described by the KB. This means that any knowledge we may have regarding the world is irrelevant and should be ignored whenever it conflicts the new knowledge, because it refers to a previous state of the world. By the above comments, it follows that matrix M should be "enhanced" in such a way as to remove the old knowledge as well as to add the new one.

We will see how this is possible with an example, which will also show the dominating effect of the RF selection and negative knowledge on the result of a revision as well as the power of parameterization. The reliability of an element x+yi will be defined as |x-y|, i.e. the difference between its true and false part. The reliability of a line will be defined as the sum of the reliabilities of all its elements, so all elements are considered for the calculation. Finally, LS will select the lines with maximum reliability.

Consider the example of section 3, about the room with lamps A and B and the knowledge that exactly one of them is on. Our knowledge could be expressed by: $p=((\neg\alpha)\wedge b)\vee(\alpha\wedge(\neg b))$. Let us also assume that through observation we conclude that A is on. The proposition representing the revision is: $q=\alpha$. Intuitively, the result of this revision should be $p'=\alpha\wedge(\neg b)$, as shown in section 3. By assigning an RF of 1 on all

$$P=\begin{bmatrix} i & 1 \\ 1 & i \end{bmatrix}, Q=\begin{bmatrix} x & 0 \end{bmatrix}, P'=P\bullet Q=\begin{bmatrix} x+i & 1 \\ 1+x & i \end{bmatrix}$$

atoms of the KB and an RF of x>0 on the update we get:

Note that the first line contains a contradictory element (x+i), whereas the second contains no contradictions. Using the parameterization above, we get reliabilities of |x–1|+1 for the first line and |x+1|+1 for the second. For all x>0 it holds that |x–1|+1<|x+1|+1, thus LS will only select the second line. Upon applying MS function, P′ matrix will be mapped to P″=[1+x i] (containing only the second line of P′).

Applying the MN function will have no effect on P′′, as there are no contradictions to resolve, thus the respective proposition of P′ is: QT(P′)=α∧(¬b), as expected.

Alternatively, the RL function could be defined as the minimum over the reliabilities of all elements in the line (only the least reliable element of the line is considered in the calculation). In this case, for x≥2 both lines would have equal reliability (equal to 1), so the LS function would select them both. Thus, for this selection of the RL function, the RF of Q is crucial; for x≥2 we get QT(P′)=(α∧b)∨(α∧(¬b))≅α, whereas for 0<x<2 we get QT(P′)=α∧(¬b).

A more intuitively correct approach for the LS function would be to select the non-contradictory lines only; if there are no such lines, then we go on by selecting the most reliable contradictory ones, as before. It can be verified that in this case, the result would have been QT(P′)=α∧(¬b), regardless of the selection of the parameters RE, RL or x.

Continuing this example, notice that if we had sent a robot into the room with the order "turn lamp A on", then we should *update* (not revise) p with q=α. For the reasons described in section 3, the proper intuitive result of the update is p′=α. Matrix Q′=[1–i 0], corresponds to the same information as Q, because TTI(Q′)=α=TTI(Q). However, Q′ includes some disbelief in the negation of α because the imaginary part of 1–i is negative (–1). Thus, Q′ contains the additional information that the action of the robot voids the previous state of A, if A was off, which is the notion behind update. The revision of P with Q′ under our scheme gives:

$$P' = \begin{bmatrix} 1 & 1 \\ 2-i & i \end{bmatrix}$$

The important point here is that there are no contradictory lines in P′, therefore, by using the rational LS function previously described (regardless of the RE and RL selection), we would select both lines of P′, so QT(P′)=(α∧b)∨(α∧(¬b))≅α. This is the expected result of an *update*. In a similar manner (using negative information) we can get a *contraction* operation, as described above. Finally, similar considerations can lead to the integration of the operation of *erasure*, which is dual to update. This fact eliminates the need for additional operators, as they can all be defined in terms of revision. Moreover, we can perform partial operations, by, for example, contracting knowledge regarding some atoms and revising others (or any other combination), a property not available in conventional revision schemes.

Finally, we can conclude that iterated revisions are inherently supported by our method. Each line in the KB matrix contains the additive information over all revisions (for a certain combination of possible worlds) regarding the truth and falsehood of each element (atom). By not removing contradictory lines from the KB, we lose no data regarding past revisions, because no possible world combinations are eliminated. Such contradictory data may be useful in future revisions.

Assume for example the propositions: p_1=α↔b, p_2=α↔c, and p_3=b↔c, with a reliability of 1, as well as the proposition p_4=α↔(¬b) with a reliability of 2. It is easily verified that the intuitively correct result for the updates p_1•p_2 and p_1•p_3 is α↔b↔c. If we subsequently revise with p_4, the intuitively correct result is different in each case. Specifically, given the increased reliability of p_4, we should have (p_1•p_2)•p_4=α↔(¬b)↔c and (p_1•p_3)•p_4=(¬α)↔b↔c. Most revision schemes (Dalal's operator for example) would give $(p_1\bullet p_2)\bullet p_4=(p_1\bullet p_3)\bullet p_4=\alpha\leftrightarrow(\neg b)$, thus losing all information regarding c. This happens because Dalal's algorithm does not support iterated revisions.

Let us assume that the reliability of x+yi is |x–y|, the reliability of a line is the sum of the element reliabilities and LS is the rational LS function described in the previous example. It is left as an exercise to verify that, under these parameters and with the RF information described above, our revision operator gives the intuitively correct results for $p_1\bullet p_2$, $p_1\bullet p_3$, $(p_1\bullet p_2)\bullet p_4$ and $(p_1\bullet p_3)\bullet p_4$. The additional information that is needed in order to do the second revision correctly is "hidden" in the contradictory lines of the first revision in both cases. If such lines were permanently discarded after the first revision, we would get the same (incorrect) result as Dalal's operator. A detailed proof of these facts is omitted due to lack of space.

Unfortunately, there is an annoying fact about the above statement. We *should* keep all the possible worlds in a matrix, however, the definition of the revision operator implies that the number of lines in a matrix is exponential with respect to the number of revisions performed. This means that the KB matrix size will soon become too large to be manageable, unless some action is taken. As a solution, we propose the technique of *abruption*, a procedure that permanently prunes some lines from the KB. The selection of the lines to be removed could be made using functions similar to the RE, RL and LS functions. However, we should bear in mind that such removal always implies loss of knowledge; only the lines considered too far from the real world should be removed. Such lines are considered so wrong that they will never affect the result of the QT, so they can be "safely" removed. In any case, the use of abruption implies a trade-off between knowledge integrity and processing speed and should be carefully designed. Obviously, this is another application-specific parameter, for which no universally defined function can work satisfactorily.

10 Conclusions and Future Work

In this paper, we used an innovative representation of propositional expressions to address the problem of belief revision. The approach was proven fruitful, resulting in a very flexible and general-purpose method of revising beliefs. The introduction of the Reliability Factor (RF) and the quantitative nature of the table representation introduce an increased expressiveness in propositional logic, allowing the use of features not normally available, like negative knowledge or the integration of update operators.

We believe that much more work needs to be done in order to fully exploit this representation's capabilities. The behavior of our algorithm under different parameter selections (RE, RL and LS functions, RF selection, abruption effects etc) is only partially explored. Such a study may reveal interesting connections between our method and existing approaches, such as the general conditions under which AGM postulates are satisfied. It would also allow us to formulate some general, formal methods regarding the integration of update operators informally described above. Moreover, the study of problems and concepts of propositional logic under this new representation might give some important results. Finally, implementation issues should be considered, such as proper user interaction techniques for input (mainly) and output, or efficient data structures for the storage and manipulation of large matrices.

References

1. Alchourron C., Gärdenfors P., Makinson D., "On the Logic of Theory Change: Partial Meet Contraction and Revision Functions", The Journal of Symbolic Logic 50, pp.510-530, 1985.
2. Dalal M., "Investigations Into a Theory of Knowledge Base Revision: Preliminary Report", In Proceedings of the Seventh National Conference on Artificial Intelligence, pp. 475-479, 1988.
3. Dalal M., "Updates in Propositional Databases", Technical Report, DCS-TR-222, Department of Computer Science, Rutgers University, February 1988.
4. Darwiche A., Pearl J., "On the Logic of Iterated Belief Revision", UCLA Cognitive Systems Laboratory, Technical Report (R-202) In R. Fagin (Ed.), Proceedings of the 1994 Conference on Theoretical Aspects of Reasoning about Knowledge (TARK '94), Pacific Grove, CA, 5-23, March 13-16, 1994. In Artificial Intelligence, 89(1-2):1--29, 1997.
5. Delgrande J., "Considerations on a Similarity-Based Approach to Belief Change", 16th International Joint Conference on Artificial Intelligence (IJCAI'99), Stockholm, Sweden, pp. 180-185, August 1999.
6. Fagin R., Ullman J., Vardi M., "On the Semantics of Updates in Databases", Proceedings of Second ACM SIGACT-SIGMOD, Atlanta, pp. 352-365, 1983.
7. Flouris G., "Belief Revision in Knowledge Bases", in Greek, M.Sc. Thesis, Computer Science Department, University of Crete, October 2001, also available as Technical Report TR-292, ICS-FORTH, October 2001, http://zeus.ics.forth.gr/forth/ics/isl/publications/paperlink/flouris_thesis.pdf.
8. Flouris G., Plexousakis D., "Belief Revision in Propositional Knowledge Bases", 8th Panhellenic Conference on Informatics, Nicosia, Cyprus, November 2001.
9. Flouris G., Plexousakis D., "Belief Revision using Table Transformation", Technical Report TR-290, July 2001, ICS-FORTH, http://zeus.ics.forth.gr/forth/ics/isl/publications/paperlink/flourisTR-290.pdf.
10. Gärdenfors P., "Belief Revision: An introduction", pp. 1-20 in Belief Revision, ed. by P. Gärdenfors, Cambridge University Press, 1992.
11. Gärdenfors P., "The dynamics of knowledge and belief: Foundational vs. coherence theories", Revue Internationale de Philosophie 44, pp. 24-46. Reprinted in Knowledge, Belief and Strategic Interaction, ed. by Bicchieri and M. L. dalla Chiara, Cambridge University Press, Cambridge, 1992, pp. 377-396.
12. Hansson S. O., "In Defense of Base Contraction", Synthese 91, pp. 239-245, 1992.
13. Katsuno H., Mendelzon A., "On the Difference Between Updating a Logical Database and Revising it", in Belief Revision, ed. Gärdenfors P., Cambridge Press, 1992.
14. Katsuno H., Mendelzon A., "Propositional Knowledge Base Revision and Minimal Change", KRR-TR-90-3, March 1990, Technical Reports on Knowledge Representation and Reasoning, University of Toronto.

15. Liberatore P., "The Complexity of Iterated Belief Revision", In Proceedings of the Sixth International Conference on Database Theory (ICDT'97), pages 276-290, 1997.
16. Makinson D., "How to Give it up: A Survey of some Formal Aspects of the Logic of Theory Change", 1985, Synthese 62, pp. 347-363.
17. Nebel B., "A Knowledge Level Analysis of Belief Revision", In Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning, pp. 301-311, 1989.
18. Williams M., "Anytime Belief Revision", 1997, International Joint Conference on Artificial Intelligence, 1997.
19. Williams M., Williams D., "A Belief Revision System for the World Wide Web", Web: http://u2.newcastle.edu.au/webworld/ai-internet.html.