A Compromised Characterization to Base Revision

Fátima C. C. Dargam
e-mail: fccd@doc.ic.ac.uk
Department of Computing
Imperial College of Science, Technology and Medicine
180 Queens Gate, London SW7 2BZ
United Kingdom

1 February 1996

Abstract

This paper proposes a compromised revision method for handling logically conflicting inputs into knowledge bases. Basically, it concerns reconciling conflicting inputs with the underlying theory, via restricting their consequences. The main idea is to update the database with as many consistent consequences of the inputs as possible, in the case that the inputs themselves are not allowed to be kept in it. And in the case that a revision applies, the idea is to keep as many as possible of the consistent consequences of the retracted sentences as a compromise.

Resolving conflicting updates in dynamic databases, for instance, are frequent and critically important problems of real applications. Such problems require the revision of theories and knowledge bases. It is not realistic to aim for a generic approach in those cases, since theory revision is fundamentally dependent on application-specific mechanisms, principles and heuristics. The approach we propose here, caters for the specific case where compromised solutions for revising knowledge bases apply, when conflicts involving updates occur. In comparison with approaches that require preference between conflicting inputs, or that avoid them by cancelling them out completely, our approach fits as an alternative which provides more informative results. Examples of inputs include database updates, actions, and beliefs.

Here, we introduce a compromised characterization to revision under the perspective of belief revision. We define a base-theory revision operator ® for specific applications which allow for compromised solutions, and a compromised contraction for revision. Correspondence theorems and some propositions establish relevant relationships among functions, postulates and some relations of this characterization. ® proposes a specific revision method which applies some compromise criteria for achieving the revised theory. Hence, the contribution of the ® compromised revision re-enforces the importance of having different theory change operators available for specific applications, in order to concieve the construction of a more realistic framework for theory revision.

1 Introduction

Resolving conflicting updates in dynamic databases, or conflicting actions in planning applications, for instance, are frequent and critically important problems of real applications. Such problems require the revision of theories and knowledge bases. As pointed out by Winslett in [Win-90], it is not realistic to aim for a generic approach in those cases, since theory revision is fundamentally dependent on application-specific mechanisms, principles and heuristics. The approach we propose in this paper, caters for the specific case where compromised solutions apply when conflicts occur. It supports a compromised way of handling conflicting updates for revising knowledge bases, what we call *compromised reasoning*[1].

In more practical terms, consider the situation where DB is a database and A an input. Assume that A is inconsistent with DB. Current belief revision/update approaches will keep A and mantain consistency by selecting some element from DB to form a revised database, usually denoted as $DB * A$. There is a lot of research in this area, both theorectical, e.g.: the AGM theory of belief revision[2], and algorithmic research, e.g.: Reason Maintenance Systems[3]. Our aim is to offer an alternative approach, which is flexible enough to keep more data in DB, in the case of conflicts. We view the above situation as a *conflict* between two inputs (DB and A) into an empty database, and we tackle the problem of reconciling these inputs. Under our approach, the conflicting input A is kept in DB only in the case that A generates inconsistency to DB indirectly[4], in which case a revision also applies in order to restore consistency. However, in the case that A is not allowed to be kept in DB, its eventual consistent consequences, w.r.t. the existing data of DB, are added to the database under the compromised policy of our approach. This way, instead of preventing updates to be performed, when they introduce inconsistency to the system, our approach reconciles the conflicting inputs by compromising on their consequences. We propose to generate the consequences of the conflicting inputs, and get rid of the inconsistency, via a minimal number of retraction of those consequences. We expect the resulting database to be consistent w.r.t. the integrity constraints, and to retain a maximal subset of the consistent consequences of those updates. This reconciliation of conflicting inputs follows some specified postulates for compromised revision.

As pointed out by Galliers [Gal-90], in most of the existing AI research work, conflicts either simply never arise, or are alternatively avoided when they do arise. However, in a constantly changing and unpredictable environment where centralised control over the entire system is most of the times impossible, inconsistencies within the system are inevitable, and conflict situations do arise. The central interest in [Gal-90] is to solve conflicts in cooperative multiagent systems, by facing their positive aspects. They claim that achievement of cooperation from conflict, among formalized agents may involve the decision of a mutually preferred compromised solution, and/or persuasion to the validity of another position. These attitudes are, undoubtedly, considered as natural among humans, in a cooperative situation. Moreover, based on studies about human society, they also claim that conflict relations constitute a fundamental social interaction process and they have important functions and consequences.

Gabbay and Hunter, in [GaHu-91,93], also support that inconsistency should be faced and formalized. They urge for a revision on the way inconsistency is currently being handled in formal logical systems, as opposed to the way it is handled by humans. They claim that there is a need for the development of a framework, in which inconsistency can be viewed in a context-dependent way. As a signal for external and/or internal actions, and not necessarily as a bad element which induces the whole system to collapse. They argue that

[1] The basic idea of *compromised reasoning* is detailed further in this section.

[2] The AGM theory was first introduced in [AlMa-82,85,86][AGM-85], and since then gained many followers who apply and modify that theory in various ways, see for instance [Mak-85],[Neb-89] [RaFo-89],[JaPa-90],[Neb-90],[Rot-91,92],[KaMe-92] [BoGo-93],[Mak-93],[Sri-93] and [FrLe-94].

[3] Reason Maintenance Systems were initiated in [Doy-79] based on justifications, and in [Kle-86a] based on assumptions. More recent research work following this line have also emerged. Some of them are found in [Elk-90],[GiMa-90],[PiCu-89],[RoPi-91],[SaIw-91] and [WaCh-91].

[4] By *indirectly* here, we mean that A does not directly violate any of DB's integrity constraints.

dealing with inconsistencies is not necessarily a job for restoring consistency, but rather for supplying rules which state how to act in the case of inconsistencies. Examples of some circumstances in which inconsistency can be desirable, meaningful and useful, provided that the system is capable of activating appropriate actions to handle it, are presented in their work.

We strongly endorse the viewpoints of Galliers and Gabbay & Hunter. Based on the same grounds, we investigate an approach which handles conflicts that introduce inconsistency into a system, and puts forward a compromised reasoning way for dealing with conflicting updates and actions, instead of simply avoiding them.

As in [Gal-90], we propose to solve conflicts by facing their positive aspects. We do so, by reconciling the conflicting updates with the underlying knowledge base, and getting as many of their consequences as possible.

We support the point in [GaHu-91,93], that inconsistency (caused by conflicts) should be faced and that we should supply mechanisms for handling situations when they arise. In the current work, we approach such situations by allowing some consequences of the conflicting updates to remain in the database. However, by reconciling the conflicting inputs, we also restore consistency, which does not conform to their view of keeping inconsistency in the system and supplying the appropriate mechanisms to handle it[5].

Our main motivation in pursuing this approach, comes from the premise that by reconciling conflicting updates with a knowledge base, our approach provides more informative results. In comparison with approaches that require preference between conflicting inputs, or that simply avoid them by cancelling them out completely, our proposal of compromised revision allows more information to be kept in a theory base.

Following our compromised revision approach to conflicts, one will possibly not get all of what he/she originally wanted[6], in the case of conflict. Instead, he/she will get extra data, leading to the direction of the original goal. This is so, because most of the extra data are related to the goal's consistent consequences.

The results we get with our approach suit the needs of particular application areas, e.g. design processes; resource allocation; and decision making.

We believe that this approach is appealing, based on the following premises:

- We derive a compromised solution to the problem, which presents some degree of impartiality.[7] And we consider that impartiality also plays an important role when deciding what to do in the case of conflict. It is not always the case that we have an established criterium, such as priority, preference, seniority, etc, to choose between conflicting updates.
- By compromising on the performance of conflicting updates, we can keep track of the original intention within the maximal set of consistent consequences which is generated. Hence, we gain the property of building up, systematically, from the initial compromised solution, further in the direction of the original goal, via subsequent compromised results.

As an application example, let us consider a research organization which has the task of deciding the allocation of funds among projects. We assume that it is necessary for the projects to discriminate all the expenses required for each of their phases, allowing for the option of satisfying only partially those phases (compromised solutions). We assume also that the decision makers are not supposed to favour any project in particular. So, if funds are not sufficient to support all the projects' requirements, our approach would be appropriate to be applied in the process of funds allocation. In the sense that it would allow for as many of all the projects' phases as possible, considering the constraints involved in the process.

[5] This view is planned as part of our further work in this approach.

[6] The idea of this revision approach conforms with the meaning of the word *compromise*. Quoting from the Oxford's Dictionary: *"Compromise" is a settlement of a dispute which each side gives up something it has asked for, and neither gets all it asked for.*

[7] This is due to the fact that the system does not simply choose any particular input, out of the conflicting ones, to be performed.

1.1 Handling Inconsistency

The majority of the existing approaches for conflict resolution and for reasoning in the presence of inconsistency in data or knowledge bases, seems to have a common concern. The one of maintaining consistency of their databases or knowledge bases, in face of contradictory information. Only a few of the existing approaches propose to keep inconsistency in the data / knowledge base, and handle it appropriately in order to supply desirable results. The formalisms which can handle inconsistent data, not necessarily as part of their knowledge bases, but as a means of reaching a revised and consistent resulting base, take somehow advantage of the conflicting input.

Examples of studies which deal with inconsistency, are the work by Baral & Subrahmanian [BaSu-92], Cholvy [Cho-90], and Bauval & Cholvy [BaCh-91] in the context of theory combination, mainly based on the maximal consistent subsets of the global theory. The technique of maximal consistent subsets, supporting the minimal change notion, is also used by Fagin et al [FUV-83] and Kupper et al [KUV-84], in the context of database updates. Gardenfors [Gar-88] and Nebel [Neb-89] also use this technique, in the area of belief revision. Ginsberg & Smith [GiSm-88], in the area of reasoning about actions. The work of Besnard [Bes-90], which uses paraconsistent logic. The work by Cerro & Herzig [CeHe-86,92], which uses modal logic for reasoning about updates in the presence of contradictory data. The work by Gabbay and Hunter [GaHu-93], which presents a meta-level system, based on first-order linear temporal logic, for handling inconsistent data. And the work by Wagner [Wag-93b], which considers formalisms which are closest to defeasible inheritance systems, where contradictory pieces of information neutralize each other.

Here, we pursue the line of using a logical framework for reasoning about updates in the presence of contradictory data. We eliminate inconsistency, by managing *safe maximal sets* within a reconciling strategy, which allows for consistent consequences of conflicting inputs to be kept in the resulting base. Due to this strategy, our approach can be classified as non-monotonic.

In general, conflict resolution is fundamentally contextual-dependent. In Belief Revision, the main concern is to solve the problem of revising derived beliefs whenever the underlying set of beliefs changes. The approaches in this area, adopt a particular revising policy as a strategy, for instance in terms of temporal priority of facts, in order to restore consistency whenever the underlying set of base beliefs is modified.

Generally, in belief revision, the input sentence to revise the belief set is supposed to contradict some of the sentences already present in this set. Hence, the only way to accept the new sentence and still keep the resulting set consistent, is to give up the belief in the old contradicting sentences. In the AGM theory [AlMa-82,85,86][AGM-85], they introduce their revising strategy by means of a set of postulates, which can be viewed as dynamic integrity constraints or transitions laws. Nevertheless, in the case of contractions, the postulates just provide a minimum set of requirements that a contraction function should satisfy. They do not uniquely determine a contraction function. Hence, the problem of choosing what to retract in order to avoid inconsistency has to be tackled by other means. Epistemic Entrenchment is a proposed way to deal with this problem [GäMa-88].

1.2 Our Approach to Handling Conflicting Inputs

Our approach proposes to reconcile conflicting inputs with respect to the underlying theory, and establishes some policies for dealing with the problem of inconsistency caused by them. The way we are approaching the problem of conflicting inputs, differs from the other existing approaches, in the sense that we allow for a special process of performance of the conflicting updates. A process of *reconciliation of conflicting inputs*, which considers restrictions of the effects of those inputs by compromising on their consequences. It allows for a maximal set of consistent consequences, which are generated from those conflicting updates with respect to the existing data, and to the integrity constraints which range over the database.

In general, we refer to our approach as CIU, meaning *Compromising Interfering Updates*.

By conflicting, or interfering, updates, we mean either simultaneous updates which interfere with each other, generating inconsistencies as part of their combined effects, or updates which are inconsistent to be performed because they conflict with the given database or scenario representation, by violating some of their constraints.

We are also concerned about the specific case of allowing for the special performance of simultaneous updates when they are individually consistent to be applied, but if performed in parallel they interfere with each other. In this case, we postulate that if an update satisfies its pre-conditions to be performed individually, it should also be allowed to be performed in parallel with other updates, as long as the performance process copes with inconsistency handling in the case of conflicts. A similar motivating approach was pursued by Cholvy [Cho-93] in the context of multi-sourced information environment. Cholvy treats the problem of consistency of information provided by different sources, considering the case that the global set of information is inconsistent even if each separate source is consistent. Notice, however, that in this work we propose to deal with inconsistency generated by conflicting updates within the same system, while Cholvy treats the inconsistent information which is due to the combination of different data/knowledge bases. A further analogy between the two approaches requires, at least, a re-definition of the basic conflicting entities, in order to cater for the representation of information sources.

Below, we present an example which illustrate the intuitive notion of our approach with relation to database updates.

Example 1.1 *Let us consider the database of formulae as shown below:*

$$\begin{array}{ll} (1) & A \rightarrow B \\ (2) & A \wedge C \rightarrow \perp \\ (3) & A \end{array}$$

If we want to update this database with the formula (4) C, then, by applying a TMS*-like approach [Doy-79], for instance, we would force C in, by removing A and all the consequences derived from A, in order to keep consistency, as shown below.*

$$\begin{array}{ll} (1) & A \rightarrow B \\ (2) & A \wedge C \rightarrow \perp \\ (4) & C \end{array}$$

In the way we approach this problem, we would also end up with either A or C, but not both. However, we want to be able to keep all the consistent derived consequences of the conflicting update. In this case, we would be able to have B as well in the resulting database, as shown below.

$$\begin{array}{ll} (1) & A \rightarrow B \\ (2) & A \wedge C \rightarrow \perp \\ (4) & C \\ (5) & B \end{array}$$

•

Remark 1.1

The example above could be interpreted with the following meanings for the sentences A, B and C:
A = "Executive Class Passenger";
B = "Extra baggage allowance";
C = "Economic Class Passenger".
Then, we would have that the database update above represents a situation, in which an executive class passenger for some reason has to be moved to the economic class, in a particular flight. However, even in the the economic class, the originally executive class passenger still keeps his/her right of having extra baggage allowance.

•

1.3 Compromised Reasoning

Compromised solutions for conflicting updates and actions meet our intuitions in different application areas. For instance, we can think of examples in the legal context, where one might want to have some consequences of actions, without performing them as they were originally specified, because they may contradict some laws or some restrictions of the underlying legal system. In this situation, one would go for a compromised solution for the performance of those actions.

Basically, compromising on the consequences of updates and actions in the case of conflict, is a suitable way for building up intermediate stages of development processes in general. If we think of applications in the area of *design*, for instance, the extra information that we get, in keeping as many of the consistent consequences of conflicting updates/actions as possible, may help us to build up an intermediate phase of the whole process, and to reduce the total amount of time needed for concluding the task under consideration.

Consider, for instance, the application of Printed Circuit Board Design, and the task of positioning integrated circuits and other components on a board. Suppose that we want to place a component, say IC1, on a void area of the board, where no components are allowed to be placed. Let us assume that this particular IC invokes the placement of other components, according to some rules and placement requirements of the board design. Hence, if we allow for the consistent consequences of the action of placing IC1 on the void area, we will end up having the placement of further components, which certainly speeds up the whole process. We will also have more information available for further components placement, than we would have if we simply refused the inconsistent action.

We can also think of compromising consequences in applications where updates or actions are restricted by the availability of common resources. In these cases, because of lack of enough resources, it might not be possible to get all the consequences for all the actions. The approach, then, is appropriate to be applied, when it is preferable to get as many of the consequences of all the actions as possible, and not just the consequences of only some of the actions. Some applications, which show these characteristics, are the ones which involve decisions about distribution of resources among different candidates, in the case that impartiality is encouraged. That is, when it is not preferred that one single candidate gets all the resources available and the others get none. The example described earlier, considering the task of allocation of funds among projects in a research organization, fits in this sort of application.

In general, via the compromised reasoning strategy of our approach, we are able to keep bits of new information, or a subset of the consequences of updates, which could not be totally assimilated, or performed, due to a conflicting situation. These consequences and bits of new information might reveal important facts about the current situation, which then might signal the system to take specific actions, depending on the application context.

1.4 Different Kinds of Conflicting Inputs

Conflicting inputs can be of various kinds. For instance, we can have simultaneous updates which interfere with each other, generating inconsistencies as part of their combined effects, or updates which are inconsistent to be performed because they conflict with the given database, by violating some of its constraints. We can also have the case in which updates are individually consistent to be applied, but if performed in parallel they interfere with each other.

Before describing the adopted policies for reconciling conflicting updates with the underlying database, we need to state clearly all the different kinds of conflicting inputs that we are considering, and also to describe how we propose to handle them.

For the examples below, consider that A and B are formulae of the language being considered. We consider classical logic as the underlying logic, including the usual connectives. A database DB is such that $DB = \Delta \bigcup P_\Delta$, where Δ denotes set of formulae which compose the body of the database, and P_Δ denotes the set of integrity constraints which rules Δ.

The different kinds of conflicting inputs we can get are the following:

(a) Conflicting inputs within the update, or within the transaction.

Example 1.2 $Update = \{A, \neg A\}$

•

(b) Inputs which conflict directly with some of the integrity constraints which rule the database.

Example 1.3 $DB = \Delta \bigcup P_\Delta$
$P_\Delta = \{ A \rightarrow \perp \}$
$Update = \{A\}$

•

(c) Inputs which conflict indirectly with some of the integrity constraints which rule the database.

Example 1.4 $DB = \Delta \bigcup P_\Delta$
$P_\Delta = \{ A \wedge C \rightarrow \perp \}$
$\Delta = \{ C \}$
$Update = \{A\}$

•

(d) Inputs which contradicts existing data of the database.

Example 1.5 $DB = \Delta \bigcup P_\Delta$
$\Delta = \{ A \}$
$Update = \{\neg A\}$

•

The way we propose to handle the above different conflicting inputs in our approach is as described below. We always have to bear in mind that the main idea of the approach is to update the database with as many consistent consequences of the conflicting inputs as possible.

(a) In the case of conflicting inputs within the update, or the transaction, the updates are rejected, since one logically cancels the other. However, if within a transaction T we have the following sequence of inputs $T = \{A, \neg A, B\}$, the subset $\{A, \neg A\}$ is removed from T and the remaining inputs in the transaction are still performed. In this case, $T = \{B\}$.

(b) In the case of inputs which conflict directly with some of the integrity constraints which rule the database, the input is not allowed to be inserted in the database. However, we would still have two optional lines to follow:
Option 1 - We reject the update completely and no changes are performed on DB.
Option 2 - We allow the consistent consequences of the input to be inserted in DB, with particular status of non-supported data. AS expected, our approach follows Option 2, since we end up having more consistent consequences of the conflicting input in the database.

(c) In the case of inputs which conflict indirectly with some of the integrity constraints which rule the database, the input is inserted in the database and a revision procedure takes place in order to restore consistency and allow the database to accomplish the new update. The revision presents special properties which preserves the consistent consequences of all the retracted formulae from DB.

(d) In the case of inputs which contradicts existing data of the database, the input is inserted in the database and a revision on DB takes place, just as described in the item above.

We describe below the adopted policies for reconciling conflicting updates with the underlying database, under a compromised reasoning perspective.

1.5 Compromised Reasoning Policies for Updates

The compromised reasoning policies for updates guide our approach in the sense of restricting the effects of those updates, by imposing restrictions on their consequences in order to keep consistency in the case of conflicts.

Single Update Case

Consider that an update U is to be performed to a database DB, adding some data into DB. DB is assumed to be initially consistent. Assume that when U is performed to DB, the updated database violates an integrity constraint IC of the set of integrity constraints which range over DB. Our concern is to reconcile U with DB, by allowing a version of this update to be performed in a compromised way, following the policies below:

- If U violates the integrity constraint IC indirectly [8], then a revision on DB takes place, allowing U to be kept in DB , and also the consistent consequences of the data which might have to be retracted, so that DB is able to incorporate U and still be consistent.
- If U violates the integrity constraint IC directly, then U is not allowed to be added to DB, yet its consistent consequences are kept in DB .
- The consequences of the data which are retracted from DB in order to accomplish a compromised update, carry a distinguished status in the database, so that further derivations can take their status into account in the case of generating further consequences. [9]
- The revision mechanism has to be defined in a way, such that a maximal consistent subset of generated consequences of the conflicting update is available from the resulting database.

Multiple Updates Case

In the case of a transaction, which involves a set of single updates, if we have "n" conflicting updates w.r.t. the integrity constraints, the resulting compromised updated database might contain at most "$n-1$" of those updates, and their subsequent consequences, provided that they all satisfy the constraints on D_{CIU}.

Transaction updates have their consistency initially checked with relation to the three conditions described below. Assume that a transaction $T = \{U_1, U_2, \cdots, U_n\}$, composed of n updates, is to be performed to DB, and I is the set of integrity constraints which rules DB. (The formulae considered here are propositional sentences from the system of propositional classical logic).

1. For any U_i and any U_j in a transaction $T = \{U_1, \cdots, U_n\}$, where $1 \leq i \leq n$; $1 \leq j \leq n$; and $i \neq j$, if U_i expresses a formula which is the complement of the formula expressed by U_j, say A and $\neg A$ [10], then the set $\{U_i, U_j\}$ is retracted from the transaction T.
2. For any U_i in a transaction $T = \{U_1, \cdots, U_n\}$, where $1 \leq i \leq n$, if U_i violates an integrity constraint in I, $\{U_i\} \bigcup I \vdash \perp$, then the update U_i is rejected, however its consistent consequences are allowed to remain as non-supported consequences in the database.

[8] Suppose that the update U represents a formula α , which conflicts with a database DB . If there is an integrity constraint IC in DB of the form $\alpha \rightarrow \perp$, then we say that the update U violates IC directly. Otherwise, U is said to violate IC indirectly. For instance, in the case that U is an atomic formula α, and the integrity constraint that U violates has a conjunction of atomic formulae, which includes α, as its antecedent part.

[9] The handling of these compromised consequences is under the responsability of the derivation mechanism of the approach. The compromised consequences are also referred to as *non-supported consequences*.

[10] This would represent adding and deleting the formula A in the same transaction.

3. For any U_i and U_j in a transaction $T = \{U_1, \cdots, U_n\}$, where $1 \leq i \leq n$; $1 \leq j \leq n$; and $i \neq j$, such that U_i and U_j are not complementary updates in T, and $\{U_i\} \bigcup I \not\vdash \perp$ anf $\{U_j\} \bigcup I \not\vdash \perp$, if $\{U_i, U_j\} \bigcup I \vdash \perp$, then a choice is made between U_i and U_j, according to meta-level information concerning, the relevance of the updates within the transaction[11]. In this case, the transaction is then reduced to $T - \{U_k\}$, where k is either i or j, depending on this meta-level based choice, however the consistent consequences of U_k are allowed to remain as non-supported consequences in the database.

In the cases described above, when consistency of the updates is not obtained initially, the updates which cause inconsistency are not supposed to be performed, since they are removed from the transaction, as described. However, the transaction is not cancelled due to the fact that some of its updates failed the initial consistency checking phase[12]. Moreover, the consistent consequences of the updates mentioned above are allowed to remain in the database as non-supported consequences.

Condition 1 above, ensures that complementary information is cancelled prior to the database transaction performance, in order to avoid redundant update execution.

Condition 2 puts forwards that an update U_i which violates directly an integrity constraint of the database cannot be performed, however, under our compromised approach, its consistent consequences can be kept in DB For instance, if $T = \{A, B, C\}$, $DB = \{\ \}$ and $I = \{C \rightarrow \perp\}$, the transaction would be reduced to $\{A, B\}$, since $\{C\} \bigcup I \vdash \perp$. Eliminating the update that violates the integrity constraint from the transaction, and allowing the other updates in T to be performed is, most of the times, an intuitive procedure which conforms with the compromised philosophy of our approach. That is of keeping in the database as much information as possible from conflicting updates. Consider the case that A expresses that worker $W1$ gets a raise of 10% on his salary; B expresses that worker $W2$ gets a raise of 30%; and C expresses that worker $W3$ gets a raise of 50% on his salary. Assume that their company has restricted raises of 50% or higher on workers' salaries. Then, update C would not be performed and would have to be negotiated later. However, this would not stop updates A and B from being performed.

Condition 3 caters for the case when two updates are individually consistent to be performed, but together they violate, already initially, the set of integrity constraints which rules the database. In the case of two conflicting updates within the same transaction, our approach allows for their consistent consequences to be kept in the database. However, if the resulting set of their consistent consequences is empty, then we have to face the problem of preference among the conflicting updates. To do this, we rely on meta-level information concerning the relevance (or the weight) of the updates involved in the conflict. Depending on this information, we can either decide for the extreme case in which no update is effectively performed on the original database, or we can have that at least one of the conflicting updates in the transaction is performed in full, provided that it does not violate the integrity constraints of the database. Take the example where a transaction is given by $T = \{A, B, C\}$, and the integrity constraint set of the database is given by $I = \{A \wedge B \rightarrow \perp\}$. Assume that A expresses that Paper-1 is submitted to a scientific conference; B expresses that Paper-1 is submitted to a scientific journal; and C expresses that Paper-2 is submitted to a conference. All papers are from the same author (Researcher-X). According to some academic constraints, the same paper cannot be sent to a conference and to a journal at the same time. So, a meta-level choice has to be made between A and B , depending on the updates' weights of transaction T. We can imagine different options for such a choice. For instance, the one based on the fact that a publication in a journal has more academic relevance than a publication in the proceedings of a conference, then B would be preferred over A. Alternatively, Researcher-X might

[11] This meta-level information is totally context-dependent. We could, for instance, have a total ordering among the updates in T, so that each U_i would be less preferable than U_{i+1}. We will not discuss details about this meta-level based choice in this paper.

[12] Most of the database-update approaches in the literature do not conform with this viewpoint, since they adopt a style denoted sometimes as *all-or-none* updates performance, in the case of inconsistency or integrity constraint violation within a transaction, e.g. [MBM-95].

be willing to take part in many conferences as he can this year, in order to establish some academic contacts, and in this case A would be preferred over B.

The remainder of this paper, introduces a compromised characterization to base revision, within the philosophy of our approach.

2 A Compromised Revision for Bases

We introduce here a compromised characterization to database revision. This characterization shows how our approach of reconciling conflicting inputs behaves under the perspective of belief revision. We propose a formal structure to our compromised revision approach, on the basis of the AGM model [AlMa-82] [AGM-85] [Gär-88]. However, we consider our theory representation as a set of sentences which is not closed under logical consequence. We refer to it as *base*. Compromised revision is presented for conventional bases and for bases driven by a protected subset of integrity constraints. Some postulates for compromised revision are established, and a revision function for bases with integrity constraints is specified.

It has already been quite recognized in the literature, that change operations on sets that are closed under logical consequence are ideal as mathematical exercises, but in real life those operations are always applied to bases for theories. The approach of using bases instead of closed theories in belief revision, has been introduced in the literature by [Neb89] and [Han-91], and since then has gained many followers. Here, we also consider a finite base as theory representation, due to our interest in simulating changes of the real world, which are always done locally within a theory representation. Most of the work on base belief revision, propose an alternative to the AGM model. This is not our intention here. Our main concern is to characterize a specific model which applies for the special case of having compromised solutions to conflicting base updates, within the philosophy of our approach, and not to propose another general belief revision model.

Base theory representations which are driven by a protected set of sentences are also considered in this characterization. Such protected part of the bases can be viewed as a set of integrity constraints - a notion from the database context. By having this notion of protected constraints which range over the remaining sentences of the base, we are able to account for theory changes in a more realistic way. This is so, because instead of always accomodating the new input simply because of its information novelty, we have the possibility of rejecting it, if it conflicts with the protected part of the base.[13]

In general, by *compromised revision* we mean that an input sentence α, which is inconsistent with a knowledge base K, will have its consistent consequences w.r.t. K, added to K. However, α can be either added to K, provided that the revised base is consistent, or rejected from being added to K. The addition or rejection of α depends on *how* it is inconsistent with K. In the case that the base K contains a protected part of integrity constraints, the input α is rejected if it is directly inconsistent with K. This means that α violates directly K's integrity constraints. And α is added to K if it is indirectly inconsistent with K. In the case that K is a conventional base set, α is only rejected from being added to K, if it contradicts a tautology of the logical system considered.

2.0.1 Considerations about Contraction in Compromised Revision

In order to revise a base K with a sentence α, it is often necessary to retract some old sentences from K. Those sentences which together with α generate inconsistency. In this case, when contracting a sentence β from K, we are not only interested in removing from K a minimal subset whose removal prevents what remains from entailing β. But we are also interested in keeping all the consistent consequences of β w.r.t. K available, provided that they also do not interfere in the revision process.

[13] In [Han-91], they also support this point by presenting a notion of belief change, which accounts for the issue of receiving new information without giving it special priority due to its novelty. They call it *non-prioritized reception of epistemic input.*

The idea behind this contraction method comes from the fact that we want to have a compromised solution for a conflicting input, in the sense that the minimal amount of information is lost. Hence, we compromise between what is being added to K and what is being retracted from it. In the case that the input sentence cannot be added to the knowledge base K, we compromise by allowing its consistent consequences to be added to K. Otherwise, if it can be added, conditioned to K being revised, we compromise by allowing the consistent consequences of the retracted sentences to be available in the revised knowledge base. By doing so, we believe that we have less loss of information from the knowledge base in the end.

Within this method of compromised contraction of bases, we have that the retracted sentences are only removed from the base partially, in their informative sense, since some of their logical consequences remain available in the knowledge base.

2.0.2 The Revision Steps

Below, we state the compromised revision steps for both conventional bases, and bases with protected integrity constraints, with relation to a particular input. We consider the bases to be initially consistent, i.e. they are consistent prior to any modification that they may suffer.

Revision Steps for Conventional Bases

- We add the new input to the base.
- If the base is inconsistent because the input contradicts a tautology of the logical system, then we make the base consistent by rejecting the input.
- If the base is inconsistent because the input generates inconsistency together with some old sentences of the base, then we make the base consistent by rejecting some old sentences of the base.
- For each sentence rejected, we allow its consequences to be added to the base, provided that they do not contradict or cause inconsistency due to the incorporation of the new input to the base.

Example 2.1 (Compromised Revision for Conventional Bases)
Consider a base K with the following elements

$$K = \{a,\ b,\ c,\ a\&d \to \bot,\ a\&c \to q\}.$$

If we want to update K with the input d, we end up with one of the following revised bases:

$$K' = \{b,\ c,\ a\&d \to \bot,\ a\&c \to q,\ q,\ d\},\ or\ K'' = \{a,\ b,\ c,\ a\&c \to q,\ d\}.$$

•

Revision Steps for Bases with Integrity Constraints

- We add the new input to the base.
- If the base is inconsistent because the input contradicts a tautology of the logical system, then we make the base consistent by rejecting the input.
- If the base is inconsistent because the input directly contradicts some of the sentences of its protected part, then we make the base consistent by rejecting the input, but allowing its consistent consequences to be added to the base.
- If the base is inconsistent because the input either contradicts an old sentence of the base (which is not protected), or contradicts together with some other sentences of the base, sentences of the protected part, then we make the base consistent by rejecting some old sentences of the base.

- For each sentence rejected, we allow its consequences to be added to the base, provided that they do not contradict or cause inconsistency due to the incorporation of the new input to the base.

Examples of Compromised Revision for Bases with Integrity Contraints

Example 2.2

Consider a base K *composed a protected subset of integrity contraints* P_K *and a non-protected subset of sentences* Δ_K*. Assume that* $P_K = \{a\&b\&c \rightarrow \bot,\ a\&q \rightarrow \bot\}$ *and* $\Delta_K = \{b,\ c,\ q,\ a\&c \rightarrow d\}$*. If we want to update* K *with the input* a*, we may end up with one of the following revised bases:*

$$K' = \{a,\ c,\ a\&c \rightarrow d\}, \quad or \quad K'' = \{a,\ b,\ a\&c \rightarrow d,\ d\}.$$

•

Example 2.3

Consider a base $K = P_K \bigcup \Delta_K$*, where* $P_K = \{a \rightarrow \bot\}$ *and* $\Delta_K = \{b,\ c,\ q,\ a\&c \rightarrow d\}$*. If we want to update* K *with the input* a*, we may end up with the following revised base:*

$$K' = \{b,\ c,\ a\&c \rightarrow d,\ d\}.$$

•

In terms of logical consequences, our approach is not so restrictive as the ones in conventional base revision, and not so permissive as closed-theory revision. This is due to the fact that we do not allow for all the consistent logical consequences of the base to be part of the revised set. However, we do allow the incorporation of some consequences of the sentences which were rejected from the base. As a result, our revised set loses less information than one obtained by a standard base-revision approach. But it is still not as big as the one obtained by the closed-theory approach. Even though a compromisingly revised set might have sentences which a revised set obtained by the closed-theory approach does not have.

Below, we informally present the logical setting, the notation and the basic assumptions considered throughout the proposed characterization.

2.0.3 Basic Assumptions & Definitions

We consider a finitary propositional language L. The underlying logic includes classical propositional logic, so that L is closed under applications of the usual boolean operators, namely $\neg$ (negation); $\wedge$ (conjunction); $\vee$ (disjunction); and $\rightarrow$ (implication).

The knowledge base, denoted by K, is initially defined as a finite set of sentences of L, that is not closed under logical consequence.

The derivability relation $\vdash$ is assumed to be compact, and to satisfy the following properties:

- If α is a classical tautology, then $\vdash \alpha$;
- $\vdash$ is consistent;
- $\vdash$ is closed under modus ponens; and
- $\vdash$ satisfies the deduction theorem. (If $\alpha \vdash \beta$ then $\vdash \alpha \rightarrow \beta$.)

$K \vdash \alpha$ denotes that the knowledge base K logically entails α, where α is a sentence of L.

We assume that $K \vdash \alpha$ if either $\alpha \in K$; or $\vdash \alpha$; or there exists a β, such that $\beta \rightarrow \alpha \in Cn(K)$ and $\beta \in Cn(K)$.

$Cn(K)$ denotes the set of logical consequences of K, such that $K \subseteq Cn(K)$, $Cn(K) = Cn(Cn(K))$, and $Cn(K_1) \subseteq Cn(K_2)$ whenever $K_1 \subseteq K_2$.

K is assumed to be initially consistent, i.e. $K \not\vdash \bot$.

In the usual way, we assume that $K \vdash \perp$ iff $K \vdash \alpha$ and $K \vdash \neg\alpha$ or $K \vdash \beta$ and $\beta \rightarrow \perp$.

$K_\perp$ denotes the inconsistent set, which contains the set of all sentences of L.

We use the symbol $+$ to denote the AGM expansion operator, such that $K+\alpha$ denotes the expansion of K by α and its logical consequences. $K + \alpha = Cn(K \bigcup \{\alpha\})$.

$K \circledast \alpha$ denotes the knowledge base which results from the base K revised by α. $\circledast$ is a function which takes a base set K and a sentence α to another base set $K \circledast \alpha$. The revision operation with $\circledast$ is a version of the AGM revision operation with $*$, (see appendix A), adpated for theory-bases.

The symbol " $-$ " used from here ownwards, unless differently stated, denotes difference in the set-theoretical sense.

2.0.4 The AGM Revision Postulates for Bases

In the postulates below, we assume that K is a theory-base, not closed under the consequence operation Cn, and α and β are sentences of the language L.

($K1$) $K \circledast \alpha$ is a base set.

($K2$) $\alpha \in K \circledast \alpha$.

($K3$) $K \circledast \alpha \subseteq K \bigcup \{\alpha\}$.

($K4$) If $K \nvdash \neg\alpha$ then $K \bigcup \{\alpha\} = K \circledast \alpha$.

($K5$) $K \circledast \alpha \vdash \perp$ if and only if $\vdash \neg\alpha$.

($K6$) If $\vdash \alpha \leftrightarrow \beta$ then $Cn(K \circledast \alpha) = Cn(K \circledast \beta)$.

($K7$) $K \circledast \alpha\&\beta \subseteq (K \circledast \alpha) + \beta$.

($K8$) If $K \circledast \alpha \nvdash \neg\beta$ then $(K \circledast \alpha) + \beta \subseteq Cn(K \circledast \alpha\&\beta)$.

Notes: The postulates above have similar intuitions of those defined for the AGM theory. By the postulate ($K1$), the resulting revised set is guaranteed to be a base set. For the case of closed-theories such a postulate made sense. However, for non-closed belief bases, it is not so relevant. By postulate ($K2$), we have that the sentence revised is also part of the resulting set. Postulate ($K3$) states that the revised set of beliefs is at least a subset of the expanded set, in the set-theoretical sense, when both operations use the same sentence. Postulate ($K4$) caters for the case in which no old belief has to be given up from the original base set. Postulates ($K3$) and ($K4$) differ from the AGM postulates (K^*3) and (K^*4), respectively, since in the latter ones they consider the expanded sets as the resulting sets from the AGM expansion. Postulate ($K5$) guarantees that the revised set is consistent, unless in the case that α is an inconsistent sentence. Postulate ($K6$) ensures the equality between the closure of the revisions of logically equivalent sentences, considering the same original set of beliefs. ($K1$)-($K6$) represents the basic set of postulates for base revision. The postulates ($K7$) and ($K8$) are extra conditions for "composite" belief revision, like the postulates (K^*7) and (K^*8), for the case of revision for closed theories.

2.0.5 Compromised Revision for Conventional Bases

Given an initial knowledge base K, as defined above, we assume that α is the input sentence to revise K. We denote as " $\bigotimes$ " the operator for compromised revision for conventional bases. We define the postulates ($\bigotimes$1)-($\bigotimes$9) as the basic requirements for achieving the new set $K \bigotimes \alpha$, which is the result from the compromised revision of the knowledge base K by α. So, $K \bigotimes \alpha$ is assumed to satisfy the postulates ($\bigotimes$1)-($\bigotimes$9).

In the postulates ($\bigotimes$1)-($\bigotimes$9) to follow, we consider the following assumptions:

- When a sentence β has to be retracted from K in order to accomplish the incorporation of α in K Ⓡ α, we say that β is rejected w.r.t. α. We denote by R_α the set of all sentences $\beta \in K$, which are rejected w.r.t. α in order to form K Ⓡ α.

- We assume that $CR(R_\alpha)$ is the largest set of consequences of all the sentences $\beta_i \in R_\alpha$, for $1 \leq i \leq n$, w.r.t. K, such that $CR(R_\alpha)$ excludes the set R_α and satisfies the following conditions: $\forall y \in CR(R_\alpha)$, $K \vdash y$ and $(K - R_\alpha) \nvdash y$; and $\forall \beta \in R_\alpha$, $CR(R_\alpha) \bigcup (K - R_\alpha) \nvdash \beta$.
- $MaxCR(R_\alpha)$ is a maximal consistent subset of $CR(R_\alpha)$ w.r.t. $(K - R_\alpha) + \alpha$, such that $MaxCR(R_\alpha) \bigcup (K - R_\alpha) \bigcup \{\alpha\} \nvdash \bot$.
- We assume that D_{CR} is a set which denotes the difference $D_{CR} = CR(R_\alpha) - MaxCR(R_\alpha)$.

Proposed Postulates

($\bigotimes$1) $K \bigotimes \alpha$ is a base set.

($\bigotimes$2) $\alpha \in K \bigotimes \alpha$.

($\bigotimes$3) If $\alpha \in K$ then $K \bigotimes \alpha = K$.

($\bigotimes$4) If $K + \alpha \nvdash \bot$ then $K \circledast \alpha \subseteq K \bigotimes \alpha$.

($\bigotimes$5) If $K + \alpha \vdash \bot$ then $K \bigotimes \alpha \subseteq (K - R_\alpha) \bigcup \{\alpha\} \bigcup MaxCR(R_\alpha)$.

($\bigotimes$6) If $\vdash \neg\alpha$, then $K \bigotimes \alpha = K$.

($\bigotimes$7) For all knowledge bases K and for all sentences α, $K \bigotimes \alpha \nvdash \bot$.

($\bigotimes$8) If $\vdash \alpha \leftrightarrow \beta$ then $Cn(K \bigotimes \alpha) = Cn(K \bigotimes \beta)$.

($\bigotimes$9) For all knowledge bases K and for all sentences α, $K \bigotimes \alpha$ is the largest set that satisfies ($\bigotimes$1)-($\bigotimes$8), such that the set D_{CR} is minimal w.r.t. $K \bigotimes \alpha$. Hence, $\forall\, x \in D_{CR}$, $(K \bigotimes \alpha) + x \vdash \bot$.

Notes:

Postulates ($\bigotimes$1) and ($\bigotimes$2) are equivalent to the postulates ($K1$) and ($K2$), for AGM base revision.
The postulate ($\bigotimes$3) states the vacuity of the compromised revision.
Postulate ($\bigotimes$4) states that compromised revision includes the AGM base-revision, in the case that the sentence α does not contradict any sentences in K. This postulate is also equivalent to postulate ($K4$).
Postulate ($\bigotimes$5) caters for the case in which the revision-input sentence contradicts some sentences of the knowledge base K. ($\bigotimes$5) guarantees that the input sentence is in the resulting knowledge base set, together with all the consistent consequences of the retracted sentences w.r.t. K, if any.
Postulates ($\bigotimes$6) and ($\bigotimes$7) cater for consistency in $K \bigotimes \alpha$. ($\bigotimes$6) guarantees that if the input sentence contradicts a tautology of the logical system, then no change is made on the original set K. And postulate ($\bigotimes$7) claims that the resulting compromisingly-revised base is always consistent.
Postulate ($\bigotimes$8) is equivalent to postulate ($K7$). It guarantees the equality between the closure of the compromised revisions of logically equivalent sentences, considering the same original sets.
Finally, postulate ($\bigotimes$9) introduces the *maximality* condition for $K \bigotimes \alpha$. It ensures that compromised-revisions satisfy our notion of "compromised reasoning", by getting as many as possible of the from the retracted sentences in order to accomplish the revision.

($\bigotimes$1)-($\bigotimes$9) represent the basic set of postulates for compromised-revisions for conventional bases. These postulates should guide the construction of a compromised-revision function for such bases.

Discussions

When comparing the set of postulates for compromised-revisions for conventional bases with the set of AGM revision postulates for bases, we notice the following:

As expected, the postulate $(K3)$ is not satisfied in compromised revision, since we might add to $K \bigotimes \alpha$ some consequences of retracted sentences which do not belong to $K \bigcup \{\alpha\}$.

Compromised revision is more strict in terms of consistency than AGM revision. While AGM revision allows the revised base to become inconsistent in the case that the input sentence contradicts a logical tautology, the compromised revision constrains this possibility. This is shown in postulates $(K5)$ and $(\bigotimes 7)$, respectively.

Postulate $(K7)$ is not satisfied in compromised revision, since we might add to $K \bigotimes \alpha\&\beta$ some consequences of retracted sentences which do not belong to $(K \bigotimes \{\alpha\}) + \beta$. For the same reason, $(K8)$ is also not satisfied in compromised revision. We cannot guarantee that the consequences of the deleted old sentences of K added to $K \bigotimes \alpha$ are present in $Cn(K \bigotimes \alpha\&\beta)$.

In the sequel of this paper, we deal with theory representations which are bases containing protected integrity constraints.

2.0.6 Compromised Revision for Bases with Integrity Contraints

The knowledge base for compromised revision of bases with integrity constraints, is initially defined as the set $K = \Delta_K \bigcup P_K$, in which Δ_K and P_K are assumed to be finite sets of sentences of L, that are not closed under logical consequence. P_K is a protected part of K. Therefore, its formulae cannot be modified by any update operation in K. P_K represents the integrity constraints ranging over Δ_K.

Given an initial knowledge base K, where $K = \Delta_K \bigcup P_K$, and a sentence α, we assume that α is the input sentence to revise K. We denote as " ® " the operator for compromised revision of bases with integrity constraints. We define the postulates (®1)-(®9) as the basic requirements for achieving the new set K ® α, which is the result from the compromised revision of the knowledge base K by α. So, K ® α is assumed to satisfy the postulates (®1)-(®9).

In the postulates (®1)-(®9) to follow, we consider some assumptions, given below. Some of them were already defined for compromised revision of conventional bases.

- $CI(\alpha)$ is the largest set of consequences of α w.r.t. Δ_K, such that the following conditions are satisfied: $\alpha \notin CI(\alpha)$;
  $CI(\alpha) \bigcup \Delta_K \nvdash \perp$ and $CI(\alpha) \bigcup \Delta_K \nvdash \alpha$.
- $MaxCI(\alpha)$ is a maximal consistent subset of $CI(\alpha)$ w.r.t. K, such that $MaxCI(\alpha) \bigcup K \nvdash \perp$.
- When a sentence β has to be retracted from K in order to accomplish the incorporation of α in K ® α, we say that β is rejected w.r.t. α. We denote by R_α the set of all sentences $\beta \in K$, which are rejected w.r.t. α in order to form K ® α.
- We assume that $CR(R_\alpha)$ is the largest set of consequences of all the sentences in R_α, w.r.t. Δ_K, such that the following conditions are satisfied: $R_\alpha \nsubseteq CR(R_\alpha)$; $\forall y \in CR(R_\alpha)$, $\Delta_K \vdash y$ and $(\Delta_K - R_\alpha) \nvdash y$; and $\forall \beta \in R_\alpha$, $CR(R_\alpha) \bigcup (\Delta_K - R_\alpha) \nvdash \beta$.
- $MaxCR(R_\alpha)$ is a maximal consistent subset of $CR(R_\alpha)$, such that $MaxCR(R_\alpha) \bigcup (\Delta_K - R_\alpha) \bigcup P_K \bigcup \{\alpha\} \nvdash \perp$.
- We assume that D_{CR} is a set which denotes the difference
  $D_{CR} = CR(R_\alpha) - MaxCR(R_\alpha)$.
  And D_{CI} is a set which denotes the difference
  $D_{CI} = CI(\alpha) - MaxCI(\alpha)$.

Proposed Postulates for Compromised Revision ®

(®1) K ® α is a structure of the same type as K.

($\circledR$2) If $\alpha \in \Delta_K$ or if $\vdash \neg\alpha$, then $K \circledR \alpha = K$.

($\circledR$3) If $K + \alpha \nvdash \bot$ then $K \circledast \alpha \subseteq K \circledR \alpha$.

($\circledR$4) If $P_K + \alpha \nvdash \bot$ and $K + \alpha \vdash \bot$ then $\alpha \in K \circledR \alpha$, and $\Delta_{K\circledR\alpha} \subseteq (\Delta_K - R_\alpha) \bigcup \{\alpha\} \bigcup MaxCR(R_\alpha)$.

($\circledR$5) If $P_K + \alpha \vdash \bot$ and it is not the case that $\vdash \neg\alpha$, then $K \circledR \alpha \nvdash \alpha$ and $\Delta_{K\circledR\alpha} \subseteq \Delta_K \bigcup MaxCI(\alpha)$.

($\circledR$6) $P_K \subset K \circledR \alpha$.

($\circledR$7) For all K and for all α, $K \circledR \alpha \nvdash \bot$.

($\circledR$8) If $\vdash \alpha \leftrightarrow \beta$ then $Cn(K \circledR \alpha) = Cn(K \circledR \beta)$.

($\circledR$9) For all knowledge bases K and for all sentences α, $K \circledR \alpha$ is the largest set that satisfies ($\circledR$1)-($\circledR$8), such that the sets D_{CR} and D_{CI} are minimal w.r.t. $K \circledR \alpha$. Hence, $\forall\, x \in D_{CR}$, $(K \circledR \alpha) + x \vdash \bot$, and $\forall\, y \in D_{CI}$, $(K \circledR \alpha) + y \vdash \bot$.

Notes:

Postulate ($\circledR$1) states that the resulting revised knowledge base has the same structure of the original base.

Postulate ($\circledR$2) states the *vacuity* of the compromised revision, in the case that the input sentence to revise K is already an element of the original base K. Furthermore, in the case that α contradicts a tautology of the system, i.e. if $\vdash \neg\alpha$, no change at all is applied to K.

By postulate ($\circledR$3), the compromised revision includes the AGM base-revision, in the case that the sentence α does not contradict any sentences in K. A simpler alternative for this postulate would be the following: If $K + \alpha \nvdash \bot$ then $K \circledR \alpha = K \bigcup \{\alpha\}$. ($\circledR$3) is the *inclusion* postulate of our compromised revision.

Postulate ($\circledR$4) caters for the case in which the revision-input sentence does not contradict the set of integrity constraints, however it contradicts some sentences of the knowledge base K. ($\circledR$4) guarantees that the sentence which revised the base is in the resulting knowledge base set, together with the set of integrity constraints. Also, all the consistent consequences of the retracted sentences w.r.t. K, if any, are added to the revised set. We consider ($\circledR$4) as the *success* postulate.

Postulate ($\circledR$5) states that if the input sentence to revise K violates some of its integrity constraints, and does not contradict any tautology of the logical system, then the resulting compromisingly-revised set does not entail the revision-input sentence. However, it entails all its logical consequences, which do not generate inconsistency w.r.t. the original knowledge base. ($\circledR$5) guarantees that the resulting revised set includes the original set K, and all the consistent consequences of α w.r.t. Δ_K, if any. We consider ($\circledR$5) as the *compromise* postulate.

($\circledR$6) is the *integrity preservation* postulate. It guarantees that all the protected sentences in P_K which are part of the original knowledge base, remain present in the resulting compromisingly-revised set.[14]

Postulate ($\circledR$7) is the *consistency* postulate. It claims that the resulting compromisingly-revised knowledge base is always consistent.

Postulate ($\circledR$8) guarantees the equality between the closure of the compromised revisions of logically equivalent sentences, considering the same original sets.

Finally, postulate ($\circledR$9) ensures that compromised-revisions satisfy our notion of "compromised reasoning", by getting as many as possible of the consequences of the revision-input sentence, as well as the consequences from the sentences retracted in order to accomplish the revision. ($\circledR$9) introduces the *maximality* conditions for $K \circledR \alpha$.

($\circledR$1)-($\circledR$9) represent the basic set of postulates for compromised-revisions of bases with integrity constraints. These postulates express the compromised nature of our method's revision process, and they should guide the construction of a compromised-revision function for such bases.

[14] By abuse of notation, we have represented as $P_K \subset K \circledR \alpha$, the meaning that $K \circledR \alpha = \langle \Delta_{K\circledR\alpha},\ P_K \rangle$.

2.0.7 On Formalizing the Compromised Revision

The formulation of a compromised revision function for bases with integrity constraints, can be defined based on the postulates (Ⓡ1)-(Ⓡ9). Since those postulates present our intuitions, providing already specific conditions which should be satisfied, the revision function can be constructed explicitly from them in a very straightforward way. However, there are some notions which still remain to be defined.

The notion of how to select the elements which are rejected w.r.t. α in order to accomplish the revision K Ⓡ α, requires a proper definition. This notion requires that we define an ordering on K, or better saying on Δ_K. Intuitively, we want to be able to know which elements of K conflict with α, so that we can build up a set containing these elements, and then get the set R_α from it. The ordering on Δ_K helps to determine the elements of R_α. Consequently, R_α is effectively the set of sentences to be retracted from K in order to accomplish the revision K Ⓡ α.

In the case of a compromised revision K Ⓡ α for bases with integrity constraints, concerning the sentences that we might want to eliminate from among the consequences of K, we do not only have to consider $\neg\alpha$, but also all the other elements of Δ_K that together with α violate some integrity constraints in P_K . Based on this idea, we define here a set denoted $\perp_\alpha$, which includes all the minimal subsets of Δ_K, with those elements.[15]

Moreover, we want to be able to perform the necessary retractions, satisfying the conservativity principal. We then define the set R_α, taking into account the minimals of the subsets of $\perp_\alpha$, w.r.t. the ordering of Δ_K.[16] R_α selects from the sets of $\perp_\alpha$ the minimal elements which should be retracted from Δ_K.

The notion of an ordering does not apply to both subsets of K, since P_K is protected and we are not interested in ordering it for any modification. Hence, we assume a partial order $\leqslant$ on the non-protected elements of K, on Δ_K. We also restrict the types of sentences in P_K, in order to comply with our formalization requirements. This implies that we change the initial definition of the knowledge base K from the set $K = \Delta_K \bigcup P_K$, to the structure $K = \langle \Delta_K, P_K \rangle$.

Preliminary Definitions

Definition 2.1 ($K = \langle \Delta_K, P_K \rangle$)

Given a language L of propositional sentences, let K be a knowledge base given by the structure $K = \langle \Delta_K, P_K \rangle$, such that P_K is a protected set of sentences of L of the type $\bigwedge_{i=1}^{i=n} \alpha_i \rightarrow \perp$, where α_i is a proposition or its negation, and Δ_K is a partially ordered set of sentences of L, w.r.t. $\leqslant$.

□

Remark 2.1

When we state the union of $K \bigcup \{\alpha\}$, for instance, we implicitly mean that α is added to Δ_K, such that $K \bigcup \{\alpha\} = \langle \Delta_K \bigcup \{\alpha\}, P_K \rangle$. Concerning the ordering, whenever not explicitly stated, α receives the highest priority in Δ_K. also, for the sake of notation simplicity, we sometimes say that $\alpha \in K$ whenever $\alpha \in \Delta_K$ or $\alpha \in P_K$.

•

The ordering $\leqslant$ on Δ_K is supposed to give the intuitive meaning of relevance to the elements of Δ_K, according to the requirements of the application area to which the system is applied. Since $\leqslant$ is a partial order, it allows a high level of expressivity on the knowledge base representation. The user has the option to define the relevance order among the elements of the set Δ_K, considering also that some elements might not be related by $\leqslant$. This means that we allow for elements with the same relevance level. If the application requires, Δ_K can also be defined as a chain, i.e. as a totally ordered set, simplifying quite a lot the revision process. But the system does not oblige the user to do that as a premise.

[15] See Definition 2.8.
[16] See Definitions 2.1 and 2.12.

When we say that $x, y \in \Delta_K$, we mean that x and y are sentences of L ordered by $\leqslant$ in Δ_K. We say that a sentence x has at least the same relevance of another sentence y, whenever $y \leqslant x$, for $x, y \in \Delta_K$. As usual, $y < x$ means $y \leqslant x$ and $x \nleqslant y$. Also, the notations $y \leqslant x$ and $y < x$, are equivalent to $x \geqslant y$ and $x > y$, respectively. It is important to notice that when we add an input sentence α to K, we mean that α is added to Δ_K with highest priority w.r.t. $\leqslant$. ($\forall x \in \Delta_K,\ x < \alpha$).

As $\leqslant$ is a partial order, it presents the properties of reflexivity, transitivity and antisymmetry. So, for any sentences α, β, γ in Δ_K, $\alpha \leqslant \alpha$; if $\alpha \leqslant \beta$ and $\beta \leqslant \gamma$, then $\alpha \leqslant \gamma$; and if $\alpha \leqslant \beta$ and $\beta \leqslant \alpha$, this implies that $\alpha = \beta$.

The definition below caters for the notion of propagation of the ordering $\leqslant$ of the elements of an arbitrary set X, to their consequences. This ordering propagation is necessary for the case that we have to discard some of the compromised consequences, in order to avoid inconsistency.

Definition 2.2 (Propagation of $\leqslant$)

Given a consistent set of propositional sentences X, ordered under the partial order $\leqslant$, assume that $Con(X)$ is the set of consequences of X w.r.t. $\vdash$. Let the ordering on the elements of $Con(X)$ be such that:

$$\forall y \in Con(X),\ \textit{if}\ Z \vdash y,\ \forall Z \subseteq X,\ \textit{and}\ y \notin X,\ \textit{then}\ x \leqslant y,\ \forall x \in Z.$$

□

Below, we state some set definitions, concerning the minimal elements of a partially ordered set.

Definition 2.3 ($min(\mathcal{Y})$)

Given a set $\mathcal{Y}$ such that $\mathcal{Y} = \{S_1, S_2, \cdots, S_n\}$, where each S_i, for $i = 1, \cdots, n$, is ordered w.r.t. the partial order $\leqslant$. Let $min(\mathcal{Y})$ be the following set:

$$min(\mathcal{Y}) = \begin{cases} \emptyset & \textit{if}\ \mathcal{Y} = \emptyset; \\ \{min(S_i) \parallel min(S_i) \subseteq S_i\,; \\ \quad \textit{and}\ \forall x \in min(S_i),\ x\ \textit{is minimal w.r.t}\ \leqslant\ \textit{in}\ S_i\} & \textit{otherwise.} \end{cases}$$

□

Definition 2.4 ($min(\mathcal{Y})^+$)

Given a set $\mathcal{Y}$ such that $\mathcal{Y} = \{S_1, S_2, \cdots, S_n\}$, where each S_i, for $i = 1, \cdots, n$, is ordered w.r.t. the partial order $\leqslant$. Assume that $min(\mathcal{Y})$, is such that $min(\mathcal{Y}) = \{min(S_1), min(S_2), \cdots, min(S_n)\}$. Let $min(\mathcal{Y})^+$ be the following set:

$$min(\mathcal{Y})^+ = \{min(S_i)_j \parallel min(S_i)_j \in min(\mathcal{Y}),\ \textit{and}\ min(S_i)_j\ \textit{has more than one element}\}.$$

Where $1 \le i \le n$, and $j = 1, \cdots, k$, for $k \le n$.

□

Definition 2.5 ($min(\mathcal{Y})^1$)

Given a set $\mathcal{Y}$ such that $\mathcal{Y} = \{S_1, S_2, \cdots, S_n\}$, where each S_i, for $i = 1, \cdots, n$, is ordered w.r.t. the partial order $\leqslant$. Assume that $min(\mathcal{Y})$, is such that $min(\mathcal{Y}) = \{min(S_1), min(S_2), \cdots, min(S_n)\}$. Let $min(\mathcal{Y})^1$ be the following set:

$$min(\mathcal{Y})^1 = \{x \parallel x \in min(S_i)_j,\ min(S_i)_j \in min(\mathcal{Y}),\ \textit{and}\ min(S_i)_j\ \textit{has only one element}\}.$$

Where $1 \le i \le n$, and $j = 1, \cdots, l$, for $l \le n$.

□

Definition 2.6 ($Min(\mathcal{Y})$)

Given a set $\mathcal{Y}$ *such that* $\mathcal{Y} = \{S_1, S_2, \cdots, S_n\}$*, where each* S_i*, for* $i = 1, \cdots, n$*, is ordered w.r.t. the partial order* $\leqslant$*. Assume that* $min(\mathcal{Y})$*, is such that* $min(\mathcal{Y}) = \{min(S_1), min(S_2), \cdots, min(S_n)\}$*. Let* $Min(\mathcal{Y})$ *be the following set:*

$$Min(\mathcal{Y}) = \{x \parallel x \in min(S_i),\ \forall min(S_i) \in min(\mathcal{Y})\}.$$

Where $1 \leq i \leq n$*.*

□

We define now the notion of a *safe maximal* subset of an arbitrary partially ordered set X, with relation to a condition c which involves X, somehow, in its premisse. We call such a set $Smax(X)_c$. Basically, $Smax(X)_c$ is such that, if it substitutes X in the condition c, c succeeds. In the general case, a maximal subset of X w.r.t. the same condition c, denoted as $Max(X)_c$, has $Smax(X)_c$ as a subset. $Smax(X)_c = Max(X)_c$ only when $Smax(X)_c$ is maximal w.r.t. $\subseteq$, conditioned to c. That is, for any element $y \in (X - Smax(X)_c)$, if $Smax(X)_c \bigcup \{y\}$ substitutes X in the condition c, c fails.

The set $Smax(X)_c$ is obtained considering the ordering $\leqslant$, the set-inclusion property of minimality, and an auxiliary set called $Fail(X)_c$, as described below:

- First we get the set $Fail(X)_c$, which contains all the minimal subsets of X, such that when they substitute X in the condition c, c fails. $Fail(X)_c = \{S_1,\ S_2, \cdots, S_n\}$.
- Then we get the set $min(Fail(X)_c)$, as in Definition 2.3, which contains all the $min(S_i)$, of minimal elements of each $S_i \in Fail(X)_c$, for $i = 1, \cdots, n$. From $min(Fail(X)_c)$, we also get $Min(Fail(X)_c)$ as stated in Definition 2.6.
- Finally, we define $Smax(X)_c$ as the set X if condition c is satisfied; and as $X - Min(Fail(X)_c)$ otherwise.

Below, we state the Definitions 2.7 and 2.8, concerning the sets $Fail(X)_c$ and $Smax(X)_c$, respectively.

Notation: We denote as $c(X/S)$, the fact that the set S substitutes the set X in condition c.

Definition 2.7 ($Fail(X)_c$)

Given a partially ordered set X*, w.r.t.* $\leqslant$*, and a condition* c *which involves* X *in its premisse, let* $Fail(X)_c$ *be the following set:*

$$Fail(X)_c = \begin{cases} \emptyset & \text{if } c \text{ is satisfied;} \\ \\ \{S \parallel S \subseteq X; & \\ \text{such that if } c(X/S),\ c \text{ fails;} & \\ \text{and } S \text{ is minimal w.r.t. } \subseteq\} & \text{otherwise.} \end{cases}$$

Each subset S_i *of* $Fail(X)_c$*, for* $i = 1, \cdots, n$*, is ordered by* $\leqslant$ *as a subset of* X*.*

□

Definition 2.8 ($Smax(X)_c$)

Given a partially ordered set X*, w.r.t.* $\leqslant$*, a condition* c *which involves* X *in its premisse, and the set* $Min(Fail(X)_c)$*, let* $Smax(X)_c$ *be a subset of* X*, obtained as follows:*

$$Smax(X)_c = \begin{cases} X & \text{if } c \text{ is satisfied;} \\ \\ X - Min(Fail(X)_c) & \text{otherwise.} \end{cases}$$

The set $Smax(X)_c$ *is ordered by* $\leqslant$ *as a subset of* X*.*

□

Remark 2.2

The set $Smax(X)_c$ is classified as a safe-maximal consistent subset of X, under condition c. In the case that all the sets in $min(Fail(X)_c)$ are unitary, $Smax(X)_c$ is also a maximal consistent subset of X w.r.t. set-inclusion. We call $Smax(X)_c$ safe-maximal, because it does not choose one element among the minimals, w.r.t. $\leqslant$, of the subsets of $Fail(X)_c$. Instead, it discards all the minimal elements that fail to accomplish the condition given. Adopting the safe-maximal *subset solution, instead of a maximal, leads us to a more impartial position when we have to choose among some compromised consequences to discard.*

•

Definitions for the case that $P_K + \alpha \vdash \bot$

For this case, we have to define the set of consequences of the input α w.r.t. Δ_K, which are consistent with K, that will be inserted to Δ_K as a revision compromise.

We will denote as $CI(\alpha)$, the set of all consequences of α w.r.t. Δ_K. Our goal is to define $CI(\alpha)$ as a finite set, such that $\alpha \notin CI(\alpha)$, and $CI(\alpha) \bigcup \Delta_K \bigcup P_K$ is consistent and do not derive α.

Remark 2.3

Some points have to be carefully dealt with, when defining $CI(\alpha)$:

1. $\Delta_K \bigcup\{\alpha\}$ *can also be inconsistent when* $P_K \bigcup\{\alpha\} \vdash \bot$, *since the set of postulates (®1)-(®9) does not restrict this case.*
2. *In the case that the condition $CI(\alpha) \bigcup \Delta_K \bigcup P_K \nvdash \bot$ is not satisfied, we have to retract from $CI(\alpha)$ the minimal elements w.r.t. $\leqslant$, which contribute for the failure of the condition above. If a maximal subset of $CI(\alpha)$ is considered in this case, it might not be necessarily unique.*
3. *In the case that the condition $CI(\alpha) \bigcup \Delta_K \nvdash \alpha$ is not fulfilled, we have to retract from $CI(\alpha)$ the minimal elements w.r.t. $\leqslant$, which make the condition above fail. Again, if a maximal subset of $CI(\alpha)$ is considered in this case, it might not be unique.*

•

Taking into account the problems of the remark above, we proceed by adopting the *safe maximal* subset notion, defined previously, instead of the maximal subset notion, in the following way:

- First we get a safe-maximal subset of Δ_K, w.r.t. condition c_1, where $c_1 = \Delta_K \bigcup\{\alpha\} \nvdash \bot$. We use the auxiliary sets $Fail(\Delta_K)_{c_1}$ and $Min(Fail(\Delta_K)_{c_1})$ to create $Smax(\Delta_K)_{c_1}$, as stated in Definitions 2.7, 2.6, and 2.8. We then prove that $Smax(\Delta_K)_{c_1} \bigcup\{\alpha\} \nvdash \bot$.[17]
- Then we define $CI(\alpha)$ is relation to $Smax(\Delta_K)_{c_1}$. This step allows us to eliminate the possibility of dealing with an inconsistent set, in the case that $\Delta_K \bigcup\{\alpha\} \nvdash \bot$.[18]
- From $CI(\alpha)$, we get a safe-maximal subset, w.r.t. condition c_2, where $c_2 = CI(\alpha) \bigcup \Delta_K \bigcup P_K \nvdash \bot$. The auxiliary sets $Fail(CI(\alpha))_{c_2}$ and $Min(Fail(CI(\alpha))_{c_2})$ are used to create $Smax(CI(\alpha))_{c_2}$. And we prove that $Smax(CI(\alpha))_{c_2} \bigcup \Delta_K \bigcup P_K \nvdash \bot$.[19]
- Finally, we get a safe-maximal subset of $Smax(CI(\alpha))_{c_2}$, w.r.t. condition c_3, where $c_3 = Smax(CI(\alpha))_{c_2} \bigcup \Delta_K \nvdash \alpha$. The set $Smax(Smax(CI(\alpha))_{c_2})_{c_3}$ is obtained using the auxiliary sets $Fail(Smax(CI(\alpha))_{c_2})_{c_3}$ and $Min(Fail(Smax(CI(\alpha))_{c_2})_{c_3})$. And it is proved that $Smax(Smax(CI(\alpha))_{c_2})_{c_2} \bigcup \Delta_K \nvdash \alpha$.[20]

We state below the remarks and definitions, for the case that $P_K \bigcup\{\alpha\} \vdash \bot$. In all of them, we assume that Δ_K is ordered by $\leqslant$, in the knowledge base $K = \langle \Delta_K,\ P_K \rangle$.

[17] See Remark 2.5; and Propositions 2.1 and 2.2.
[18] See Definition 2.9.
[19] See Remark 2.5; and Proposition 2.2.
[20] See Remark 2.7, as well as Proposition 2.4.

So, $Smax(Smax(CI(\alpha))_{c_2})_{c_3}$ is the safe-maximal version of the set $CI(\alpha)$, such that the problems of Remark 2.3 are avoided.

Remark 2.4 ($Fail(\Delta_K)_{c_1}$)
Given a knowledge base $K = \langle \Delta_K, P_K \rangle$ *and an input* α, *such that condition* $c_1 = \Delta_K \bigcup \{\alpha\} \vdash \perp$, *by Definition 2.7, we have that* $Fail(\Delta_K)_{c_1}$ *is the following set:*

$$Fail(\Delta_K)_{c_1} = \begin{cases} \emptyset & \textit{if } c_1 \textit{ is satisfied;} \\ \\ \{S \parallel S \subseteq \Delta_K\,; \\ \textit{such that } S \cup \{\alpha\} \vdash \perp, \\ \textit{and } S \textit{ is minimal w.r.t. } \subseteq\} & \textit{otherwise.} \end{cases}$$

Each subset S_i *of* $Fail(\Delta_K)_{c_1}$, *for* $i = 1, \cdots, l$, *is ordered by* $\leqslant$ *as a subset of* Δ_K.

•

Proposition 2.1 [21]
Given a base $K = \langle \Delta_K, P_K \rangle$, *and an input sentence* α, *such that* $\Delta_K \bigcup \{\alpha\} \vdash \perp$, *and it is not the case that* $\vdash \neg\alpha$, *then, given the set* $Fail(\Delta_K)_{c_1} = \{S_1, S_2, \cdots, S_j\}$, *it is sufficient to retract one element from each* S_i, *such that* $(\Delta_K - \mathcal{S}(Fail(\Delta_K)_{c_1})) \bigcup \{\alpha\} \not\vdash \perp$, *where* $\mathcal{S}(Fail(\Delta_K)_{c_1}) = \{x_i \parallel \forall S_i \in Fail(\Delta_K)_{c_1}, \exists y \in S_i, \textit{such that } x_i = y$.

Remark 2.5 ($Smax(\Delta_K)_{c_1}$)
Given a knowledge base $K = \langle \Delta_K, P_K \rangle$ *and an input* α, *such that condition* $c_1 = \Delta_K \bigcup \{\alpha\} \vdash \perp$, *by Definition 2.8, we have that* $Smax(\Delta_K)_{c_1}$ *is the following set:*

$$Smax(\Delta_K)_{c_1} = \begin{cases} \Delta_K & \textit{if } c_1 \textit{ is satisfied;} \\ \\ \Delta_K - Min(Fail(\Delta_K)_{c_1}) & \textit{otherwise.} \end{cases}$$

$Smax(\Delta_K)_{c_1}$ *is ordered by* $\leqslant$ *as a subset of* Δ_K.

•

Proposition 2.2
Given a base $K = \langle \Delta_K, P_K \rangle$, *for any input sentence* α, *such that* $\Delta_K \bigcup \{\alpha\} \vdash \perp$, *and it is not the case that* $\vdash \neg\alpha$, $Smax(\Delta_K)_{c_1} + \alpha \not\vdash \perp$.

Now we define $CI(\alpha)$ w.r.t. $Smax(\Delta_K)_{c_1}$.

Definition 2.9 ($CI(\alpha)$)
Given a knowledge base $K = \langle \Delta_K, P_K \rangle$ *and an input sentence* α, *such that it is not the case that* $\vdash \neg\alpha$, *let* $CI(\alpha)$ *be the set of consequences of* α, *w.r.t.* Δ_K, *such that:*

$$\begin{aligned} CI(\alpha) = \; & \{y \parallel Smax(\Delta_K)_{c_1} \not\vdash y \;\textit{ and }\; Smax(\Delta_K)_{c_1} \cup \{\alpha\} \vdash y; \\ & y \neq \alpha, \textit{ and } Smax(\Delta_K)_{c_1} \cup \{y\} \not\vdash \alpha\}, \end{aligned}$$

where $CI(\alpha)$ *is finite, maximal w.r.t.* $\subseteq$, *and is ordered by* $\leqslant$, *according to Definition 2.2.*

□

Example 2.4
Consider a base $K = \langle \Delta_K, P_K \rangle$, *where*
$\Delta_K = \{b, c, \neg a, a \rightarrow p, p \wedge b \rightarrow q, p \wedge q \rightarrow \neg c\}$,
for $b \leqslant c$, $a \rightarrow p \leqslant \neg a$, $b \leqslant \neg a$, $a \rightarrow p \leqslant p \wedge b \rightarrow q$, *and* $p \wedge b \rightarrow q \leqslant p \wedge q \rightarrow \neg c$.
Assume that $P_K = \{a \rightarrow \perp, e \wedge b \rightarrow \perp\}$. *If we consider the update input* a, *by definition we have that* $P_K + a \vdash \perp$, *then:* $Fail(\Delta_K)_{c_1} = \{\{\neg a\}\}$,
$Min(Fail(\Delta_K)_{c_1}) = \{\neg a\}$,
$Smax(\Delta_K)_{c_1} = \{b, c, a \rightarrow p, p \wedge b \rightarrow q, p \wedge q \rightarrow \neg c\}$, *and*
$CI(\alpha) = \{p, q\}$.

[21] The omitted proofs of the propositions and theorems cited in this paper, are found in [Dar-96].

•

The motivation behind obtaining $CI(\alpha)$, is that we want to be able to include in Δ_K all the consistent consequences of α, such that $CI(\alpha) \bigcup \Delta_K \bigcup P_K$ is consistent. Moreover, we do not want to have α available, (neither present, nor derivable), from the resulting base. However, Definition 2.9, as it stands does not satisfy these conditions. Notice that it is not guaranteed that $CI(\alpha) \bigcup \Delta_K \nvdash \alpha$, nor that $CI(\alpha) \bigcup \Delta_K \bigcup P_K \nvdash \perp$. In Example 2.4, for instance, that last condition is not satisfied. Hence, we still need to specify a subset of $CI(\alpha)$, preferrably maximal w.r.t. set-inclusion, that satisfies those conditions. Such subset of $CI(\alpha)$, is then obtained as a safe-maximal subset, as stated below.

Remark 2.6 ($Smax(CI(\alpha))_{c_2}$)
Given a knowledge base $K = \langle \Delta_K,\ P_K \rangle$ *and an input* α*, such that* $CI(\alpha)$ *is non-empty and condition* $c_2 = CI(\alpha) \bigcup \Delta_K \bigcup P_K \nvdash \perp$*, by Definition 2.8, we have that* $Smax(CI(\alpha))_{c_2}$ *is the following set:*

$$Smax(CI(\alpha))_{c_2} = \begin{cases} CI(\alpha) & \text{if } c_2 \text{ is satisfied;} \\ CI(\alpha) - Min(Fail(CI(\alpha))_{c_2}) & \text{otherwise.} \end{cases}$$

$Smax(CI(\alpha))_{c_2}$ *is ordered by* $\leqslant$ *as a subset of* $CI(\alpha)$*.*

•

Proposition 2.3
Given a base $K = \langle \Delta_K,\ P_K \rangle$*, for any input sentence* α*, such that* $CI(\alpha)$ *is non-empty,* $\Delta_K + \alpha \vdash \perp$*, and it is not the case that* $\vdash \neg\alpha$*,* $Smax(CI(\alpha))_{c_2} \bigcup \Delta_K \bigcup P_K \nvdash \perp$*.*

Remark 2.7 ($Smax(Smax(CI(\alpha))_{c_2})_{c_3}$)
Given a knowledge base $K = \langle \Delta_K,\ P_K \rangle$ *and an input* α*, such that* $CI(\alpha)$ *is non-empty and condition* $c_3 = Smax(CI(\alpha))_{c_2} \bigcup \Delta_K \nvdash \alpha$*, by Definition 2.8, we have that* $Smax(Smax(CI(\alpha))_{c_2})_{c_3}$ *is the following set:*

$$Smax(Smax(CI(\alpha))_{c_2})_{c_3} = \begin{cases} Smax(CI(\alpha))_{c_2}, \quad \text{if } c_3 \text{ is satisfied;} \\ Smax(CI(\alpha))_{c_2} - Min(Fail(Smax(CI(\alpha))_{c_2})_{c_3}), \\ \quad \text{otherwise.} \end{cases}$$

$Smax(Smax(CI(\alpha))_{c_2})_{c_3}$ *is ordered by* $\leqslant$*, as a subset of* $CI(\alpha)$*.*

•

Proposition 2.4
Given a base $K = \langle \Delta_K,\ P_K \rangle$*, for any input sentence* α*, such that* $CI(\alpha)$ *is non-empty, anditisnotthecasethat* $\vdash \neg\alpha$*,* $Smax(Smax(CI(\alpha))_{c_2})_{c_3} \bigcup \Delta_K \nvdash \alpha$*.*

Definitions for the case that $P_K + \alpha \nvdash \perp$ and $K + \alpha \vdash \perp$

We need to get the set $\perp_\alpha$, which includes all the minimal subsets of Δ_K, whose elements together with the input sentence α and the set of integrity constraints P_K, generate inconsistency. So, by Definition 2.7, we obtain such a set as shown below.

Remark 2.8 ($\perp_\alpha$)
Given a knowledge base $K = \langle \Delta_K,\ P_K \rangle$ *and an input sentence* α*, such that it is not the case that* $\vdash \neg\alpha$*, assume that* c_4 *is the condition* $\Delta_K \bigcup P_K \bigcup \{\alpha\} \nvdash \perp$*. Let* $\perp_\alpha$ *be the set* $Fail(\Delta_K)_{c_4}$*, such that:*

$$\perp_\alpha = \begin{cases} \emptyset & \text{if } c_4 \text{ is satisfied;} \\ \{\ S \parallel S \subseteq \Delta_K\ ;\ S \cup P_K \cup \{\alpha\} \vdash \perp; & \\ \quad \text{and } S \text{ is minimal w.r.t. } \subseteq \} & \text{otherwise.} \end{cases}$$

Where each set S_i *of* $\perp_\alpha$*, for* $i = 1, 2, \cdots, n$*, is ordered by* $\leqslant$*, as a subset of* Δ_K*.*

As stated in Definition 2.3, since $\leqslant$ is a partial order on Δ_K, we might have more than one minimal element in the sets S_i of the family $\perp_\alpha$.

Example 2.5

Consider a base $K = \langle \Delta_K,\ P_K \rangle$, *where* $\Delta_K = \{b, c, d, q\}$, *for* $c \leqslant d$, $d \leqslant q$, *and* $b \leqslant d$. *Assume that* $P_K = \{a \wedge b \wedge c \wedge d \rightarrow \perp, a \wedge q \rightarrow \perp\}$. *If we consider the update input* a, *by definition we have that:*
$\perp_a = \{\{b, c, d\}, \{q\}\}$, *and*
$min(\perp_a) = \{\{b, c\}, \{q\}\}$.

By Proposition 2.1, we guarantee that it is sufficient to retract one only element of each subset of $\perp_\alpha$, from $\Delta_K \bigcup \{\alpha\}$, in order to achieve consistency. That is, If $\{\alpha\} \bigcup K \vdash \perp$, then, given a set $\perp_\alpha = \{S_1,\ S_2, \cdots,\ S_n\}$, it is sufficient to retract one element from each of its subsets S_i, for $i = 1, \cdots, n$, such that: $((\Delta_K \cup \{\alpha\}) - S(\perp_\alpha)) \cup P_K \nvdash \perp$, where: $S(\perp_\alpha) = \{x_i \parallel \forall S_i \in \perp_\alpha,\ \exists y \in S_i,$ such that $x_i = y\}$.

In order to retract a minimum number of sentences from Δ_K, to accomplish the inclusion of the input sentence α in the revised set $K ® \alpha$, we would like to be able to define a choice mechanism, in which only one minimal element out of each set S_i of $\perp_\alpha$ could be chosen. However, in most cases, such mechanisms are based on empiric premisses which are often non-justified for a general framework.

One safe option, is to retract from Δ_K all the minimal elements of each set S_i of $\perp_\alpha$, by obtaining a safe-maximal subset of Δ_K, $Smax(\Delta_K)_{c_4}$, relative to the same condition of $\perp_\alpha$, as defined previously. This is certainly not the ideal approach, since we throw away more sentences than necessary in order to get rid of inconsistency. However, it plays a neutral, or impartial, role in choosing among the minimal elements. One can argue that this option is justified by the fact that, since the minimal elements are not related (or comparable) by the ordering $\leqslant$, there is no criterium which allows us to choose a particular one among them for retraction. Hence, the final decision is to remove all of them for regaining consistency. We call this option *"the impartial-choice for minimal change"*.

Another option would be to allow the current application to define which elements to retract from Δ_K. This option is free from a non-justifiable general choice mechanism. Instead, it can be viewed as a user-choice approach, since it allows the user to decide which minimal element to discard from each set of minimals of S_i of $\perp_\alpha$, when they have more than one element. By doing so, we can have that only one minimal element from each set of minimals of S_i is going to be retracted from Δ_K. This is shown in Proposition ?? to be a sufficient condition for avoiding inconsistency in the revised base. Consequently, within this option, the conservativity principal can be fully satisfied, since it supports the performance of revision by maintaining as many as possible of the old sentences in Δ_K. We call this option *"the user-choice for minimal change"*.

Here, we adopt a combined approach, with both *"the impartial-choice for minimal change"* and *"the user-choice for minimal change"* options. We assume that the set R_α selects the minimal elements to be retracted from Δ_K, in order to accomplish $K ® \alpha$, having *"the user-choice for minimal change"* as the main option, when it applies, and *"the impartial-choice for minimal change"* as default.

In order to construct the set R_α, we obtain the sets $min(\perp_\alpha)$, $min(\perp_\alpha)^+$, and $min(\perp_\alpha)^1$, according to the general definitions 2.3, 2.4 and 2.5, respectively.

Then, we need to define a function which caters for the choice of retraction, combining *"the impartial-choice for minimal change"* and *"the user-choice for minimal change"*. We call such a function *Choice*. It is defined, having as input parameters the option from the user and an entry-set. And as output, it supplies the set of chosen sentences.

The intuitive idea of the parameter *option* of the *Choice* function, is that the user has to supply an input to it, when requested. The two possible inputs for this parameter is either the option u, or the option d; meaning that it is a user-choice or a default-choice, respectively. Hence, when the function *Choice* is invoked, we assume that an option u or d is already available from the user, via some user-interface mechanism.

When the set $min(\perp_\alpha)^+$ is empty, there is no need to request the user for the choice of one of the minimal elements of each set S_i, since all the $min(S_i)$ sets of $min(\perp_\alpha)$ are unitary. In this case, the default-choice is applied directly.

When the user is requested the choice among the minimals of each set of $min(\perp_\alpha)^+$, he can also decide for the default option, which is *"the impartial-choice for minimal change"*. In this case the parameter $option$ of the $Choice$ function has d as answer. Hence, we have to cater for this case as well.

We define below a choice request function for the user, which is performed via some user-interface procedure and return u or d as an answer. We call this function Req. Whenever the set $min(\perp_\alpha)^+$ is non-empty, Req asks the user for a choice of how to perform the retraction needed for revision.

Definition 2.10 (*Req*)

Assume that L is a propositional language closed under the logical operators. Given a set $min(\perp_\alpha)^+$, let Req be a function from the set In into itself, $Req : In \longrightarrow In$, where In represents the set of all the possible user inputs to the system, such that $In \subseteq L$. And Set is the set of all sets of sentences of L. When the function applies, we write $Req(\{d, u\}) = option$, where $option$ is obtained conditined to the following:

$$option = \begin{cases} \{d\} & \text{if } min(\perp_\alpha)^+ = \emptyset; \\ \\ \{\ x \ \| \ x \in \{d, u\}; \\ \ \text{and } x \text{ is chosen by the user}\} & \text{otherwise.} \end{cases}$$

□

If the user accepts the task of making *"the user-choice for minimal change"*, i.e. $option = \{u\}$, he has to have the set $min(\perp_\alpha)^+$ available, to choose the minimal elements from. The definition below caters for this case.

Definition 2.11 (*Choice*)

Given a language L, a set $min(\perp_\alpha)$, and a set $min(\perp_\alpha)^+$, let $Choice$ be a function from the domain represented by the tuple $\langle In, \Omega\rangle$ into Ω, $Choice : \langle In, \Omega\rangle \longrightarrow \Omega$, where In is as in definition 2.10, and Ω is the set of all sets of sentences of L. We write $Choice(option, entry\text{-}set) = chosen\text{-}set$, where $option$ is either $\{d\}$ or $\{u\}$, and $entry\text{-}set$ is conditioned to $option$ as follows:

$$entry\text{-}set = \begin{cases} min(\perp_\alpha) & \text{for } option = \{d\}; \\ min(\perp_\alpha)^+ & \text{for } option = \{u\}. \end{cases}$$

The resuting $chosen\text{-}set$ of minimals is obtained in the following way:

$$chosen\text{-}set = \begin{cases} \cup\,\{X \ \| \ X \in entry\text{-}set\} & \text{if } option = \{d\}; \\ \\ \{x_j \ \| \ \forall min(S_i)_j \in entry\text{-}set, \\ \exists y \in min(S_i)_j, \text{such that } x_j = y; \\ \text{and } x_j \text{ is chosen by the user}\} & \text{if } option = \{u\}. \end{cases}$$

Where $1 \leq i \leq n$, and $j = 1, \cdots, k$, for $k \leq n$.[22]

□

Remark 2.9

In Definition 2.11, we count again on a user-interface procedure which, given an entry-set as a family of non-unitary sets, returns a set of chosen minimals. And each of its elements is chosen from each of the sets of the entry-set family.

•

Now we can define the set R_α, as motivated previously.

[22]The index n refers to the number of elements (sets) in $\perp_\alpha$ and $min(\perp_\alpha)$. And the index k refers to the number of elements (sets) in $min(\perp_\alpha)^+$.

Definition 2.12 (R_α)

Given a knowledge base $K = \langle \Delta_K,\ P_K \rangle$*, an input sentence* α*, and the sets* $\bot_\alpha$*, such that* $\bot_\alpha \neq \emptyset$*, chosen-set and* $min(\bot_\alpha)^1$*. Let* R_α *be the following set:*

$$R_\alpha = chosen\text{-}set \cup min(\bot_\alpha)^1.$$

Where R_α *is ordered w.r.t.* $\leqslant$*, as a subset of* Δ_K*.*

□

Remark 2.10

Notice that in the case that the set chosen-set is obtained with option $= \{d\}$*, the set* $min(\bot_\alpha)^1$ *is already included in it.*

•

Example 2.6 (R_α)

Consider a base $K = \langle \Delta_K,\ P_K \rangle$*, where* $\Delta_K = \{b,\ c,\ d,\ e,\ f,\ q,\ d \rightarrow e\}$*, for* $b \leqslant d$*,* $c \leqslant d$*,* $d \leqslant q$*,* $d \leqslant d \rightarrow e$*,* $e \leqslant c$*, and* $f \leqslant q$*. Assume that* $P_K = \{a \wedge b \wedge d \wedge f \rightarrow \bot,\ a \wedge c \wedge e \rightarrow \bot,\ a \wedge q \rightarrow \bot\}$*. If we consider the update input* a*, by definition we have that:*
$\bot_a = \{\{b, f, d\},\ \{e, c\},\ \{q\}\}$*,*
where:
$min(\bot_a) = \{\{b, f\},\ \{e\},\ \{q\}\}$*;*
$min(\bot_a)^+ = \{\{b, f\}\}$*; and*
$min(\bot_a)^1 = \{e,\ q\}$*.*
If option $= \{d\}$*, we have that* $R_a = \{b,\ e,\ f,\ q\}$*; and*
if option $= \{u\}$*, we have that* R_a *is either the set* $\{b,\ e,\ q\}$ *or the set* $\{\ e,\ f,\ q\}$*, depending on the user's choice.*

•

Now we know, from R_α, which elements to retract from Δ_K, in order to introduce the input α and keep consistency in the new base. However, we still need to guarantee that retracting R_α from Δ_K, we cannot derive from $(\Delta_K - R_\alpha)$ any element of R_α. In Example 2.6, for instance, we would have that $(\Delta_K - R_\alpha) \rightarrow e$ and $e \in R_\alpha$. In order to formalize this notion, we get a safe-maximal subset of $(\Delta_K - X)$, for a general set X, that should be retracted from Δ_K, where $X \subseteq \Delta_K$.

Remark 2.11 ($Smax(\Delta_K - X)_{c_5}$)

Given a knowledge base $K = \langle \Delta_K,\ P_K \rangle$*, a set* $X \subseteq \Delta_K$*, and the condition* $c_5 = (\Delta_K - X) \nvdash x,\ \forall x \in X$*. By Definition 2.8, we have that* $Smax(\Delta_K - X)_{c_5}$ *is the following set:*

$$Smax(\Delta_K - X)_{c_5} = \begin{cases} (\Delta_K - X) & \text{if } c_5 \text{ is satisfied;} \\ \\ (\Delta_K - X) - Min(Fail(\Delta_K - X)_{c_5}) & \text{otherwise.} \end{cases}$$

$Smax(\Delta_K - X)_{c_5}$ *is ordered by* $\leqslant$*, as a subset of* Δ_K*.*

•

The proposition below, guarantees that when we retract from Δ_K a set X, for $X \subseteq \Delta_K$, we either have that $(\Delta_K - X) \nvdash x,\ \forall x \in X$, or that $Smax(\Delta_K - X)_{c_5} \nvdash x,\ \forall x \in X$, where $Smax(\Delta_K - X)_{c_5}$ is a safe maximal subset of $(\Delta_K - X)$, conditioned to c_5.

Proposition 2.5

Given a base $K = \langle \Delta_K,\ P_K \rangle$*, for any set* $X \subseteq \Delta_K$*, we have that* $Smax(\Delta_K - X)_{c_5} \nvdash x,\ \forall x \in X$*.*

So, we already know which elements we have to retract when an input α conflicts with K, and we also have guaranteed that retracting those elements from Δ_K will not make them derivable from the resulting set. However, as described previously, the contraction for compromised revision allows the consistent consequences of the sentences to be retracted, to become available in the resulting revised base. Hence, we also have to cater for introducing the consequences of the elements of R_α w.r.t. Δ_K, provided that they do not conflict with the base $K = \langle (Smax(\Delta_K - R_\alpha)_{c_5} \cup \alpha),\ P_K\rangle$.

We use the notion of propagation of the ordering $\leqslant$, as in Definition 2.2, for defining the set of consequences of the elements that should be retracted from Δ_K. We call such a set $CR(X')$. In $CR(X')$, the set X' is the union of X, for any arbitrary set $X \in \Delta_K$, and the set $Min(Fail(\Delta_K - X)_{c_5})$, which contains the elements of Δ_K that should be retracted from it in order to guarantee that no element of X is derivable after X has been retracted from Δ_K. Considering X' for obtaining $CR(X')$, we get the consequences of all the elements effectively retracted from Δ_K.

Definition 2.13 ($CR(X')$)

Given a knowledge base $K = \langle \Delta_K,\ P_K\rangle$ *and a set* $X \subseteq \Delta_K$, *such that* X *should be retracted from* Δ_K, *together with* $Min(Fail(\Delta_K - X)_{c_5})$, *so that it is guaranteed that* $Smax(\Delta_K - X)_{c_5} \nvdash x,\ \forall x \in X$, *assume that* $X' = X \bigcup Min(Fail(\Delta_K - X)_{c_5})$. *Let* $CR(X')$ *be the set of consequences of the elements in* X' *w.r.t.* Δ_K, *considering* $\vdash$, *such that:*

$$\begin{aligned} CR(X') \ = \ & \{y \parallel \Delta_K \vdash y \text{ and } Smax(\Delta_K - X)_{c_5} \nvdash y; \\ & \text{and } \forall x \in X',\ y \neq x, \text{ and } Smax(\Delta_K - X)_{c_5} \cup \{y\} \nvdash x\}, \end{aligned}$$

where $CR(X')$ *is maximal w.r.t.* $\subseteq$, *and is ordered by* $\leqslant$, *according to Definition 2.2.*

□

Example 2.7 ($CR(X')$)

Given a base $K = \langle \Delta_K,\ P_K\rangle$, *where*
$\Delta_K = \{b,\ c,\ d,\ q,\ q \rightarrow e,\ c \rightarrow f,\ d \wedge e \rightarrow h,\ h \rightarrow q\}$,
such that $b \leqslant d$, $c \leqslant d$, $d \leqslant q$, $q \leqslant q \rightarrow e$, $c \leqslant c \rightarrow f$, $d \leqslant d \wedge e \rightarrow h$, *and* $h \leqslant h \rightarrow q$. *Let* $X \subseteq \Delta_K$, *be the set* $X = \{b,\ c,\ q\}$. *By Remarks ??, ??, and 2.11,* $Smax(\Delta_K - X)_{c_5} = (\Delta_K - X) = \{d,\ q \rightarrow e,\ c \rightarrow f,\ d \wedge e \rightarrow h,\ h \rightarrow q\}$. *By Definition 2.13, we have that* $X' = X$, *and* $CR(X') = \{e,\ f\}$, *where* e *and* f *are not related by the ordering* $\leqslant$.

•

We are interested in including the set $CR(X')$ in $Smax(\Delta_K - X)_{c_5}$, without being able to derive any element of X from the resulting base. But with Definition 2.13, this is not yet possible. Below, we state some points that need to be observed.

Remark 2.12

Notice that it is not guaranteed that $CR(X') \bigcup Smax(\Delta_K - X)_{c_5} \nvdash x,\ \forall x \in X'$. *In Example 2.7, for instance, we would be able to derive* q *from* $CR(X') \bigcup Smax(\Delta_K - X)_{c_5}$. *We still need to specify a subset of* $CR(X')$, *preferrably maximal w.r.t. set-inclusion, that satisfies this condition. However, we do not need to check for consistency of the base* $\langle (\Delta_K - X)_{c_5} \cup CR(X'),\ P_K\rangle$, *since the original base,* $K = \langle \Delta_K,\ P_K\rangle$, *is assumed to be consistent and as* $X' \subseteq \Delta_K$, $CR(X') \subset Cn(\Delta_K)$, *and* $Cn(\Delta_K) \neq K_\perp$.

•

We then state formally the subset of $CR(X')$, which satisfies the condition stated in Remark 2.12, by obtaining a safe-maximal subset of $CR(X')$, $Smax(CR(X'))_{c_6}$, relative to condition $c_6 = CR(X') \bigcup Smax(\Delta_K - X)_{c_5} \nvdash x,\ \forall x \in X'$, and to the set $Fail(CR(X'))_{c_6}$.

Remark 2.13 ($Smax(CR(X'))_{c_6}$)

Given a knowledge base $K = \langle \Delta_K,\ P_K\rangle$, *a set* $X \subseteq \Delta_K$, *such that* $CR(X')$ *is non-empty, and the condition* $c_6 = CR(X') \cup Smax(\Delta_K - X)_{c_5} \nvdash x,\ \forall x \in X'$. *By Definition 2.8, we have that* $Smax(CR(X'))_{c_6}$ *is the following set:*

$$Smax(CR(X'))_{c_6} = \begin{cases} CR(X') & \text{if } c_6 \text{ is satisfied;} \\ \\ CR(X') - Min(Fail(CR(X'))_{c_6}) & \text{otherwise.} \end{cases}$$

$Smax(CR(X'))_{c_6}$ is ordered by $\leqslant$, as a subset of $CR(X')$.

•

Proposition 2.6

Given a base $K = \langle \Delta_K,\ P_K \rangle$, for any set $X' \subseteq \Delta_K$, such that $CR(X')$ is non-empty, $Smax(CR(X'))_{c_6} \cup Smax(\Delta_K - X)_{c_5} \nvdash x,\ \forall x \in X'$.

Example 2.8 ($Smax(CR(X'))_{c_6}$)

Considering Example 2.7, in which $\Delta_K = \{b,\ c,\ d,\ q,\ q \rightarrow e,\ c \rightarrow f,\ d \wedge e \rightarrow h,\ h \rightarrow q\}$, such that $b \leqslant d$, $c \leqslant d$, $d \leqslant q$, $q \leqslant q \rightarrow e$, $c \leqslant c \rightarrow f$, $d \leqslant d \wedge e \rightarrow h$, and $h \leqslant h \rightarrow q$; $X = \{b,\ c,\ q\}$; $X' = X$ and $CR(X') = \{e,\ f\}$, where e and f are not related by the ordering $\leqslant$. We have that: $Fail(CR(X'))_{c_6} = \{\{e\}\}$, $Min(Fail(CR(X'))_{c_6}) = \{e\}$, and $Smax(CR(X'))_{c_6} = \{f\}$,

•

Since $Smax(CR(X'))_{c_6}$ is defined for a general compromised contraction of a set $X \subseteq \Delta_K$ from Δ_K, $Smax(CR(X'))_{c_6}$ is not sensible to the revision of K by an input α. By this we mean that $CR(X')$ and $Smax(CR(X'))_{c_6}$ refer to the consequences of a subset X which is retracted from Δ_K, without any link to a compromised revision on K. Hence, consistency is not guaranteed when we have $Smax(CR(X'))_{c_6} \cup (Smax((\Delta_K - X)_{c_5} \cup \{\alpha\}) \cup P_K$.

When we have a compromised contraction in Δ_K, triggered by a revision w.r.t. an input α, we have to cater for a restricted version of $Smax(CR(X'))_{c_6}$, which takes into account consistency w.r.t. the input α, and maximality w.r.t. $\subseteq$. Here, again, we adopt the notion of safe-maximal subset. This time, we need a safe-maximal subset of $Smax(CR(X'))_{c_6}$, where $X' = R'_\alpha$, for $R'_\alpha = R_\alpha \bigcup Min(Fail(\Delta_K - X)_{c_5})$, since we want to refer to the set of elements that should be retracted from Δ_K, relative to an input α. We then construct the set $Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7}$, where condition c_7 is $Smax(CR(R'_\alpha))_{c_6} \bigcup (Smax((\Delta_K - R_\alpha)_{c_5} \cup \{\alpha\}) \bigcup P_K \nvdash \perp$.

Remark 2.14 ($Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7}$)

Given a knowledge base $K = \langle \Delta_K,\ P_K \rangle$, an input sentence α, such that it is not the case that $\vdash \neg\alpha$, $P_K \bigcup \{\alpha\} \nvdash \perp$ and $K \bigcup \{\alpha\} \vdash \perp$, a set R_α, as in Definition 2.12, such that $Smax(CR(R'_\alpha))_{c_6}$ is non-empty, and the condition c_7 is $Smax(CR(R'_\alpha))_{c_6} \bigcup (Smax(\Delta_K - R_\alpha)_{c_5} \cup \{\alpha\}) \bigcup P_K \nvdash \perp$. By Definition 2.8, we have that $Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7}$ is the following set:

$$Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7} = \begin{cases} Smax(CR(R'_\alpha))_{c_6}, & \text{if } c_7 \text{ is satisfied;} \\ \\ Smax(CR(R'_\alpha))_{c_6} - Min(Fail(Smax(CR(R'_\alpha))_{c_6})_{c_7}), \\ \text{otherwise.} \end{cases}$$

The set $Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7}$ is ordered by $\leqslant$, as a subset of $CR(R'_\alpha)$.

•

Proposition 2.7

Given a base $K = \langle \Delta_K,\ P_K \rangle$, and an input sentence α, such that it is not the case that $\vdash \neg\alpha$, $P_K \bigcup \{\alpha\} \nvdash \perp$ and $K \bigcup \{\alpha\} \vdash \perp$, a set R_α, as in Definition 2.12, such that $Smax(CR(R'_\alpha)_{c_6}$ is non-empty, $Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7} \bigcup (Smax(\Delta_K - R_\alpha)_{c_5} \cup \{\alpha\}) \bigcup P_K \nvdash \perp$.

Definitions for Compromised Contraction & Revision

Below we define a general compromised contraction for our method. Notice that Definition 2.14 does not cater for the accomplishment of a compromised revision. It only specifies a compromised contraction, independent from any other operation.

Definition 2.14 (Compromised Contraction $K \ominus X'$)
Given a knowledge base $K = \langle \Delta_K, P_K \rangle$, and a set X', such that $X' \in \Delta_K$. And given the set $CR(X')$, as in Definition 2.13, let $\ominus$ denote the compromised contraction operation, which retracts X' from K, such that $K \ominus X'$, is the following:

$$K \ominus X' = \langle (Smax(\Delta_K - X)_{c_5} \cup Smax(CR(X'))_{c_6}), P_K \rangle.$$

□

From the definition above, we can construct the compromised contraction which sensible to a revision on the base. That is, the contraction of the elements of R'_α from K, in order to accomplish the revision $K \circledR \alpha$. Next, we formalize this notion of contraction, and we call it compromised contraction for revision.

Definition 2.15 (Compromised Contraction for Revision)
Given a knowledge base $K = \langle \Delta_K, P_K \rangle$, an input sentence α, and a set R_α, as in Definition 2.12. Let $[\ K \ominus R'_\alpha\ ]_\alpha$ be the compromised contraction of R'_α from K, for the achievement of the revision of K by α, where $R'_\alpha = R_\alpha \bigcup Min(Fail(\Delta_K - R_\alpha)_{c_5})$. $[K \ominus R'_\alpha]_\alpha = \langle \Delta_{[KR'_\alpha]_\alpha}, P_K \rangle$, and $\Delta_{[KR'_\alpha]_\alpha}$ is obtained in the following way:

$$\Delta_{[KR'_\alpha]_\alpha} = Smax(\Delta_K - R_\alpha)_{c_5} \cup Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7}.$$

□

Now, we can build up our compromised revision function. But first, we re-describe the compromised revision steps, more specifically now, by taking into account the definitions of this section.

The Revision Steps Revisited

- We add the new input α to the non-protected part Δ of the base $K = \langle \Delta_K, P_K \rangle$, such that $x \leqslant \alpha$, $\forall x \in \Delta_K$. The ordered set Δ_K augmented by α is then referred to as Δ'_K.
- If the base is inconsistent because α contradicts a tautology of the logical system, then we make the base consistent by rejecting the input. If $\vdash \neg\alpha$, then $K \circledR \alpha = K$.
- If the base is inconsistent because $P_K \bigcup \{\alpha\} \vdash \bot$, then we make the base consistent by rejecting α, but allowing its consistent consequences to be added to the base. That is, we get $CI(\alpha)$ and $Smax(Smax(CI(\alpha))_{c_2})_{c_3}$, and our compromised result is given by the knowledge base
$K \circledR \alpha = \langle (\Delta_K \bigcup Smax(Smax(CI(\alpha))_{c_2})_{c_3}), P_K \rangle$.
- If the base is inconsistent because $\Delta'_K \bigcup P_K \vdash \bot$ when $P_K \bigcup \{\alpha\} \nvdash \bot$, then we make the base consistent by keeping α in it and rejecting from it some old sentences. We also take into account the consistent consequences of the retracted sentences. This is done in the following way:

  First we get $\bot_\alpha$; R_α, $Smax(\Delta - R_\alpha)_{c_5}$, $CR(R'_\alpha)$ and $Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7}$.

  Then, we make the compromised contraction $[\ K \ominus R'_\alpha\ ]_\alpha = \langle \Delta_{[KR'_\alpha]_\alpha}, P_K \rangle$, where $\Delta_{[KR'_\alpha]_\alpha} = Smax(\Delta - R_\alpha)_{c_5} \cup Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7}$.

  Finally, we perform the compromised revision of the base K by α, such that:

$$K \circledR \alpha = \langle (\Delta_{[KR'_\alpha]_\alpha} \cup \{\alpha\}), P_K \rangle.$$

We formalize now the definition of the compromised revision function considering the steps described above.

Definition 2.16 (Compromised Revision Function)

Given a knowledge base $K = \langle \Delta_K,\ P_K \rangle$ *and an input sentence* α*, let the operation* Ⓡ *denote the compromised revision of* K *by* α*, such that the result* K Ⓡ α *is a new knowledge base with the same structure of* K*. We denote by* $\Delta_{K Ⓡ \alpha}$ *the resulting non-protected part of* K Ⓡ α*, such that* K Ⓡ $\alpha = \langle \Delta_{K Ⓡ \alpha},\ P_K \rangle$*. And* $\Delta_{K Ⓡ \alpha}$ *is obtained, such that one of the following conditions holds:*

(Case 1) *If* $\alpha \in \Delta_K$ *or* $\vdash \neg\alpha$*, then* $\Delta_{K Ⓡ \alpha} = \Delta_K$*.*

(Case 2) *If* $K + \alpha \nvdash \perp$ [23]*, then* $\Delta_{K Ⓡ \alpha} = \Delta_K \bigcup \{\alpha\}$*.*

(Case 3) *If* $P_K + \alpha \vdash \perp$*, then* $\Delta_{K Ⓡ \alpha} = \Delta_K \bigcup Smax(Smax(CI(\alpha))_{c_2})_{c_3}$*.*

(Case 4) *If* $K + \alpha \vdash \perp$ *and* $P_K + \alpha \nvdash \perp$*, then* $\Delta_{K Ⓡ \alpha} = Smax(\Delta_K - R_\alpha)_{c_5} \cup Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7} \bigcup \{\alpha\}$*.*

□

2.0.8 Some Properties of Compromised Revision

In this section, we present some properties of the compromised revision function, via proposition and theorem statements[24].

The proposition below states that all the consistent consequences of an input sentence α, which violates some integrity contraints of a base K, and which were not orginally in K, are available in K Ⓡ α.

Proposition 2.8
Given a base $K = \langle \Delta_K,\ P_K \rangle$*, for any input sentence* α*, such that it is not the case that* $\vdash \neg\alpha$*, and* $CI(\alpha)$ *is non-empty, if* $P_K + \alpha \vdash \perp$*, then* $\forall x \in Smax(Smax(CI(\alpha))_{c_2})_{c_3}$*,* $K \nvdash x$ *and* K Ⓡ $\alpha \vdash x$*.*

The proposition to follow states that all the consistent consequences of the retracted sentences from Δ_K, w.r.t. an input sentence α, in order to achieve K Ⓡ α, are derivable from K Ⓡ α.

Proposition 2.9
Given a base $K = \langle \Delta_K,\ P_K \rangle$*, for any input sentence* α*, such that it is not the case that* $\vdash \neg\alpha$*, if* $K + \alpha \vdash \perp$ *and* $P_K + \alpha \nvdash \perp$*, then* $\forall x \in Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7}$*,* $K \vdash x$ *and also* K Ⓡ $\alpha \vdash x$*.*

The two propositions to follow, state important conditions concerning the elements of the base K Ⓡ α.

Proposition 2.10
Given a base $K = \langle \Delta_K,\ P_K \rangle$*, for any input sentence* α*, such that it is not the case that* $\vdash \neg\alpha$*, if* $K + \alpha \vdash \perp$ *and* $P_K + \alpha \nvdash \perp$*, then* $\forall x \in R'_\alpha$*,* K Ⓡ $\alpha \nvdash x$*.*

Proposition 2.11
Given a base $K = \langle \Delta_K,\ P_K \rangle$*, for any input sentence* α*, such that it is not the case that* $\vdash \neg\alpha$*, if a sentence* $\beta \in K$Ⓡα*, then either* $\beta \in \Delta_K$ *and* $\beta \notin R'_\alpha$*; or* $\beta \notin \Delta_K$ *and either* $\beta \in Smax(Smax(CI(\alpha))_{c_2})_{c_3}$*, or* $\beta = \alpha$*, or* $\beta \in Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7}$*.*

Below, we state the consistency theorem of the compromised revision K Ⓡ α.

Theorem 2.1 (Consistency)
Given a base $K = \langle \Delta_K,\ P_K \rangle$*, for any input sentence* α*,* K Ⓡ $\alpha \nvdash \perp$*.*

Proof:

[23] By $K + \alpha \nvdash \perp$, we mean that $\Delta_K \bigcup P_K \bigcup \{\alpha\} \nvdash \perp$.

[24] The proofs omitted in this section are in the appendix of this paper.

By Definition 2.16, we state that $K \circledR \alpha = \langle \Delta_{K \circledR \alpha}, P_K \rangle$, where $\Delta_{K \circledR \alpha}$ is obtained conditioned to four cases.

In case 1, we have that if $\alpha \in \Delta_K$ or $\vdash \neg\alpha$, then $\Delta_{K \circledR \alpha} = \Delta_K$. So, it is guaranteed that $K \circledR \alpha \nvdash \bot$, since by assumption $K \nvdash \bot$.

In case 2, we also guarantee that $K \circledR \alpha \nvdash \bot$, for $\Delta_{K \circledR \alpha} = \Delta_K \bigcup \{\alpha\}$, and by hypotheses we have that $K + \alpha \nvdash \bot$.

In case 3, if $P_K + \alpha \vdash \bot$, then $\Delta_{K \circledR \alpha} = \Delta_K \bigcup Smax(Smax(CI(\alpha))_{c_2})_{c_3}$. By Proposition 2.3, we have proved that $Smax(CI(\alpha))_{c_2} \bigcup \Delta_K \bigcup P_K \nvdash \bot$. Since by Definition 2.8 and Remark 2.7, $Smax(Smax(CI(\alpha))_{c_2})_{c_3}$ is a subset of $Smax(CI(\alpha))_{c_2}$, we also have that
$Smax(Smax(CI(\alpha))_{c_2})_{c_3} \bigcup \Delta_K \bigcup P_K \nvdash \bot$. Hence, we guarantee that also in case 3 $K \circledR \alpha \nvdash \bot$.

In case 4, if $K + \alpha \vdash \bot$ and $P_K + \alpha \nvdash \bot$, then we have that $\Delta_{K \circledR \alpha} = Smax(\Delta_K - R_\alpha)_{c_5} \cup Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7} \bigcup \{\alpha\}$. By Proposition 2.7, we have proved that
$Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7} \bigcup Smax(\Delta_K - R_\alpha)_{c_5} \bigcup \{\alpha\} \bigcup P_K \nvdash \bot$. Hence, it is already guaranteed for case 4 that $K \circledR \alpha \nvdash \bot$.

■

Tenacity, as in [Han-91], is also a property of compromised revision.

Proposition 2.12 (Tenacity)
$\forall x \in K$ either $x \in K \circledR \alpha$ or $\neg x \in Cn(K \circledR \alpha)$.

Persistence in Compromised Revision

Intuitively, the persistence notion - a well-established notion within belief revision approaches, states that as much of the former base should survive a revision as possible. Hence, by revising a base K with a sentence α and then retracting α, we should be able to derive from the resulting base, all the consequences of K that do not directly contradict α.

Below, we present a compromised version of the persistence notion, considering that we should be able to derive from $(K \circledR \alpha) \ominus \{\alpha\}$ all the consequences of K that do not directly contradict α, and also that do not violate integrity contraints in K.

Proposition 2.13 (Compromised Persistence)
Given a knowledge base $K = \langle \Delta_K, P_K \rangle$ if we revise it by a sentence α, such that it is not the case that $\vdash \neg\alpha$, then $\forall x$ such that $K \vdash x$, $(K \circledR \alpha) \ominus \{\alpha\} \vdash x$, provided that $x \neq \alpha$ and that $x \notin R'_\alpha$.

Proof:

By Definition 2.16, we have that in the cases 1, 2 and 3, for $K \circledR \alpha$, no sentence is retracted from the base K. Then, we already have that $\forall x$ such that $K \vdash x$, $K \circledR \alpha \vdash x$. If we then retract $\{\alpha\}$ from $K \circledR \alpha$, by Definition 2.14 we do the following:
$(K \circledR \alpha) \ominus \{\alpha\} = \langle (Smax(\Delta_{K \circledR \alpha} - \alpha)_{c_5} \cup Smax(CR(\alpha'))_{c_6}), P_K \rangle$.

In the case that $\alpha \notin \Delta_K$, there is no chance that $(K \circledR \alpha) \ominus \{\alpha\}$ retracts from $\Delta_{K \circledR \alpha}$ any original element of K. Hence, for the cases 1, 2 and 3, of Definition 2.16, we have that compromised persistence is satisfied.[25] So, $\forall x$ such that $K \vdash x$, $(K \circledR \alpha) \ominus \{\alpha\} \vdash x$.

From case 4 of Definition 2.14, and from Definition 2.14, $(K \circledR \alpha) \ominus \{\alpha\} = \langle (Smax(\Delta_{K \circledR \alpha} - \alpha)_{c_5} \cup Smax(CR(\alpha'))_{c_6}),\ P_K \rangle$, where $\Delta_{K \circledR \alpha} = Smax(\Delta_K - R_\alpha)_{c_5} \cup Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7} \bigcup \{\alpha\}$. It is then easy to see that the consequences of K which are not anymore available from $(K \circledR \alpha) \ominus \{\alpha\}$ are the ones which allowed the elements of the set R'_α to be derivable, where $R'_\alpha \subseteq \Delta_K$. Also, if α was derivable from K, it is no longer derivable from $(K \circledR \alpha) \ominus \{\alpha\}$. Hence, also for case 4 we have that compromised persistence is satisfied, and $\forall x$ such that $K \vdash x$, $(K \circledR \alpha) \ominus \{\alpha\} \vdash x$, under the condition that $x \neq \alpha$ and that $x \notin R'_\alpha$ as stated in the proposition.

■

The Relationship between Revision and Contraction

It seems plausible that we also try to establish a relationship between our notions of compromised revision and contraction, as, for instance, the Levi Idendity does in the AGM theory.

For the AGM theory, the relation between revision and contraction is stated as follows:

$$K * \alpha \ = \ (K - \neg\alpha) \ + \ \alpha$$

where $-$ and $+$ denote the AGM contraction and expansion, respectively.

In the case of theory-bases, we can adapt this notion to:

$$K \circledast \alpha \ = \ (K - \{\neg\alpha\}) \ \cup \ \{\alpha\}.$$

For our compromised approach, we cannot simply say that $K \circledR \alpha = (K \ominus \{\neg\alpha\}) \cup \{\alpha\}$, since we also have to consider the violation of integrity constraints. Moreover, our relation between revision and contraction has to cater for the four possible cases of the input with relation to the knowledge base, as stated in Definition 2.16. Hence, we have that the correspondence between compromised revision and contraction is given as follows.

Definition 2.17 (Revision via Compromised Contraction)

A compromised revision of a knowledge base $K = \langle \Delta_K,\ P_K \rangle$ by a sentence α, can be seen as a composition of a compromised contraction and an expansion, in the set theoretical sense, such that:

$$K \circledR \alpha \ = \ [K \ominus A]_\alpha \bigcup B,$$

where A and B are sets which are conditionally obtained as follows:

(Case 1) *If $\alpha \in \Delta$ or $\vdash \neg\alpha$, then $A = \emptyset$ and $B = \emptyset$.*

(Case 2) *If $K + \alpha \nvdash \bot$, then $A = \emptyset$ and $B = \{\alpha\}$.*

(Case 3) *If $P_K + \alpha \vdash \bot$, then $A = \emptyset$ and $B = Smax(Smax(CI(\alpha))_{c_2})_{c_3}$.*

(Case 4) *If $K + \alpha \vdash \bot$ and $P_K + \alpha \nvdash \bot$, then $A = R'_\alpha$ and $B = \{\alpha\}$.*

□

Notes:

[25] In fact, for cases 1, 2 and 3, even the original notion of persistence would be satisfied.

It is trivial to show that the substitutions of the given sets A and B in $([K \ominus A]_\alpha) \bigcup C$, give the respective results for $K \circledR \alpha$, as stated in Definition 2.16.
Case 4 above, is the one that represents the success of our compromised revision, and also the one which resembles most the relation between revision and contraction of AGM. By making the propper substitutions for case 4, we have that $K \circledR \alpha = [K \ominus R'_\alpha]_\alpha \bigcup \{\alpha\}$.

Remark 2.15
By $[K \ominus A]_\alpha \bigcup B$, *we mean that* B *is included in the non-protected part* $\Delta_{[K A]_\alpha}$ *of* $[K \ominus A]_\alpha$, *such that the minimal elements in* B *have higher priority than the maximal elements of* $\Delta_{[K A]_\alpha}$, *w.r.t.* $\leqslant$.

•

Correspondence between the Postulates and the Revision Function

The theorems below state the correspondence between the compromised revision function formalization and the proposed postulates for compromised revision. Their proofs can be found in [Dar-96].

Since we have adopted the safe-maximal subset notion, instead of the notion of a maximal subset in our formalization, we cannot guarantee the maximality conditions of postulate ($\circledR$9). Therefore, the theorems below will refer to the postulates ($\circledR$1) to ($\circledR$8).

Theorem 2.2
If a contraction function is defined as a compromised contraction $\ominus$, *Definition 2.15 then the revision function* $\circledR$ *obtained from Definition 2.16, satisfies the postulates* ($\circledR$1) *to* ($\circledR$8).

Theorem 2.3
Given a compromised contraction function $\ominus$, *as in Definition 2.15, for any knowledge base* $K = \langle \Delta, P_K \rangle$, *and any sentence* α, *a revision function is a compromised revision* $\circledR$, *as in definition 2.16, if and only if it satisfies the postulates* ($\circledR$1) *to* ($\circledR$8).

2.1 Discussions

In this formalization of our approach to conflicting updates, the compromised revision was designed, based on a user-oriented choice for a minimal loss of information of the original base K[26], and on an impartial solution, via the notion of safe-maximality, for the other retraction choices among conflicting data, which arose when generating candidates of compromised consequences to be added in the final revised base.[27]

Our motivation to allow the user to choose a minimal element to be retracted from the set $min(\perp_\alpha)$, is grounded on the following arguments.

Although we have that the non-protected set of the base, Δ_K, is partially ordered according to application's requirements, in the case that the sentences are not comparable by the ordering, we lack the application background knowledge in order to design a selection function for choosing one of the minimal elements.

By allowing the user to make the choice, we are not imposing that the contraction function of our system is the only option to retract conflicting data.[28] Instead, we are offering a combined choice between the function and the application-oriented option, hoping for a more adequate result.

The high persistence flavour that the user-option brings to our system, meets our basic goals of contracting the former base minimally when revising it, while also catering for the availability of the consistent consequences of either restracted sentences or

[26] See Definitions 2.10, 2.11 and 2.12.
[27] See Definitions 2.8, and Remarks 2.5, 2.6, 2.7, 2.11, 2.13 and 2.14.
[28] In some cases where weak contraction functions, like the full meet for instance, are specified, the result can be very restrictive.

conflicting inputs. Furthermore, we gain in persistence without having to apply unjustified selection mechanisms w.r.t. the current application.

For deciding among the minimal elements of the conflicting compromised consequences of either α or R_α, which should not be added to the revised base in order to keep consistency, we have adopted an impartial solution via the safe-maximality notion. This notion restricts the notion of a maximal subset relative to a certain condition. Instead of using a selection function, based on an arbitrary ordering, to pick up one element of the set of minimals, we have decided to be more impartial and to get rid of all the minimals which are involved in the failure of the condition in question.

The major adavantage of this option is that it does not present the problem of non-uniqueness that we have with a maximal subset option, for instance. Also, it does not count with unjustified orderings for selecting a particular minimal element to be retracted.

Moreover, if an application requires a total ordered base, instead of a partially ordered one, all the safe-maximal sets of our system will be maximal. In this case also, the user will not be required for any choice, since there would be only one minimal element for every subset of the base that fails the underlying condition.

A drawback of the safe-maximality solution comes when we have many elements in the base which are not related by the ordering. In this case, such a solution would discard too much from the set of compromised consequences. Nevertheless, even in such a case, we believe that we would not underestimate the system's revision as a whole. Because we would be restricting the number of compromised consequences to be added to the revised base, and this is the special feature that our approach proposes, in relation to other revision methods, for keeping more information when retracting data, in the case of $CR'(R_\alpha)$, or for allowing partial acquisition of inputs, in the case of $CI'(\alpha)$.

There are some important points which were not covered by this revision characterization, since they were not central to the purpose of this paper. Among those points, we can cite the problem of the computational tractability of compromised revisions.

3 Final Remarks

3.0.1 Summary

In this paper, we have introduced the notion of compromised reasoning; and we point out some application areas, which may enjoy the benefits of compromised solutions. We have introduce the basic specification of our approach for the case of database updates, describing the adopted policies for reconciling conflicting updates. We have presented a compromised characterization to revision under the perspective of belief revision, by establishing some postulates and defining a base-theory revision function for specific applications which allow for compromised solutions. We have defined a compromised contraction for revision and stated the relation betwen that contraction and the compromised revision. Correspondence theorems and some relevant propositions were presented, establishing the relationships among functions and postulates.

We re-state that our main concern with the compromised revision formalization, was to characterize a specific model which applies for the special case of having compromised solutions to conflicting base updates, within the philosophy of our approach, and not to propose another general belief revision model.

The specific revision method proposed here, re-enforces the importance of having different theory change operators available for specific applications, in order to concieve the construction of a more realistic framework for theory revision.

3.1 Further Work

Under the compromised philosophy of the approach presented here, some other pieces of work were developed by the author. In [Dar-96], Dargam introduces two more formalizations for this approach. One is for labelled databases. It is based on the framework of Labelled Deductive Systems [Gab-94] using first order logic, and states formally all the concepts defined for the compromised policies of the approach. The other formalization addresses the problem of dealing with inconsistency after the performance of a database transaction, within the context of deductive databases. In this formalization, CIU is defined on the basis of the integrity-checking method for deductive databases described in [SaKo-87].

A planned further work in this research line, is the investigation of compromised solutions for modelling simultaneous occurrence of actions, where we have to tackle problems which arise when reasoning about possible conflicts and combined effects of these actions. We believe that this area can benefit much from compromising on solutions. We have studied the existing approaches related to these aspects in the context of actions, but so far not much has been developed on it.

In this paper, we have seen that compromised revision opens a lot of interesting points for research and discussions. Among many ideas for further work on this subject, we are committed to investigate in detail the relationship between the compromised revision and the AGM revision for bases, and also between the compromised revision and the update operation introduced in [KaMe-92]. We believe that the results to be achieved through this planned work can bring relevant contributions, concerning revision of bases which are ruled by integrity constraints.

Acknowledgements

This work was financially supported by the Brazilian Research Council, CNPq, under the grant 202078/90.6. The author is very grateful to Dov Gabbay for his support as a research supervisor, and to Odinaldo Rodrigues and Hans Jürgen Ohlbach, for their important comments and suggestions on earlier versions of this work.

References

[AlMa-82] C. Alchourrón, D. Makinson, *"The Logic of Theory Change: Contraction functions and their associated functions"*, Theoria 48,1982.

[AlMa-85] C. Alchourrón, D. Makinson, *"On the Logic of Theory Change: Safe Contractions"*, Studia Logica, 44, 1985.

[AlMa-86] C. Alchourrón, D. Makinson, *"Maps between some different kinds of contractions functions"*, Studia Logica, 45, 1986.

[AGM-85] C. Alchourrón, P. Gärdenfors, D. Makinson, *"On the Logic of Theory Change: partial meet functions for contraction and revision"*, Journal of Symbolic Logic, 50, 1985.

[BaCh-91] A. Bauval; L. Cholvy, *"Automated Reasoning in case of Inconsistency"*, in Proceedings of the first World Conference on Fundamentals of AI, 1991.

[Bes-90] P. Besnard, *"Logics for Automated Reasoning in case of Inconsistency"*, in the Proceedings of AI: Methodology, Systems and Applications, 1990.

[BoBe-93] C. Boutilier, V. Becher, *"Abduction as Belief Revision: A Model of Preferred Explanation"*, in Proc. of the AAAI Conference, pp. 642-648, 1993.

[BoGo-93] C. Boutilier, M. Goldszmidt, *"Revision by Conditionals Beliefs"*, in Proc. of the AAAI Conference, 1993.

[CeHe-86] L. Farinas del Cerro; A. Herzig, *"Reasoning about Database Updates"*, in Proceedings of the Workshop on Foundations of Deductive Databases and Logic, 1986.

[CeHe-92] L. Farinas del Cerro; A. Herzig, *"Constructive Minimal Changes"*, in Report IRIT, 1992.

[Cho-90] L. Cholvy, *"Querying an Inconsistent Database"*, in the Proceedings of AI: Methodology, Systems and Applications, 1990.

[Cho-93] L. Cholvy, *"Proving Theorems in a Multi-Source Environment"*, in the Proceedings of IJCAI-93, Chambery, vol.1 , pp 66-71, 1993.

[Dar-93] F.C.C. Dargam, *"Resolving Conflicting Actions and Updates"*, Transfer Report, Department of Computing, Imperial College, August 1993.

[Dar-96] F.C.C. Dargam, *"On Reconciling Conflicting Updates", (A Compromised Revision Approach)*, PhD. Thesis, Department of Computing, Imperial College, to appear.

[DaGa-93] F.C.C. Dargam; D. Gabbay, *"Resolving Conflicting Actions and Updates-Extended Abstract"*, Compulog Net Meeting on Knowledge Representation and Reasoning CNKRR'93, Lisbon, July 1993.

[FrLe-94] M. Freund, D. Lehmann, *"Belief Revision and Rational Inference"* Technical Report, TR94-16, Institute of Computer Science, The Hebrew University of Jerusalem, Israel, 1994.

[FUV-83] R. Fagin; J.D. Ullman; M. Vardi, *"On the Semantics of updates in Databases"*, in Proceedings of ACM-PODS, 1983.

[FKUV-88] R. Fagin;G. Kupper; J.D. Ullman; M. Vardi, *"Updating Logical Databases"*, in Advances in Computer Research, P. Kanellakis, editor, Morgan-Kaufman, Los Altos, CA, 1988.

[Gab-94] Dov Gabbay, *"LDS - Labelled Deductive Systems - Volume I Foundations"*, MPI-I-94-223, Max-Planck-Institut fur Informatik, Saarbrucken, Germany, 1994.

[GaHu-91] D. Gabbay; A. Hunter, *"Making Inconsistency Respectable - Part 1: A Logical Framework for Inconsistency in Reasoning"*, in Fundamentals of AI Research, LNCS 535, Springer-Verlag, 1991.

[GaHu-93] D. Gabbay; A. Hunter, *"Making Inconsistency Respectable - Part 2: Meta-level handling of inconsistency"*, LNCS 747, Springer-Verlag, 1993.

[Gal-90] J. Galliers, *"The Positive Role of Conflict in Cooperative Multiagent Systems"*, in Decentralized AI, edited by Y. Demazeau and J.P. Muller, North-Holland, pp 33-46 ,1990.

[Gär-88] P. Gärdenfors, *"Knowledge in Flux: Modeling the Dynamics of Epistemic States"*, Bradford Books, Cambridge, MA: The MIT Press, 1988.

[Gär-90] P. Gärdenfors, *"The Dynamics of Belief Systems: Foundations x Coherence Theories"*, Revue Internationale de Philosophie, vol. 44, n. 172, 1/1990.

[GäMa-88] P. Gärdenfors, D. Makinson, *"Revisions of Knowledge Systems using Epistemic Entrenchment"* in Proc. 2nd Conf. Theorectical Aspects of Reasoning about Knowledge, Edited by M. Vardi, Morgan Kaufmann, 1988.

[GGMO-94] D. Gabbay; L. Giordano; A. Martelli; N. Olivetti, *"Conditional Logic Programming"*, To appear in the Proceedings of ICLP'94.

[GiSm-88] M.L. Ginsberg; D.E. Smith, *"Reasoning About Action I : A Possible Worlds Approach"*, Artificial Intelligence, 35:165-195, 1988.

[GuLl-90] A. Guessoum; J.W. Lloyd, *"Updating Knowledge Bases"*, in New Generation Computing, 8(1), 71-89, 1990.

[JaPa-90] P.Jackson, J. Pais, *"Semantic Accounts of Belief Revision"*, in LNAI 515, J.P. Martins & M. Reinfrank (Eds.), TMS, ECAI-90 Workshop Proceedings, 1990.

[KaMe-92] H. Katsuno, A.O. Mendelzon, *"On the Difference between Updating a Knowledge Base and Revising it"*, in Belief Revision, edited by P.Gärdenfors, Cambridge University Press, pp. 183-203, 1992.

[KeWi-85] A. Keller, M. Winslett, *"On the use of an Extended Relational Model to handle Changing Incomplete Information"*, IEEE Transactions on Software Engineering, SE-11:7, pp. 620-633, 1985.

[KUV-84] G.M. Kupper; J.D. Ullman; M. Vardi, *"On the Equivalence of Logical Databases"*, in Proceedings of ACM-PODS, 1984.

[LDP-92] J. Lang; D. Dubois; H. Prades, *"Dealing with Multi-source Information in Possibilistic Logic"*, in Proceedings of ECAI, 1992.

[Mak-89] D. Makinson, *"General Theory of Cumulative Inference"*, in Nonmonotonic Reasoning, edited by Reinfrank et al., LNCS (LNAI) 346, Springer-Verlag, 1989.

[Mak-93] D. Makinson, *"Five Faces of Minimality"*, Studia Logica, Vol.53, n.3, 1993.

[MBM-95] D. Monteiro; E. Bertino; M. Martelli, *"Transactions and Updates in Deductive Databases"*, Technical Report DOC-95/2, Department of Computing, Imperial College, 1995.

[Mod-91] S. Modgil, *"Transaction Management in a Deductive Database"*, M.Sc. Thesis, Department of Computing, Imperial College, University of London, 1991.

[Mor-92] M. Morreau, *"Planning from first Principles"*, in Belief Revision, edited by P.Gärdenfors, Cambridge University Press, pp.204-219, 1992.

[NaRo-90] S. Naqvi; F. Rossi, *"Reasoning in Inconsistent Databases"*, in the Proceedings of the 1990 North American Conf. on Logic Programming, pp 255-272, 1990.

[Neb-89] B. Nebel, *"A Knowledge level Analysis of Belief Revision"*, in first Conference on Principles of Knowledge Representation and Reasoning, 1989.

[Neb-90] B. Nebel, *"Reasoning and Revision in Hybrid Representation Systems"*, Chapter 6 - Belief Revision, in LNAI 422, 1990.

[RaFo-89] A. Rao, N. Foo, *"Minimal Change and Maximal Coherence: A Basis for Belief Revision and Reasoning about Actions"*, in Proceedings of the 11th International Joint Conference on Artificial Intelligence, pp. 966-971, 1989.

[RoNa-89] F. Rossi; S. Naqvi, *"Contributions to the View Update Problem"*, in the Proceedings of the 6th International Conf. on Logic Programming, 1989.

[Rot-91] H. Rott, *"Two Methods of Constructing Contractions and Revisions of Knowledge Systems"*, Jounal of Philosophical Logic, 20, 1991.

[Rot-92] H. Rott, *"On the Logic of Theory Change: more maps between different kinds of contractions functions"* in Belief Revision, edited by P.Gärdenfors, Cambridge University Press, pp. 122-141, 1992.

[SaKo-87] F. Sadri; R. Kowalski, *"A Theorem-Proving Approach to Database Integrity"*, in Foundations of Deductive Databases & Logic Programming, Minker ed., 1987.

[Wag-93a] G. Wagner, *"Reasoning with Inconsistency in Extended Deductive Databases"*, in the Proceedings of the 2nd International Workshop on Logic Programming and Non-monotonic Reasoning, Pereira & Nerode editors, Lisbon, 1993.

[Wag-93b] G. Wagner, *"Vivid Logic"*, Lecture Notes in Artificial Intelligence, LNAI 764, Springer Verlag, 1994.

[Win-90] M. Winslett, *"Updating Logical Databases"*, Cambridge University Press, 1990.