A Compromised Characterization to Belief Revision

Imperial College Research Report DoC 96/2

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

25 March 1996

Abstract

This paper proposes a 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.

In more practical terms, consider the situation where K is a database and A an input. Assume that A is inconsistent with K. Current belief revision/update approaches will keep A and mantain consistency by selecting some element from K to form a revised database, usually denoted as $K * A$. There is a lot of research in this area, both theorectical, e.g.: the AGM theory of belief revision, and algorithmic research, e.g.: Reason Maintenance Systems.

Our aim is to offer an alternative approach, restricted to some specific applications, which is flexible enough to keep more data in K in the case of conflicts. We view the above situation as a *conflict* between two inputs (K 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 K only in the case that A does not directly contradict the K's integrity constraints, 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 K, its eventual consistent consequences,

w.r.t. the existing data of K , 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 proposes to generate the consequences of the conflicting inputs, and to get rid of the inconsistency, via a minimal number of retractions 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 non-supported consequences. This reconciliation of conflicting inputs follows some specified postulates for compromised revision.

Justifications for the proposed approach are mainly based on its practical applications. In particular *design processes*, where one builds up the goal state of a particular task, via performances of intermediary updates. Within this procedure, compromised results of updates can help to build up the goal state, when conflicts arise. This is so, because in general, our approach provides more information about the setting in order to carry on the application development. We also find applications in AI, in particular in *theory revision*, where our approach can provide a compromised way for revising a theory base with conflicting information.

■ Overview of the Paper

This paper is organized in the following way. In the introduction section, we present our main motivations and we discuss the general issue of conflict resolution and theory revision. We introduce the notion of compromised reasoning; and we point out some application areas, which may enjoy the benefits of the compromised solutions proposed in this work. As an additional point, some claims and viewpoints, extracted from the literature, are briefly described in order to serve as support for the current work. Also in Section 1, we introduce the basic specification of our approach for the case of database updates, describing the adopted policies for reconciling conflicting updates under a compromised reasoning perspective.

In Section 2, 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. A compromised contraction for revision is also defined. 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.

In Section 3, we summarize the obtained results and discuss about the approach of compromising conflicting updates. Existing work in labelled databases and in deductive databases under this approach are described briefly. Limitations of the formalized compromised revision presented in this paper, as well as proposals of further work and future research are presented.

Appendix A contains the proofs of the propositions cited in Section 2.

1 Introduction

One of the main goals in Artificial Intelligence is to design an intelligent agent, which comprises a knowledge base and a set of beliefs about the world, and which is able to incorporate new, and possibly contradictory, knowledge into that set. Pursuing this goal, researchers from different sub-areas of AI[1], like belief revision; diagnosis; learning; planning and reasoning about effects of actions, have to face common problems of theory revision[2].

Reasoning about updates, actions, changes and their implications, composes the scope of the basic motivation for this work. Within this scope, our main concern is to deal with the problem of conflicting updates. That is, the problem of deciding what to do when updates interfere with each other, generating contradictory data, or violating constraints that are given by the system in which they are applied.

Among the AI sub-areas mentioned before, we are interested in applications which involve database updates, belief revision, and reasoning about the effects of actions.

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*[3].

Our approach is mainly suitable for applications which allow for compromised solutions, i.e. solutions which present the closest result with relation to the expected one. Some of the applications which can benefit from our approach, are in the area of *design processes*. Here, one builds up the goal state of a particular task, via performances of intermediary updates. This procedure allows for compromised results of updates when conflicts arise.

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[4], and algorithmic research, e.g.: Reason Maintenance Systems[5]. 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[6], 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

[1] For a comprehensive reading about Artificial Intelligence and its applications, the reader is referred to [Ban-90],[PaDe-90],[Nil-80], Ric-89], [Sha-92] and [Win-92].

[2] See, for instance, [Bre-91],[Găr-88],[Mak-85], [Neb-91] and [RaFo-89].

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

[4] 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].

[5] 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].

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

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.

1.1 Motivations

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 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[7].

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

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

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[8], 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.[9] 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.

In a way, the proposed approach tries to represent a common aspect of life. In many occasions of everyday life situations, we try to do our best to get what we can, out of the things we originally wanted to achieve in life, and for some reason (possibly because of a conflicting situation at the time being considered), we do not satisfy all the requirements to get it. Hence, we end up with the most we can get out of the situation, without being able to get everything which was stated in the original goal. By pursuing this approach, it does not mean that we are working on the representation of the *frustrated loser*. On the contrary, we want to put forward an approach which has the positive aspect of *compromising and going further*, rather than *giving up*, when it faces a conflicting situation.

1.2 Some Approaches which handle 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.

[8] 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.*

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

For this work, we are interested in 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, which somehow takes advantage of the conflicting input.

Among the works which deal with inconsistency, we find in the literature logical studies which basically approach the problem in two main different lines: one which uses classical logic and eliminates inconsistency, by managing maximal consistent sets, for instance; and the other which uses non-classical logics in order to handle inconsistency.

Examples of studies which use classical logic are the works by Baral & Subrahmanian et.al. [MBKS-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. As well as Ginsberg & Smith [GiSm-88], in the area of reasoning about actions.

Examples of studies which do not use classical logic are listed here. 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-94], 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 maximal consistent 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 context-dependent. In the case of conflicting updates, the area of Database Theory has been the focus of many studies, which propose solutions to the problem in basically two main contextual approaches: in the View Updates context, from which we can cite the works by Fagin et al [FUV-83], [FKUV-88] and by Rossi & Naqvi [RoNa-89]; and in the context of Logical Databases, for instance, the works by Manchada & Warren [MaWa-87], Naqvi & Rossi [NaRo-90], and Guessoum & Lloyd [GuLl-90]. In [Win-90], we find a survey and classification[10] of a number of the proposed semantics for the context of updating logical databases, restricted to the propositional case. The main goal of the survey is to expose the differences between semantics that are relevant when one is deciding which semantics to use for a particular application.

In the case of actions, the areas of planning and reasoning about the effects of actions, have given grounds for the development of different approaches to handle the problem of conflicts and reasoning with inconsistent data.

The area of *reasoning about effects of actions* is considered a driving force behind some developments in the area of updates [Win-90]. Basically, the main problem in this area is that of adding a new formula, which describes the known effects of a given action, to a knowledge base which may contradict the formula added. The objective of this area is to formalize reasonable[11] ways of drawing common-sense conclusions about the effects of actions.

In *planning*, the main problem is to devise a sequence of actions that will change the current state of the world to some desired goal state. The work in planning counts on the contributions of the

[10] This classification ranges from model-based to formula-based and from permissive to restrictive semantics.

[11] By *reasonable*, here, we mean the way a human would do it, given the same information about the initial scenario and about the action to be performed.

reasearch done in the area of reasoning about effects of actions, since in order to achieve the goal state, we must be able to revise the knowledge base according to the known effects of the actions to be performed. Most of the approaches which deal with conflicting or interfering actions are oriented towards restricting their performance, and by doing so, avoiding the introduction of inconsistency into the system, with respect to the underlying theory.

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].

The problem of resolving conflicting data in order to keep the consistency of the information, also arises in an environment which handles different information sources (data / knowledge bases). In a multi-source information environment context, we may have inconsistent data in the global set of information, even if each separate source is consistent. Since the different sources of information are independently developed and also managed, the notion of global consistency does not exist. Nevertheless, the problem of consistency arises when grouping the several sources for the need of a particular application.

This kind of problem is treated by Dubois et al, in [LDP-92], using possibilistic logic, and by Cholvy in [Cho-93]. Within an application which needs to use information provided by several sources, Cholvy describes the case of solving a particular part of a global problem by dealing with several expert systems. Hence, the knowledge coming from each expert system has to be combined, by virtually grouping the knowledge of the different systems, since the knowledge may remain distributed among the systems. Cholvy defines a logical framework which allows reasoning with multi-sourced information, and which does not collapse in the case of inconsistency[12].

1.3 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.*

[12] As it is the case in classical logic.

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 two examples which illustrate the intuitive notion of our approach with relation to database updates and to actions, respectively.

An idea of how our approach would behave for the database update case, can be given via the example below.

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

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

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.*

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

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.

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

•

Remark 1.1

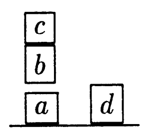
The example above could be interpreted with the following meanings for the sentences A, B and C:
A = "Executive Class Passenger";
B = "Extra baggage allowance";
C = "Economic Class Passenger".
Then, we would have that the database update above represents a situation, in which an executive class passenger 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.

•

In order to present an intuitive notion of our approach w.r.t. actions, let us consider a planning application, where actions are scheduled to be performed. In many applications, an action is taken, as part of a plan, in order to realize some goals. In a logical representation, these goals are logical consequences of the post-conditions of the actions, also known as side-effects, and the post-conditions themselves. Compromising on some of these goals, in case of conflict, means allowing for some consequences while retaining consistency.

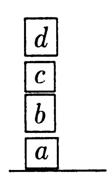
The following example illustrates the point of having actions which are initially inconsistent to be applied.

Example 1.2 *Let us consider a scenario in the Blocks World, where we have blocks a, b, c, and d, as shown the picture below:*



Suppose we want to perform the action of placing block d on top of block b. This action is not consistent, since it does not satisfy the constraint of the Blocks World, which says that if a block is not "free" another block cannot be placed on top of it. Nevertheless, if we think about compromising on the consequences, we may notice that the consequence of having d on top of b cannot be accepted, but we can accept the consequences of having d above b and d above a, which are the resulting consequences of performing this same action. So the solution to this problem would end up placing d on top of block c. What we are pursuing in this case is not the original update with all the consequences of the action, but, instead, we are implementing what we can call a compromised solution, by eliminating the inconsistent consequences because of the existing restrictions.

Resulting Scenario:



•

Remark 1.2

A possible interpretation for the previous example would be the situation in which the blocks a,b and c represent owners of flats of a building, which have been already sold by a state agency. The position that they occupy in the stack refer to their flats' respective floors in the building. Block d can be seen as a potential flat buyer, who, for instance, implies a constraint of not getting a flat on the first two floors, but higher. So, if a floor immediately above the second floor does not have available flats any longer, any other floor above would also be a solution.

•

The sub-section to follow introduces the notion of "compromised reasoning", concerning conflict resolution and theory revision, by presenting some examples of application areas which adopt compromised solutions in various ways.

1.4 Compromised Reasoning & Application Areas

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, in section 1.1, 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. Examples found in [Dar-89,91], summarized below, illustrate this case in the military defense application.

Consider a defense system in which a knowledge base KB contains information about all the possible threats, in terms of the enemies' weapons, concerning a particular geographic area. Assume that an input from a sensoring unit detects the presence of a threat γ which could not be expected to be at that particular area, according to the current knowledge base. Threat γ then conflicts with some information in KB. Nevertheless, the input comes from a reliable sensoring unit and has to be taken into account. One way of dealing with this situation could be to analyse the consequences of threat γ in KB, and according to them, take a specific counter-attack action which takes the current scenario into consideration. In this case, our approach could be applied as a means of reconciling γ with KB.

1.5 Reconciling Interfering Inputs

Here, we present our approach for reconciling inputs in the case of conflicts, for the case of database updates. We refer to it as *Compromising Interfering Updates*, CIU for short. In section 1.5.1, we briefly describe the basic mechanism employed by CIU. Section 1.5.3 introduces the adopted policies for reconciling conflicting updates with the underlying database under a compromised reasoning perspective.

1.5.1 CIU Basic Description

The main concern of the CIU approach is to deal with the problem of resolving conflicting updates in a database. It handles inconsistency by putting forwards a compromised policy which restricts the consequences of those updates. It allows for a maximal set of the consequences to hold, provided that they do not imply inconsistency when added to the database.

CIU can be described as a module of a reasoning system, which is invoked whenever we have conflicting updates, w.r.t. databases and to their sets of integrity constraints. Figure 1 illustrates such a system including CIU module. The arrows in the end of the links in figure 1, indicate the possible flow of data among the modules.

The way CIU appears in the context of database updates is quite straightfoward and, a priori, does not seem to have anything new to add as an original contribution to the area. However, it is the way in which CIU approaches the problem of dealing with conflicting updates that will certainly bring an original flavour to this work.

As shown in Figure 1, we assume that we have a database module D which is ruled by a module of integrity constraints I . The integrity constraints are assumed to be protected against any update modification, and they restrict the possible transactions on D. The database can be, for instance, a declarative representation of a scenario, in terms of the facts that hold in the current state. A finite set of updates, to be performed on the database, is given as input to the system. We assume that the updates executing module only effectively performs the updates in the case that they modify D without violating any integrity constraint. Otherwise, CIU module is invoked in order to perform the compromised version of the set of updates. In the end, the compromised updated database is supposed to be consistent and to satisfy the constraints in module I.

The peculiar characteristic that CIU has in dealing with updates is that, instead of preventing

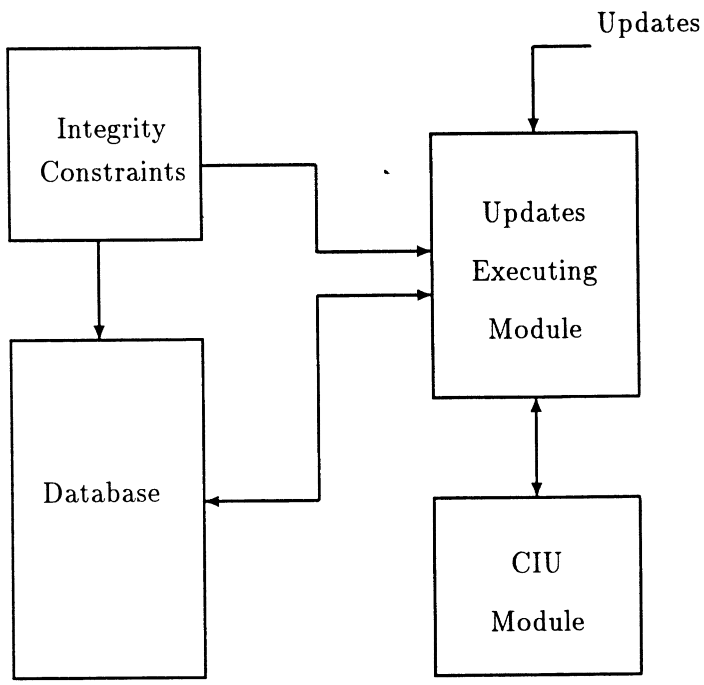


Figure 1: CIU in a Database System.

an update to be performed, when inconsistency arises[13], CIU proceeds and generates the consequences of the conflicting updates.

When CIU is invoked, it instigates the compromised performance of the conflicting updates, by firstly generating all their consequences/derivations. Later it takes care of restoring consistency in the database. In order to restore consistency, a special revision procedure takes place. It is based on the minimal elimination of the formulae involved in the generation of inconsistency, and guided by the compromised reasoning policies of the approach.

1.5.2 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 1.3 $\quad Update = \{A, \neg A\}$

•

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

Example 1.4 $\quad DB = \Delta \bigcup P_\Delta$
$P_\Delta = \{ A \rightarrow \bot\}$
$Update = \{A\}$

•

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

Example 1.5 $\quad DB = \Delta \bigcup P_\Delta$
$P_\Delta = \{ A \wedge C \rightarrow \bot\}$
$\Delta = \{ C\}$
$Update = \{A\}$

•

[13] This would make the approach equivalent to many existing ones which do not allow for updates to be performed if they are not consistent with the theory.

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

Example 1.6 $DB = \Delta \bigcup P_{\Delta}$
$\Delta = \{ A\}$
$Update = \{\neg A\}$

•

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.

1.5.3 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 [14], 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. [15]
- 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 $\neg A$ [16], 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, $\{U_i\} \bigcup I \vdash \bot$, then the update U_i is rejected, however its consistent consequences are allowed to remain as non-supported consequences in the database.

[14]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 $\alpha \rightarrow \bot$, 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.

[15]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*.

[16]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 $\{U_i\} \bigcup I \nvdash \perp$ anf $\{U_j\} \bigcup I \nvdash \perp$, 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[17]. 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[18]. 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

[17]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.

[18]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].

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 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 next part of this paper, introduces a formalization for the compromised revision of our approach, within the belief revision perspective.

2 Compromised Revision for Conflicting Updates

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]. But unlike in the AGM approach, 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 subject to a protected subset of integrity constraints. Some postulates for compromised revision are established, and a revision function for bases with integrity constraints is specified.

As a background introduction, the next section reviews the basic notions of the area of Belief Revision.

2.1 Belief Revision

Basically, *Belief Revision* is concerned with the problem of modelling the dynamics of the knowledge and beliefs of an agent [Gär-88]. In a computational setting, Belief Revision faces three main methodological issues:

- The representation of the beliefs in the database;
- The definition of the relation between the elements explicitly represented in the database and the beliefs that may be derived from them; and
- The choice about what to retract, in order to avoid inconsistency.

Belief Changes can be classified into three main basic types, namely *"Expansion"*, *"Revision"*, and *"Contraction"*. Considering a belief system K, in which beliefs are represented by sentences, an expansion of K occurs when a new sentence and its logical consequences are added to K. We say that a belief system K is revised by a sentence, say A, when A is inconsistent with K and is added to it. So, in order that the resulting belief system be consistent, the revision operation takes care of deleting some of the old sentences of K. Finally, contraction of a belief system K by a sentence A indicates that A is retracted from K, without any addition of new sentences. However, in this case, some other sentences in K might have to be given up, in order to keep the resulting belief system consistent.

There are two methods to follow, when dealing with the problem of belief revisions and contractions. One is to present explicit constructions of the revision proccess; while the other refers to the formulation of postulates for such constructions. According to [GäMa-88], the postulates should be viewed as dynamic integrity constraints or transition laws.

In the sections to follow we make a review of the way Alchourrón, Gärdenfors, and Makinson defined their approach to Belief Revision, the so-called AGM Theory. For more details refer to [AlMa-82],[AGM-85], and [Gär-88].

The language considered throughout this section survey is based on first order logic[19], and is denoted by L. The underlying logic includes classical propositional logic and is compact, that means that a set of formulae has a model in this logic if and only if every finite subset of this set of formulae also has a model. A **belief system** K is defined as a set of sentences of L. $K \vdash A$ denotes that system K logically entails A. It is also assumed that $\vdash$ satisfies "disjunction in the premises", i.e. that $K \bigcup \{B \vee C\} \vdash A$ whenever both $K \bigcup \{B\} \vdash A$ and $K \bigcup \{C\} \vdash A$. A **belief set** is defined as a set K of sentences of L which satisfies the following integrity constraint: If $K \vdash A$ then $A \in K$. The only inconsistent belief set, under this definition, is the set of all sentences in L, denoted by $K_{\perp}$, since by classical logic, whenever K is inconsistent, then $K \vdash A$ for every sentence A in L. Belief sets are used in the AGM theory as models of belief states. A **belief state** is a representation of the beliefs and knowledge of an agent at a certain time point. A belief set is required to be closed under the consequence relation. $Cn(K)$ 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 $K_1 \subseteq K_2$. The consequence operation Cn also includes tautological implication, and is compact. A **base** for a belief set K, denoted as B_K, is defined as a finite subset of K such that $Cn(B_K) = K$. B_K is said to be a base for K iff B_K is a finite subset of K and $Cn(K) = K$.

2.1.1 The AGM Rationality Postulates

In this section, we show the sets of postulates for the three main epistemic (belief) changes, proposed by Alchourrón, Gärdenfors, and Makinson [AGM-85], and consequently named after them: *the AGM postulates*.

Expansions

Expansion is supposed to be the simplest case of belief change, since it deals with the addition of a new sentence together with its logical consequences to the belief set. The expansion operator is denoted as " + " and $K + A$ denotes the result of expanding the belief set K with the sentence A. Expansions can be defined as the logical closure of K together with A:

$$K + A = \{B \mid K \bigcup \{A\} \vdash B\}$$

The AGM Postulates for Expansions

$(K^{+}1)$ $K + A$ is a belief set.

[19] L is closed under applications of the boolean operators, namely $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), and $\rightarrow$ (implication).

(K^+2) $A \in K+A$.

(K^+3) $K \subseteq K+A$.

(K^+4) If $A \in K$ then $K+A = K$.

(K^+5) If $K \subseteq H$ then $K+A \subseteq H+A$.

(K^+6) For all belief sets K and all sentences A, $K+A$ is the smallest belief set that satisfies (K^+1)-(K^+5).

Notes: By the postulate (K^+1), the resulting expanded set is guaranteed to be a belief set, and by postulate (K^+2), we have that the new sentence is also part of the expanded set. Postulate (K^+3) guarantees that the original set of beliefs does not loose any element with the expansion operation. Postulate (K^+4) states the vacuity of the expansion operation, in the case that the sentence to be added to the belief set is already an element of the set. Postulate (K^+5) guarantees the monotonicity property between expansions of belief sets. Finally, postulate (K^+6) ensures that expansions satisfy the minimal change notion, by stating that only the information which is relevant is added to the original set during an expansion operation.

Revisions

Revision is not considered to be a simple case of belief change, because, unlike expansion, the sentence to revise the belief set is supposed to contradict some of the sentences 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. The revision operator is denoted as " $*$ " and $K * A$ denotes the result of revising the belief set K with the sentence A. The set of postulates for revisions is shown below.

The AGM Postulates for Revisions

(K^*1) $K * A$ is a belief set.

(K^*2) $A \in K * A$.

(K^*3) $K * A \subseteq K + A$.

(K^*4) If $\neg A \notin K$ then $K + A \subseteq K * A$.

(K^*5) $K * A = K_\perp$ if and only if $\vdash \neg A$.

(K^*6) If $\vdash A \leftrightarrow B$ then $K * A = K * B$.

(K^*7) $K * A\&B \subseteq (K * A) + B$.

(K^*8) If $\neg B \notin K * A$ then $(K * A) + B \subseteq K * A\&B$.

Notes: By the postulate (K^*1), the resulting revised set is guaranteed to be a belief set, and by postulate (K^*2), we have that the sentence revised is also part of the resulting set. Postulate (K^*3) states that the revised set of beliefs is at least a subset of the expanded set, when both operations use the same sentence. Postulate (K^*4) caters for the case in which no old belief has to be given up from the original belief set. Postulate (K^*5) guarantees that the revised set is consistent, unless the negation of the sentence to be revised is a tautology. Postulate (K^*6) ensures the equality between revisions of logically equivalent sentences, considering the same original set of beliefs. (K^*1)-(K^*6) represents the basic set of postulates for revision. The postulates (K^*7) and (K^*8) are extra conditions for "composite" belief revision, and they are analogous to postulates (K^*3) and (K^*4) in the "single" case.

Contractions

In contractions, we are supposed to give up the belief in the specified sentence. However, we may well have to contract the original set by other sentences, in the case that the specified sentence to be contracted is supported (implied) by them. The main problem that contractions have to face, and consequently revisions as well, is that the way to achieve a contraction may not be unique. The set of postulates for contractions, shown below, just provide a minimum set of requirements that a contraction function should satisfy, and it does not uniquely determine a contraction function. Hence, the problem of choosing which sentences to be retracted has to be tackled by other means[20].

The contraction operator is denoted as " $-$ " and $K - A$ denotes the result of contracting the belief set K by the sentence A.

The AGM Postulates for Contractions

(K^-1) $K - A$ is a belief set.

(K^-2) $K - A \subseteq K$.

(K^-3) If $A \notin K$ then $K - A = K$.

(K^-4) If not $\vdash A$ then $A \notin K - A$.

(K^-5) If $A \in K$ then $K \subseteq (K - A) + A$.

(K^-6) If $\vdash A \leftrightarrow B$ then $K - A = K - B$.

(K^-7) $K - A \bigcap K - B \subseteq K - A\&B$.

(K^-8) If $A \notin K - A\&B$ then $K - A\&B \subseteq K - B$.

Notes:

By the postulate (K^-1), the resulting contracted set is guaranteed to be a belief set. Postulate (K^-2) claims that the resulting contracted set of beliefs is at least a subset of the original set. Postulate (K^-3) states the vacuity of the contraction operation, in the case that the sentence to be deleted in the belief set, is not an element of the set. Postulate (K^-4) indicates the success condition of the contraction, which states that if the sentence to be removed is not a logically valid one, then it is not present in the resulting set of beliefs. Postulate (K^-5) states the recovery of the original set of beliefs. The contracted theory should contain enough information to recover all sentences deleted. Postulate (K^-6) ensures the equality between contractions of logically equivalent sentences, w.r.t. the same original set of beliefs. (K^-1)-(K^-6) represents

[20] *Epistemic Entrenchment*, for instance, is a proposed way to deal with this problem. More details about it is found in [GäMa-88], and also summarized further in this appendix.

the basic set of postulates for contractions. As for the revision postulates, the postulates ($K^{-}7$) and ($K^{-}8$) are extra conditions for "composite" contraction, and they are analogous to postulates ($K^{-}3$) and ($K^{-}4$) in the "single" case.

The notion of minimal change of the belief set, when having to retract a sentence from it, was the underlying motivating idea for the contraction postulates. This notion is based on the *Conservativity Principle*, which states that if a change in the belief set has to be made in order to accomodate a new concept, then as much as possible of the old beliefs should be maintained.

2.1.2 Contraction Functions

Contraction as well as revision cannot be expressed as simple operations, as they have no obvious unique results. When trying to remove a sentence A from a closed theory K, there are usually many maximal subsets[21] of K which fail to imply A. The same holds for revision.

In the AGM work, in order to approach the problem of finding intuitive and plausible theory-change operations, they define some contraction functions which are then tested against the proposed rationality postulates. Further in this section, we present the definitions of some of those contraction functions. Notice that only contraction functions are defined, since revision can be obtained via contraction, and vice-versa.

The relation between revision and contraction can be explicitly shown by the so-called *Levi Identity* [Lev-77], presented below, in which a revision of a knowledge set can be seen as a composition of a contraction and an expansion. The Levi Identity is motivated by the strategy of accomodating the new belief, by retracting from the initial belief set all the old beliefs that contradict it.

$$\text{Levi Identity} \qquad K * A \;=\; (K - \neg A) + A$$

Contractions can also be defined in terms of revisions. This view was proposed by Harper [Har-77], known as *Harper Identity*, and motivated by the claim that since $\neg A$ is a logical consequence of $K * \neg A$, then $K - A$ should retain only the beliefs which are both elements of K and $K * \neg A$.

$$\text{Harper Identity} \qquad K - A \;=\; K \bigcap (K * \neg A)$$

Maxichoice Contraction

Maxichoice contraction functions are the ones which identify $K - A$ with one of the maximal subsets in $K \bot A$, where $K \bot A$ denotes the set of all belief sets that fail to imply A. This function relies on a selection function S that picks out an element $S(K \bot A)$ of $K \bot A$, for any K and any A whenever $K \bot A$ is nonempty.

[21] The notion of maximal subset for contraction defines a belief set K' as a maximal subset of K, which fails to imply A and which satisfies the following conditions: 1. $K' \subseteq K$; 2. $A \notin K'$; and 3. for any sentence B that is in K but not in K', $B \rightarrow A$ is in K'.

Given a belief set K, a sentence A and a selection function S, a maxichoice contraction, $K - A$, is obtained as follows:

$$\begin{array}{lll} K - A = S(K \perp A) & \text{if not } \vdash A\text{; and} \\ K - A = K & \text{otherwise.} \end{array}$$

Any maxchoice contraction function satisfies the postulates $(K^{-}1)$ to $(K^{-}6)$. It also satisfies the following fullness condition, denoted by $(K^{-}F)$:

$$(K^{-}F) \quad \text{If } B \in K \text{ and } B \notin K - A, \text{ then } B \rightarrow A \in K - A \text{ for any belief set } K.$$

From the condition above, we can see that contractions produced by the maxichoice contraction function are in general too large, since K is maximal. This implies that either B or $\neg B$ is in K, for any B in the language.

Full Meet Contraction

The full meet contraction function is the one which assumes that $K - A$ contains only the propositions that are common to all the maximal subsets in $K \perp A$.

Given a belief set K and a sentence A, a full meet contraction, $K - A$, is obtained as follows:

$$\begin{array}{lll} K - A = \bigcap (K \perp A) & \text{whenever not } \vdash A \text{ and } K \perp A \text{ is nonempty; and} \\ K - A = K & \text{otherwise.} \end{array}$$

Any full meet contraction function satisfies the postulates $(K^{-}1)$ to $(K^{-}6)$. It also satisfies the following intersection condition, denoted by $(K^{-}I)$:

$$(K^{-}I) \quad \text{For all } A \text{ and } B, \; K - A\&B = K - A \bigcap K - B.$$

Notice that, due to the intersection condition, contractions produced by the full meet contraction function results in sets that are far too small. Futhermore, when retracting A from K with a full meet contraction operation, we are left with the sentences of K which are already consequences of $\neg A$. If this result is carried over to theory revision, then we have that if $\neg A \in K$ and if not $\vdash \neg A$ then $K * A = Cn(\{A\})$, which is clearly unintuitive.

Partial Meet Contraction

The partial meet contraction function defines $K - A$ by investigating the consequences of using some of the maximal subsets in $K \perp A$. This function uses a selection function S which this time is assumed to pick out a nonempty subset of $K \perp A$, denoted by $S(K \perp A)$, and not a singleton as in the maxichoice contraction. In the case that $K \perp A$ is empty, then $S(K \perp A) = K$.

Given a belief set K, a sentence A and a selection function S, a partial meet contraction, $K - A$, is obtained as follows:

$$\begin{array}{rcll} K - A & = & \bigcap S(K\bot A) & \text{if not } \vdash A\text{; and} \\ K - A & = & K & \text{otherwise.} \end{array}$$

Any partial meet contraction function satisfies the postulates $(K^{-}1)$ to $(K^{-}6)$.

Notice that the idea of the selection function S picking out the best elements of $K\bot A$ can be made more precise by assuming that there is an *ordering* of the maximal subsets in $K\bot A$, that is used in order to select the top elements. We see this notion in the contraction function defined below.

Transitively Relational Partial Meet Contraction

This contraction function assumes a transitive and reflexive ordering relation on the union of all the maximal subsets in $K\bot A$, that is used to pick out top elements from that set.

Assume that $M(K)$ is the union of the family of all the sets in $K\bot A$. Assume also that there is a transitive and reflexive ordering relation, $\leq$, operating on $M(K)$. When $K\bot A$ is non-empty, this relation defines a selection function, denoted by $\mathcal{S}(K\bot A)$, that picks out the top elements in the ordering, as follows:

$$\mathcal{S}(K\bot A) \;=\; K' \in K\bot A \mid K'' \leq K' \quad \text{for all } K'' \in K\bot A$$

Given a belief set K, a sentence A and a selection function $\mathcal{S}$, as above, we have that the transitively relational partial meet contraction, $K - A$, is obtained as follows:

$$\begin{array}{rcll} K - A & = & \bigcap \mathcal{S}(K\bot A) & \text{if not } \vdash A\text{; and} \\ K - A & = & K & \text{otherwise.} \end{array}$$

Any transitively relational partial meet contraction function satisfies the postulates $(K^{-}1)$ to $(K^{-}8)$.

Relational Partial Meet Contraction

A relational partial meet contraction generalizes the notion of a partial meet contraction which involves a selection function defined by an ordering relation. The definition of the previous contraction function is also an example of a relational partial meet contraction, in the case that the ordering relation is transitive.

2.1.3 Epistemic Entrenchment

The notion of epistemic entrenchment was introduced within the AGM theory[22], in order to express that some of the sentences in the belief set have greater informative or explanatory power than others. Intuitively, this notion has the meaning that the more epistemically entrenched a sentence is, the harder it will be to get rid of it during contraction operations. Hence, epistemic entrenchment gives another way of modelling contractions.

When a belief set K is contracted or revised, the underlying idea for constructing a contraction function is that the sentences in K that are given up are those having the lowest degrees of epistemic entrenchment. The notation $A \leq B$ is used to express that B is at least as epistemic entrenched as A. This indicates that if we have to choose between A and B to be retracted from a belief set, then A would be deleted. It is relevant to notice that $\leq$ is only defined w.r.t. a given knowledge set K. Different knowledge sets may be associated with different orderings of epistemic entrenchment.

Postulates for Epistemic Entrenchment

($EE1$) If $A \leq B$ and $B \leq C$, then $A \leq C$.

($EE2$) If $A \vdash B$, then $A \leq B$.

($EE3$) For any A and B, $A \leq A\&B$ or $B \leq A\&B$.

($EE4$) When $K \neq K_{\perp}$, $A \notin K$ iff $A \leq B$ for all B.

($EE5$) If $B \leq A$ for all B, then $\vdash A$.

Notes: The postulate ($EE1$) states the transitivity property of the epistemic entrenchment ordering relation. Postulate ($EE2$) expresses the dominance between two sentences of a belief set, when one entails the other. For instance, in the case of retracting B, when $A \vdash B$, A would have to be retracted as well. Postulate ($EE3$) states that any conjunction is at least as epistemically entrenched as any of its components. This determines that only one of the conjunction components should be taken away, in order to obtain the retraction of the whole conjunction. And finally, postulates ($EE4$) and ($EE5$) indicate respectively the minimality and the maximality conditions of the epistemic entrenchment ordering relation.

Consequences of the EE Postulates

An ordering relation that satisfies the postulates ($EE1$) to ($EE5$), also has the properties below:

(i) $A \leq B$ or $B \leq A$.

(ii) If $B\&C \leq A$, then $B \leq A$ or $C \leq A$.

(iii) $A < B$ iff $A\&B < B$.

[22] See [GäMa-88] for more details on Epistemic Entrenchment.

(iv) If $C \leq A$ and $C \leq B$ then $C \leq A\&B$.

(v) If $A \leq B$ then $A \leq A\&B$.

Where $A < B$ denotes the strict relation, defined as " $A \leq B$ and not $B \leq A$ ", representing that B is epistemically more entrenched tan A.

Connections between EE, Contraction and Revision

Below we present two conditions, from [GäMa-88], which show the connections between epistemic entrenchment and contraction and revision.

$(C \leq)$ $A \leq B$ iff $A \notin K - A\&B$ or $\vdash A\&B$.

$(C-)$ $B \in K - A$ iff $B \in K$ and either $A < A \vee B$, or $\vdash A$.

Notes:

$(C \leq)$ assures that when we contract K w.r.t. $A\&B$ we give up A or B, minimally speaking. Moreover, A should be retracted just in case B is at least epistemically entrenched as A. The limiting case for this condition, is when both A and B are logically valid, because then, they are of equal epistemic entrenchment. $(C-)$ gives an explicit definition of a contraction function in terms of the epistemic entrenchment relation.

2.1.4 Discussions

The work of AGM proposes a formalism for theory changes, which provides important insights to belief revision. However, it does not seem to be adequate to be used in a computational setting. Moreover, it shows some drawbacks w.r.t. the contraction functions. We list below some of the problems encountered in the proposed formalism:

- A belief state is presented as a closed theory, and closed theories are generally far too large to be dealt with in a computational context;
- The formalism, in general, does not allow for iterations. Considerations about what is kept by the revision process, in the next theory, is only possible when we know exactly what the beliefs are in the initial theory presentation.
- The problem of what to choose to give away, when contracting a belief set, is not well resolved. By using the contraction functions, in many cases we do not know exactly what sentences should be kept in the revised theory. Moreover, we either end up with a set which is too large and full of unwanted information or too small, or alternatively, we transfer the choice problem of which sentences to be kept in the revised theory to an ordering-relation-based selection function.

- In [GäMa-88], they show, by means of some theorem results relating epistemic entrenchment with the conditions ($C \leq$) and ($C-$), and with contraction, that the problem of constructing appropriate contraction and revision functions can be reduced to the problem of providing an appropriate ordering to epistemic entrenchment. However, the task of specifying which sentence in a theory is more relevant than the other, is not an easy one. Hence, the epistemic entrenchment ordering is difficult to be determined, besides having to be regenerated after each occurrance of a belief change.

It then becomes clear that the AGM theory provides general guidelines about the behaviour of belief change processes. However, it does not specify much about how one should build effective functions which correspond exactly to such processes.

A solution for dealing with closed theories, would be to have a finite representation. However, in the general case, there seems to be no trivial ways to derive finite representations from a revised theory. The application of the conservativity principle in the revision process, seems to be a debatable issue, since it causes revised theory sets to be too full of unwanted information. A solution to this problem would be to distinguish within the formalism, between an information or belief base, and a belief set. This way, closed derivations would apply only to the belief base, and not to the belief set. This approach would give more informational power to the application of the conservativity principle, and would consequently decrease the number of sentences which should be kept in the resulting revised set. The subject of modification of finite theory bases, instead of closed theories, is dealt with in the next section.

Some proposed solutions to the problem of iteration within the revision process, have already emerged in the literature. In [FrLe-94], for instance, they introduce new postulates to deal with iterated revisions.

We find in the literature some proposed alternatives to the problem of the epistemic entrenchment ordering. In [Rya-92], for instance, they introduce the concept of ordered theory presentations, which embeds an ordering within the way the theory is presented, and automatically updates that ordering whenever a theory revision takes place.

2.2 Theory Changes x Finite Base Changes

Finite base changes seem to be an interesting alternative to the AGM work of theory changes. In base revision, as opposed to what happens in theory revision, it is simple to accomplish the situation in which we retract a sentence from the base, getting rid of its consequences as well.

As expected, applications such as database updates and conterfactual reasoning, perform revision and contraction operations on finite bases, and not on closed theories. Basically, for those applications, the contraction operation determines the family of maximal subsets of the finite base, which do not imply the sentence retracted.

Base Contraction Function (1)

Assume that B is a finite base, and A is a sentence. Then, a base contraction operation, denoted by $B \sim A$, can be obtained as follows:

$$
\begin{aligned}
B \sim A &= \textstyle\bigcap_{C \in (B \perp A)} Cn(C) && \text{if } A \notin Cn(\emptyset) \text{ ; and} \\
B \sim A &= Cn(B) && \text{otherwise.}
\end{aligned}
$$

Where $B \perp A$ denotes the family of maximal subsets of the base B, not implying A.

The base contraction operation, as defined above, does not satisfy the recovery postulate (K^-5), for theory contraction. This result comes from the fact that $B \sim A$ is not a partial meet contraction on closed theories. That means that, in general, we do not have:

$$B \sim A \subseteq Cn(B) \bigcap Cn(\neg A)$$

Nevertheless, in [Neb-90], they show that base contraction is a special case of theory contraction, via a theorem which states the equivalence between base and partial meet contraction for closed theories. In order to obtain this result, they define a new base contraction operation by adding $Cn(B) \bigcap Cn(\neg A)$ to the right side of the one defined in ??, as shown below. They also specify a choice function to select the elements of $Cn(B) \perp A$ containing maximal subsets of *relevant*[23] sentences.

Base Contraction Function (2)

Given a finite base B, and a sentence A, a base contraction operation, denoted by $B \simeq A$, can be obtained as follows:

$$\begin{array}{lll} B \simeq A = & Cn((\bigcap_{C \in (B \perp A)} Cn(C) \bigcup (Cn(B) \bigcap Cn(\neg A))) & \text{if } A \notin Cn(\emptyset) \text{ ; and} \\ B \simeq A = & Cn(B) & \text{otherwise.} \end{array}$$

In [Mak-87], they show that $\simeq$ and $\sim$ are revision-equivalent operations.

Selection Function $\mathcal{S}_B$

Assume that $Cn(B) \perp A$ contains all the maximal subsets of B, not implying A. A selection function, denoted by $\mathcal{S}_B$, applied to the set $(Cn(B) \perp A)$, is obtained as follows:

$$\mathcal{S}_B(Cn(B) \perp A) = C \in (Cn(B) \perp A) \mid \forall C' \in (Cn(B) \perp A) \; C' \bigcap B \not\supset C \bigcap B$$

Using the selection function defined above, a partial meet contraction is defined on $Cn(B)$, and with this definition the equivalence theorem between $Cn(B) - A$ and $B \simeq A$, is stated and proved in [Neb-90].

2.2.1 Epistemic Relevance

In order to modify a knowledge base, we have to decide which sentences are more important, i.e., which sentences are epistemically relevant w.r.t. the basis of our body of knowledge.

In the AGM theory, the notion of Epistemic Entrenchment respects the logical force of sentences within the theory, and is strictly restricted to the contraction operations. For instance, consider the principle below, which is derived from the Epistemic Entrenchment Postulates, about degrees of entrenchment, (see [GäMa-88] for more details).

$$B \in K - A \text{ iff } (A \vee \neg B) <_n (A \vee B) \text{ or } B \in Cn(\emptyset)$$

Where A and B are propositions, and $A <_n B$ means that A is epistemically less entrenched than B with entrenchment degree n.

[23] The notion of relevance for base elements, is similar to the notion of entrenchment for theory elements. More details about epistemic relevance are given in section 2.2.1.

In Nebel's notion of Epistemic Relevance [Neb-89], the marked sentences within the base, are the ones which are considered somehow crucial, regardless of their logical relations. Moreover, it is not even necessary to refer to a theory base, but only choose the epistemic relevant sentences. Epistemic Relevance does not allow for the derivation of the principle above.

Within the notion of epistemic relevance, one might not only distinguish between sentences which are in the base and those others that are not, but also one may assign multiple degrees of relevance to sentences in order to distinguish between, for instance, simple facts and integrity constraints in a logical database.

2.2.2 Some Issues on Finite Base Contractions

Contracted bases represented by a finite axiomatization

In the definitions of the base contraction functions presented previously, we get as a result a closed theory. This is definitely not what we wanted. We wanted a finite representation of that result. However, if we use the properties of Cn, we can derive a finite representation of $B \simeq A$, as follows:

$$\begin{aligned} B \simeq A &= Cn((\bigcap_{C \in (B \perp A)} Cn(C)) \bigcup (Cn(B) \bigcap Cn(\neg A))) \\ &= Cn((\bigvee_{C \in (B \perp A)} C) \wedge (B \vee \neg A)) \\ &= Cn((\bigvee_{C \in (B \perp A)} C \wedge \neg A) \vee B) \end{aligned}$$

Hence, assuming that B is finite, $B \simeq A$ can be represented by a finite disjunction. This approach provides a solution for representing contracted bases by finite axiomatizations.

New sets of sentences in contracted bases

For some application areas, like counterfactual reasoning, where only one revision operation is needed to obtain the desired result, this problem is not of any importance. However, for logical database updates for instance, which might require to deal with a long sequence of change operations, we do not want to view a changed theory as just one sentence, like the one shown above. Otherwise, it could be retracted completely by the next contraction operation. Among the solutions to this problem, we find in the literature an attractive one in [FKUV-88], where they view a changed theory as a collection of flocks, which are alternative theories that can be revised further.

As in the original approach to belief revision, an underlying problem to finding solutions for representing a changed theory base, is the point of how to determine the new set of epistemic relevant sentences in a changed theory base. It seems then that this problem was not solved by the consideration of theory bases rather than closed theories.

Computational limitations of finite base contractions

If we consider first-order-predicate logic, then we can not go very far, since consistency is undecidable. However, for most applications, we consider restricted formalisms which make the situation become more feasible. Nevertheless the problem is still difficult. If no simplication method is applied, the base can grow exponentially after a certain number of change operations. Even in propositional logic the problem is NP-complete.

2.3 A Compromised Characterization to Revision

It has already been 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 presentation, due to our interest in simulating changes of the real world, which are always done locally within a theory presentation. 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 presentations which are constrained by a protected set of sentences are also considered in this characterization. Such protected parts 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 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.[24]

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, not 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.

2.3.1 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.

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

[24] 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 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.

2.3.2 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. So, if an input causes a base to become inconsistent, then the following steps are taken.

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

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

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

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

•

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 of 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:

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

•

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 large 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, i.e. it is not necessarily a subset of the closed theory.

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

2.3.3 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. ($\alpha \vdash \beta$ iff $\vdash \alpha \rightarrow \beta$.)

$K \vdash \alpha$ 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. $K \nvdash \bot$.

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

$K_\bot$ 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 $\circledast$ is a version of the AGM revision operation $*$, (see appendix A), adapted for theory-bases.

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

2.3.4 Finite Bases Revision Postulates

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.

($K^{\circledast}1$) $K \circledast \alpha$ is a base set.

($K^{\circledast}2$) $\alpha \in K \circledast \alpha$.

($K^{\circledast}3$) $K \circledast \alpha \subseteq K \bigcup \{\alpha\}$.

($K^{\circledast}4$) If $K \nvdash \neg\alpha$ then $K \bigcup \{\alpha\} = K \circledast \alpha$.

($K^{\circledast}5$) $K \circledast \alpha \vdash \bot$ if and only if $\vdash \neg\alpha$.

($K^{\circledast}6$) If $\vdash \alpha \leftrightarrow \beta$ then $Cn(K \circledast \alpha) = Cn(K \circledast \beta)$.

($K^{\circledast}7$) $K \circledast \alpha\&\beta \subseteq (K \circledast \alpha) + \beta$.

($K^{\circledast}8$) 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 ($K^{\circledast}1$), the resulting revised set is guaranteed to be a base set. For the case of closed-theories such a postulate makes sense. However, for non-closed belief bases, it is not so relevant. By postulate ($K^{\circledast}2$), we have that the sentence revised is also part of the resulting set. Postulate ($K^{\circledast}3$) 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 ($K^{\circledast}4$) caters for the case in which no old belief has to be given up from the original base set. Postulates ($K^{\circledast}3$) and ($K^{\circledast}4$) 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 ($K^{\circledast}5$) guarantees that the revised set is consistent, unless in the case that α is an inconsistent sentence. Postulate ($K^{\circledast}6$) ensures the equality between the closure of the revisions of logically equivalent sentences, considering the same original set of beliefs. ($K^{\circledast}1$)-($K^{\circledast}6$) represents the basic set of postulates for base revision. The postulates ($K^{\circledast}7$) and ($K^{\circledast}8$) are extra conditions for "composite" belief revision, like the postulates (K^*7) and (K^*8), for the case of revision for closed theories.

2.3.5 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 " $\otimes$ " the operator for compromised revision for conventional bases. We define the postulates ($\otimes$1)-($\otimes$9) as the basic requirements for achieving the new set $K \otimes \alpha$, which is the result from the compromised revision of the knowledge base K by α. So, $K \otimes \alpha$ is assumed to satisfy the postulates ($\otimes$1)-($\otimes$9).

In the postulates ($\otimes$1)-($\otimes$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 $K \otimes \alpha$, 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 $K \otimes \alpha$.
- 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

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

($\otimes$2) If $\nvdash \neg\alpha$, then $\alpha \in K \otimes \alpha$.

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

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

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

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

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

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

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

Notes:

Postulates ($\otimes$1) and ($\otimes$2) are equivalent to the postulates ($K^{\circledast}1$) and ($K^{\circledast}2$), for AGM base revision.

The postulate ($\otimes$3) states the vacuity of the compromised revision.

Postulate ($\otimes$4) states that compromised revision includes the AGM base-revision, in the case that the sentence α does not contradict any sentences in K. This postulate is also equivalent to postulate ($K^{\circledast}4$).

Postulate ($\otimes$5) caters for the case in which the revision-input sentence contradicts some sentences of the knowledge base K. ($\otimes$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 ($\otimes$6) and ($\otimes$7) cater for consistency in $K \otimes \alpha$. ($\otimes$6) guarantees that if the input sentence contradicts a tautology of the logical system, then no change is made on the original set K. And postulate ($\otimes$7) claims that the resulting compromisingly-revised base is always consistent.

Postulate ($\otimes$8) is equivalent to postulate ($K^{\circledast}7$). It guarantees the equality between the closure of the compromised revisions of logically equivalent sentences, considering the same original sets.

Finally, postulate ($\otimes$9) introduces the *maximality* condition for $K \otimes \alpha$. 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.

($\otimes$1)-($\otimes$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 ($K^{\circledast}3$) is not satisfied in compromised revision, since we might add to $K \otimes \alpha$ some consequences of retracted sentences which do not belong to $K \cup \{\alpha\}$.

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 ($K^{\circledast}5$) and ($\otimes$7), respectively.

Postulate ($K^{\circledast}7$) is not satisfied in compromised revision, since we might add to $K \otimes \alpha\&\beta$ some consequences of retracted sentences which do not belong to $(K \otimes \{\alpha\}) + \beta$. For the same reason, ($K^{\circledast}8$) is also not satisfied in compromised revision. We cannot guarantee that the consequences of the deleted old sentences of K added to $K \otimes \alpha$ are present in $Cn(K \otimes \alpha\&\beta)$.

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

2.3.6 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: $\alpha \notin CI(\alpha)$;
  $CI(\alpha) \bigcup \Delta_K \nvdash \bot$ and $CI(\alpha) \bigcup \Delta_K \nvdash \alpha$.
- $MaxCI(\alpha)$ is a maximal consistent subset of $CI(\alpha)$ w.r.t. K, such that $MaxCI(\alpha) \bigcup K \nvdash \bot$.
- 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 $MaxCR(R_\alpha) \bigcup (\Delta_K - R_\alpha) \bigcup P_K \bigcup \{\alpha\} \nvdash \bot$.
- 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.

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

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

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

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

(®6) $P_K \subseteq K \circledR \alpha$.

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

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

(®9) For all knowledge bases K and for all sentences α, $K \circledR \alpha$ is the largest set that satisfies (®1)-(®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 (®1) states that the resulting revised knowledge base has the same structure of the original base.

Postulate (®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 (®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 $K \circledR \alpha = K \bigcup \{\alpha\}$. (®3) is the *inclusion* postulate of our compromised revision.

Postulate (®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. (®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 (®4) as the *success* postulate.

Postulate (®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. (®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 (®5) as the *compromise* postulate.

(®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.

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

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

Finally, postulate (®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. (®9) introduces the *maximality* conditions for $K \circledR \alpha$.

(®1)-(®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.

2.4 Towards the Formulation of the Compromised Revision Function

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 $K \circledR \alpha$, requires a proper definition. One way to establish this notion, can be defined by applying 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 $K \circledR \alpha$.

The strategy described above resembles the notion of *safe contraction/revision* introduced in [Mak-85], [AlMa-85]. In safe contraction, an ordering ($<$) which is irreflexive and transitive, is defined on K. In general terms, if α is the sentence to revise K, then $\neg\alpha$ should be the sentence that one wishes to eliminate from among the consequences of K. So, an element x of K is said to be *safe* w.r.t. $\neg\alpha$ (modulo $<$), if and only if every minimal subset M of K that implies $\neg\alpha$ either does not contain x, or contains at least one element y that is less relevant, or worse, than x w.r.t. the ordering, $y < x$. In [Mak-85], they write as $K/\neg\alpha$ for the set of all elements of K that are safe w.r.t. $\neg\alpha$, modulo $<$. and they define the safe contraction $K \dot{-} \neg\alpha$ as the set of all elements of K that are implied by $K/\neg\alpha$, $K \dot{-} \neg\alpha = Cn(K/\neg\alpha) \bigcap K$.

In the case of a compromised revision $K \circledR \alpha$ 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 $\bot_\alpha$, which includes all the minimal subsets of Δ_K, with those elements.[25]

Moreover, we want to be able to perform the necessary retractions, satisfying the conservativity principle. We then define the set R_α, taking into account the minimals of the subsets of $\bot_\alpha$, w.r.t. the ordering of Δ_K.[26] R_α selects from the sets of $\bot_\alpha$ the minimal elements which should be retracted from Δ_K.

Hence, a safe-contraction-like approach can be considered, for each element β in the set R_α. Furthermore, according to our compromising philosophy, we have to allow for all the consistent consequences of the elements of R_α to be kept in the resulting base.[27]

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 $K = \langle \Delta_K, P_K \rangle$.

[25] See Definition 2.11.

[26] See Definitions 2.1 and 2.13.

[27] The set $MaxCR(R_\alpha)_\alpha$, already used in the postulates and precisely defined further in this section, comprises this notion.

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 an atomic 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* $\alpha \in P_K$*. We also sometimes write* $\Delta_K \subseteq K$ *and* $P_K \subseteq K$*, by abuse of notation.*

•

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 those elements are mathematically incomparable w.r.t. their relevance levels. 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.

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* $\leqslant$*, let the ordering on the consequences of* X *be such that if* $Z \vdash y$ *and* $y \notin X$*, for any set* $Z \subseteq X$ *such that* Z *is minimal w.r.t.* $\subseteq$*, then* $x \leqslant y$*,* $\forall x \in Z$*.*

□

Example 2.4 ($\leqslant$ **Propagation)**

Consider an ordered set $X = \{a,\ b,\ c,\ c \rightarrow d,\ a \rightarrow e\}$, *such that* $a \leqslant b$, $a \leqslant c$, $c \leqslant c \rightarrow d$, *and* $a \leqslant a \rightarrow e$. *Assume that* $Con(X) = \{a,\ b,\ c,\ c \rightarrow d,\ a \rightarrow e,\ d,\ e\}$. *Hence, the ordering on* $Con(X)$ *is such that:* $a \leqslant c$, $c \leqslant c \rightarrow d$, $a \leqslant a \rightarrow e$, $c \leqslant d$, $c \rightarrow d \leqslant d$, $a \leqslant e$, *and* $a \rightarrow e \leqslant e$.

•

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 a subset of a set* S, *which is ordered w.r.t. the partial order* $\leqslant$. *Let* $min(\mathcal{Y})$ *be the following set:*

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

□

The sets $min(\mathcal{Y})^+$ and $min(\mathcal{Y})^1$ defined below, are subsets of $min(\mathcal{Y})$, which include the minimal sets $min(S_i)$ with more than one element, and with one element, respectively.

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:*

$$min(\mathcal{Y})^+ = \{min(S_i)_j \mid min(S_i)_j \in min(\mathcal{Y}),\ \text{and } |min(S_i)_j| > 1\}.$$

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

□

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:*

$$min(\mathcal{Y})^1 = \{x \mid x \in min(S_i)_j,\ min(S_i)_j \in min(\mathcal{Y}),\ \text{and } |min(S_i)_j| = 1\}.$$

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

□

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:*

$$Min(\mathcal{Y}) = \bigcup min(S_i), \ s.t. \ min(S_i) \in min(\mathcal{Y}).$$

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. The condition c represents a property that needs to verified. As an example, c could state that X is consistent. 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.

In order to refine the construction of the set $Min(\mathcal{Y})$, given in Definition 2.6, so that less elements of original ordered set are removed from it, we propose the alternative definition below, in which we consider the common elements of the sets $min(S_i)$ in $min(\mathcal{Y})$.

Definition 2.7 (Refined $Min(\mathcal{Y})$: $RMin(\mathcal{Y})$)

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 $RMin(\mathcal{Y})$ be the following set:

$$RMin(\mathcal{Y}) = min(\mathcal{Y})^1 \cup (Min(\mathcal{Y}) - M^*) \cup CM.$$

Where: $M^* = \{y \mid y \in min^x, \forall min^x \in M\}$; $M = \{min^x \mid \forall min^x \in Min(\mathcal{Y})\}$; *and* $min^x = \{S_x \mid S_x = min(S_i), \ \forall min(S_i) \in min(\mathcal{Y}), \ s.\ t.\ x \in min(S_i), \ and\ x \in Min(\mathcal{Y})\}$.

And $CM = \{x \mid x \in Min(\mathcal{Y}), \ such\ that\ min^x \in M\}$.

□

Remark 2.2

In Definition 2.7, the set min^x contains the set-elements of $min(\mathcal{Y})$, which have the the element x in common, for an element x in $Min(\mathcal{Y})$. The set M contains all the sets min^x, for all x in $Min(\mathcal{Y})$. The set M^* is the union of all the sets min^x in M. the set CM contains the common elements x of the sets min^x in M.

•

Example 2.5

Assume that $min(\mathcal{Y}) = \{min(S_1), min(S_2), min(S_3), min(S_4), min(S_5), min(S_6), min(S_7), min(S_8), min(S_9), min(S_{10})\}$, *where* $min(S_1) = \{a\}$; $min(S_2) = \{a, d\}$; $min(S_3) = \{j\}$; $min(S_4) = \{f, g\}$; $min(S_5) = \{a, e\}$; $min(S_6) = \{f, b\}$; $min(S_7) = \{c, h\}$; $min(S_8) = \{c, i\}$; $min(S_9) = \{a, f, c\}$; *and* $min(S_{10}) = \{l, m\}$. *According to Definitions 2.5, 2.6 and 2.7, we have that:*

$min(\mathcal{Y})^{1)} = \{a, j\}$;

$Min(\mathcal{Y}) = \{a, b, c, d, e, f, g, h, i, j, l, m\}$;

$M = \{min^a, min^c, min^f\}$;

$min^a = \{min(S_1), min(S_2), min(S_5), min(S_9)\}$;

$min^c = \{min(S_7), min(S_8)\}$;

$min^f = \{min(S_4), min(S_6), min(S_9)\}$;

$M^* = \{a, b, c, d, e, f, g, h, i\}$;

$CM = \{a, c, f\}$; *and*

$RMin(\mathcal{Y}) = \{a, c, f, g, j, l, m\}$.

•

Below, we state the Definitions 2.8 and 2.9, concerning the sets $Fail(X)_c$ and $Smax(X)_c$, respectively. Where $Smax(X)_c$ is defined w.r.t. the refined version of the set $Min(Fail(X)_c)$, $RMin(Fail(X)_c)$.

Notation 2.1

In Definition 2.8, we denote as $c(X/S)$, *the fact that the set* S *substitutes the set* X *in condition* c.

•

Definition 2.8 ($Fail(X)_c$)

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:*

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

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

□

Definition 2.9 ($Smax(X)_c$)

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

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

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

□

Remark 2.3

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 the maximal consistent subset of* X *w.r.t. set-inclusion. We call* $Smax(X)_c$ *safe-maximal, because it does not choose among the minimals, w.r.t.* $\leqslant$, *of the subsets of* $Fail(X)_c$. *Instead, it discards the minimal elements, which are in* $RMin(Fail(X)_c)$, *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 the input 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 does not derive α.

Remark 2.4

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 $RMin(Fail(\Delta_K)_{c_1})$ to create $Smax(\Delta_K)_{c_1}$, as stated in Definitions 2.8, 2.7, and 2.9. We then prove that $Smax(\Delta_K)_{c_1} \bigcup\{\alpha\} \nvdash \bot$. (See Remarks 2.5, and 2.6; as well as Propositions 2.1 and 2.2).

Preliminary Definitions:

Knowledge Base $K = \langle \Delta_K, P_K \rangle$
$P_K = \{ x \mid x \in L, \text{and } x \text{ is of the form } \bigwedge_{i=1}^{i=n} \alpha_i \rightarrow \bot,$
where α_i is a proposition or its negation$\}$.
Δ_K is a partially ordered set of sentences, w.r.t. $\leqslant$.

Propagation of $\leqslant$:
If $Z \vdash y$ and $y \notin \Delta_K$, for any set $Z \subseteq \Delta_K$, such that Z is minimal w.r.t. $\subseteq$, then $x \leqslant y$, $\forall x \in Z$.

For any set $\mathcal{Y} = \{S_1, S_2, \cdots, S_n\}$, where each S_i, for $i = 1, \cdots, n$, is ordered w.r.t. $\leqslant$.

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

$min(\mathcal{Y})^+ = \{min(S_i)_j \mid min(S_i)_j \in min(\mathcal{Y}), \text{ and } |min(S_i)_j| > 1\}$.

$min(\mathcal{Y})^1 = \{x \mid x \in min(S_i)_j, \ min(S_i)_j \in min(\mathcal{Y}), \text{ and } |min(S_i)_j| = 1\}$.

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

$RMin(\mathcal{Y}) = min(\mathcal{Y})^1 \cup (Min(\mathcal{Y}) - M^*) \cup CM$.
Where: $M^* = \{y \mid y \in min^x, \ \forall min^x \in M\}$;
$M = \{min^x \mid \forall min^x \in Min(\mathcal{Y})\}$;
$min^x = \{S_x \mid S_x = min(S_i), \ \forall min(S_i) \in min(\mathcal{Y}), \text{ such that } x \in min(S_i), \text{ and } x \in Min(\mathcal{Y})\}$.
And $CM = \{x \mid x \in Min(\mathcal{Y}), \text{ such that } min^x \in M\}$.

Given an ordered set X w.r.t. $\leqslant$, and a condition c which involves X in its premisse,

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

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

Figure 2: Summary of Preliminary Definitions for Compromised Revision

- Then we define $CI(\alpha)$ in 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 $\Delta_K \bigcup\{\alpha\} \vdash \perp$. (See Definition 2.10).

- From $CI(\alpha)$, we get a safe-maximal subset, w.r.t. condition c_2, where $c_2 = CI(\alpha) \bigcup \Delta_K \bigcup P_K \nvdash \perp$. The auxiliary sets $Fail(CI(\alpha))_{c_2}$ and $RMin(Fail(CI(\alpha))_{c_2})$ are used to create $Smax(CI(\alpha))_{c_2}$. And we prove that $Smax(CI(\alpha))_{c_2} \bigcup \Delta_K \bigcup P_K \nvdash \perp$. (See Remarks 2.7, and 2.8; and Proposition 2.3).

- Finally, we get a safe-maximal subset of $Smax(CI(\alpha))_{c_2}$, w.r.t. the condition $c_3 = Smax(CI(\alpha))_{c_2} \bigcup \Delta_K \nvdash \alpha$, denoted as $Smax(Smax(CI(\alpha))_{c_2})_{c_3}$. This set is obtained using the auxiliary set $Fail(Smax(CI(\alpha))_{c_2})_{c_3}$ and the set $RMin(Fail(Smax(CI(\alpha))_{c_2})_{c_3})$. $Smax(Smax(CI(\alpha))_{c_2})_{c_3} \bigcup \Delta_K \nvdash \alpha$ is also proved. (See Remarks 2.9, and 2.10, 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 2.4 are avoided.

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

Remark 2.5 ($Fail(\Delta_K)_{c_1}$)

Given a knowledge base $K = \langle \Delta_K, P_K \rangle$ an input α, and the condition $c_1 = \Delta_K \bigcup\{\alpha\} \nvdash \perp$. By Definition 2.8, we have that $Fail(\Delta_K)_{c_1}$ is the following set:

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

Each set 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 [28]

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 $\vdash \neg\alpha$, and the set $Fail(\Delta_K)_{c_1} = \{S_1, S_2, \cdots, S_l\}$, it is sufficient to retract one element from each $S_i \subseteq Fail(\Delta_K)_{c_1}$, such that $(\Delta_K - \mathcal{S}(Fail(\Delta_K)_{c_1})) \bigcup\{\alpha\} \nvdash \perp$, where $\mathcal{S}(Fail(\Delta_K)_{c_1}) = \{x_i \mid \forall S_i \in Fail(\Delta_K)_{c_1}, \exists y \in S_i, \text{such that } x_i = y\}$.

Remark 2.6 ($Smax(\Delta_K)_{c_1}$)

Given a knowledge base $K = \langle \Delta_K, P_K \rangle$ an input α, and the condition $c_1 = \Delta_K \bigcup\{\alpha\} \nvdash \perp$. By Definition 2.9, we have that $Smax(\Delta_K)_{c_1}$ is the following set:

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

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

[28] Proofs of the propositions cited in this section, are found in appendix A of this paper.

•

Proposition 2.2

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

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

Definition 2.10 ($CI(\alpha)$)

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:*

$$\begin{aligned} CI(\alpha) \ = \ & \{y \mid Smax(\Delta_K)_{c_1} \nvdash y \ \text{and} \ Smax(\Delta_K)_{c_1} \cup \{\alpha\} \vdash y; \\ & y \neq \alpha, \ \text{and} \ Smax(\Delta_K)_{c_1} \cup \{y\} \nvdash \alpha\}, \end{aligned}$$

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

□

Example 2.6

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

•

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, (i.e. neither present nor derivable), from the resulting base. However, Definition 2.10, as it stands does not satisfy these conditions. Notice that it is not guaranteed that $CI(\alpha) \bigcup \Delta_K \nvdash \alpha$, nor that $CI(\alpha) \bigcup \Delta_K \bigcup P_K \nvdash \bot$. In Example 2.6, for instance, that last condition is not satisfied. Hence, we still need to specify a subset of $CI(\alpha)$, that satisfies those conditions. Such subset of $CI(\alpha)$ is then obtained as a safe-maximal subset, as stated below.

Remark 2.7 ($Fail(CI(\alpha))_{c_2}$)

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

$$Fail(CI(\alpha))_{c_2} = \begin{cases} \emptyset & \textit{if } c_2 \textit{ is satisfied;} \\ \{S \mid S \subseteq CI(\alpha); \\ \textit{such that } S \cup \Delta_K \vdash \bot, \textit{and} \\ S \textit{ is minimal w.r.t. } \subseteq\} & \textit{otherwise.} \end{cases}$$

Each set S_i *of* $Fail(CI(\alpha))_{c2}$*, for* $i = 1, \cdots, m$*, is ordered by* $\leqslant$ *as a subset of* $CI(\alpha)$*.*

•

Remark 2.8 ($Smax(CI(\alpha))_{c2}$)

Given a knowledge base $K = \langle \Delta_K, P_K \rangle$*, an input* α *such that* $CI(\alpha)$ *is non-empty, and the condition* $c_2 = CI(\alpha) \bigcup \Delta_K \bigcup P_K \nvdash \perp$*, by Definition 2.9, we have that* $Smax(CI(\alpha))_{c2}$ *is the following set:*

$$Smax(CI(\alpha))_{c2} = \begin{cases} CI(\alpha) & \text{if } c_2 \text{ is satisfied;} \\ \\ CI(\alpha) - RMin(Fail(CI(\alpha))_{c2}) & \text{otherwise.} \end{cases}$$

$Smax(CI(\alpha))_{c2}$ *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* $\vdash \neg\alpha$*,* $Smax(CI(\alpha))_{c2} \bigcup \Delta_K \bigcup P_K \nvdash \perp$*.*

Remark 2.9 ($Fail(Smax(CI(\alpha))_{c2})_{c3}$)

Given a knowledge base $K = \langle \Delta_K, P_K \rangle$*, an input sentence* α *such that* $CI(\alpha)$ *is non-empty, and the condition* $c_3 = Smax(CI(\alpha))_{c2} \cup \Delta_K \nvdash \alpha$*, by Definition 2.8, we have that* $Fail(Smax(CI(\alpha))_{c2})_{c3}$ *is the following set:*

$$Fail(Smax(CI(\alpha))_{c2})_{c3} = \begin{cases} \emptyset & \text{if } c_3 \text{ is satisfied;} \\ \\ \{S \mid S \subseteq Smax(CI(\alpha))_{c2}; \\ \text{such that } S \cup \Delta_K \vdash \alpha, \text{and} \\ S \text{ is minimal w.r.t. } \subseteq\} & \text{otherwise.} \end{cases}$$

Each set S_i *of* $Fail(Smax(CI(\alpha))_{c2})_{c3}$*, for* $i = 1, \cdots, n$*, is ordered by* $\leqslant$ *as a subset of* $CI(\alpha)$*.*

•

Remark 2.10 ($Smax(Smax(CI(\alpha))_{c2})_{c3}$)

Given a knowledge base $K = \langle \Delta_K, P_K \rangle$*, an input* α *such that* $CI(\alpha)$ *is non-empty, and the condition* $c_3 = Smax(CI(\alpha))_{c2} \bigcup \Delta_K \nvdash \alpha$*, by Definition 2.9, we have that* $Smax(Smax(CI(\alpha))_{c2})_{c3}$ *is the following set:*

$$Smax(Smax(CI(\alpha))_{c2})_{c3} = \begin{cases} Smax(CI(\alpha))_{c2}, \quad \text{if } c_3 \text{ is satisfied;} \\ \\ Smax(CI(\alpha))_{c2} - RMin(Fail(Smax(CI(\alpha))_{c2})_{c3}), \\ \text{otherwise.} \end{cases}$$

$Smax(Smax(CI(\alpha))_{c2})_{c3}$ *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, and it is not the case that* $\vdash \neg\alpha$*,* $Smax(Smax(CI(\alpha))_{c2})_{c3} \bigcup \Delta_K \nvdash \alpha$*.*

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

Given $K = \langle \Delta_K, P_K \rangle$, and $c_1 = \Delta_K + \alpha \nvdash \bot$,

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

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

$$CI(\alpha) = \{y \mid Smax(\Delta_K)_{c_1} \nvdash y \text{ and } Smax(\Delta_K)_{c_1} \cup \{\alpha\} \vdash y; \\ y \neq \alpha, \text{ and } Smax(\Delta_K)_{c_1} \cup \{y\} \nvdash \alpha\},$$

Given $c_2 = CI(\alpha) \bigcup \Delta_K \nvdash \bot$,

$$Fail(CI(\alpha))_{c_2} = \begin{cases} \emptyset & \text{if } c_2 \text{ is satisfied;} \\ \{S \mid S \subseteq CI(\alpha); \\ \text{such that } S \cup \Delta_K \vdash \bot, \text{and} \\ S \text{ is minimal w.r.t. } \subseteq\} & \text{otherwise.} \end{cases}$$

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

Given $c_3 = Smax(CI(\alpha))_{c_2} \cup \Delta_K \nvdash \alpha$,

$$Fail(Smax(CI(\alpha))_{c_2})_{c_3} = \begin{cases} \emptyset, \\ \quad \text{if } c_3 \text{ is satisfied;} \\ \{S \mid S \subseteq Smax(CI(\alpha))_{c_2}; \\ \text{such that } S \cup \Delta_K \vdash \alpha, \text{and} \\ S \text{ is minimal w.r.t. } \subseteq\}, \\ \quad \text{otherwise.} \end{cases}$$

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

Figure 3: Summary of Definitions for the case that $P_K + \alpha \vdash \bot$.

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

In 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 2.8, we obtain such a set as shown below.

Remark 2.11 ($\perp_\alpha$)

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

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

Where each set S_i in $\perp_\alpha$, 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 2.7

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
$min(\perp_a) = \{\{b, c\}, \{q\}\}$.

•

The proposition below guarantees 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. It is a specific case of Proposition 2.1.

Proposition 2.5

Given a base $K = \langle \Delta_K, P_K \rangle$, and an input sentence α, such that it is not the case that $\vdash \neg\alpha$ nor that $\{\alpha\} \bigcup P_K \vdash \perp$. 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 of each set $S_i \in \perp_\alpha$, for $i = 1, \cdots, n$, from $\Delta_K \cup \{\alpha\}$ such that $((\Delta_K \cup \{\alpha\}) - S(\perp_\alpha)) \cup P_K \nvdash \perp$, where $S(\perp_\alpha) = \{x_i \mid \forall S_i \in \perp_\alpha, \exists y \in S_i, \text{ such that } x_i = y\}$.

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 ® α, 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 at least one minimal element 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 not yet 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 at least one minimal element of each set, by adopting an algorithm which identifies common elements among the sets, for regaining consistency. We call this option *"the impartial-choice for safe-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 2.5 to be a sufficient condition for avoiding inconsistency in the revised base. Consequently, within this option, the conservativity principle 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 safe-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 ® α, 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 $min(\perp_\alpha)$, $min(\perp_\alpha)^+$, $min(\perp_\alpha)^1$, $Min(\perp_\alpha)$, and $RMin(\perp_\alpha)$, according to the general definitions 2.3, 2.4 2.5, 2.6 and 2.7, respectively.

Then, we need to define a function which caters for the choice of retraction, combining *"the impartial-choice for safe-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 safe-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 mechanism 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 2.11 (Req)

Assume that L is a propositional language closed under the logical operators. Given a set $min(\bot_\alpha)^+$, 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:

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

□

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

Definition 2.12 ($Choice$)

Given a language L, and a set $\bot_\alpha$, 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 2.11, and Ω is the set of all sets of sentences of L. We write $Choice(option, \bot_\alpha) = chosen\text{-}set$, where $option$ is either $\{d\}$ or $\{u\}$, and the resuting $chosen\text{-}set$ of minimals is obtained in the following way:

$$chosen\text{-}set = \begin{cases} RMin(\bot_\alpha) & \text{if } option = \{d\}; \\ \{x_j \mid \forall min(S_i)_j \in min(\bot_\alpha)^+, \\ \exists y \in min(S_i)_j, \text{such that } x_j = y; \\ \text{and } x_j \text{ is chosen by the user}\} & \text{if } option = \{u\}. \end{cases}$$

Where $1 \leq i \leq n$, and $j = 1, \cdots, k$, for $k \leq n$.[29]

□

Remark 2.12

In Definition 2.12, we count again on a user-interface procedure which, given the set $min(\bot_\alpha)^+$ 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 $min(\bot_\alpha)^+$ family.

•

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

Definition 2.13 (R_α)

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

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

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

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

□

Remark 2.13

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

•

Example 2.8 (R_α)

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

$\perp_a = \{\{b, f, d\}, \{e, c\}, \{q\}\}$,

where:

$min(\perp_a) = \{\{b, f\}, \{e\}, \{q\}\}$;

$min(\perp_a)^+ = \{\{b, f\}\}$; *and*

$min(\perp_a)^1 = \{e, q\}$.

$RMin(\perp_a) = \{b, e, f, q\}$.

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 2.8, for instance, we would have that $(\Delta_K - R_\alpha) \to e$ 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 $X \subseteq \Delta_K$.

First, we obtain the set $Fail(\Delta_K - X)_{c_5}$, where c_5 is the condition $(\Delta_K - X) \nvdash x, \forall x \in X$. Then, we get the safe-maximal subset of $(\Delta_K - X)$, $Smax(\Delta_K - X)_{c_5}$, relative to condition c_5.

Remark 2.14 ($Fail(\Delta_K - X)_{c_5}$)

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* $Fail(\Delta_K - X)_{c_5}$ *is the following set:*

$$Fail(\Delta_K - X)_{c_5} = \begin{cases} \emptyset & \text{if } c_5 \text{ is satisfied ;} \\ \{S \mid S \subseteq (\Delta_K - X); \\ \text{such that } \exists x \in X, \\ (\Delta_K - X) \vdash x, \text{and} \\ S \text{ is minimal w.r.t. } \subseteq\} & \text{otherwise.} \end{cases}$$

Each set S_i of $Fail(\Delta_K - X)_{c_5}$, for $i = 1, \cdots, k$, is ordered by $\leqslant$ as a subset of $CR(X')$.

•

Remark 2.15 ($Smax(\Delta_K - X)_{c_5}$)

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 2.9, we have that $Smax(\Delta_K - X)_{c_5}$ is the following set:

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

$Smax(\Delta_K - X)_{c_5}$ 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 have 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 2.6

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 $\langle (Smax(\Delta_K - R_\alpha)_{c_5} \cup \{\alpha\}),\ P_K \rangle$.

We use the notion of propagation of the ordering $\leqslant$, as in Definition 2.2, for defining the ordering of 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 $RMin(Fail(\Delta_K - X)_{c_5})$, 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 2.14 ($CR(X')$)

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 $RMin(Fail(\Delta_K - X)_{c_5})$, so that it is guaranteed that $Smax(\Delta_K - X)_{c_5} \nvdash x,\ \forall x \in X$, assume that $X' = X \bigcup RMin(Fail(\Delta_K - X)_{c_5})$. Let $CR(X')$ be the set of consequences of the elements in X' w.r.t. Δ_K, considering $\vdash$, such that:

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

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

□

Example 2.9 ($CR(X')$)

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 2.14, ??, and 2.15,* $Smax(\Delta_K - X)_{c_5} = (\Delta_K - X) = \{d, q \rightarrow e, c \rightarrow f, d \wedge e \rightarrow h, h \rightarrow q\}$. *By Definition 2.14, 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 2.14, this is not yet possible. Below, we state some points that need to be observed.

Remark 2.16

Notice that it is not guaranteed that $CR(X') \bigcup Smax(\Delta_K - X)_{c_5} \nvdash x, \forall x \in X'$. *In Example 2.9, 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* $\langle Smax(\Delta_K - X)_{c_5} \cup CR(X'), P_K \rangle$, *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 2.16, 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 2.17 ($Fail(CR(X'))_{c_6}$)

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* $c_6 = CR(X') \bigcup Smax(\Delta_K - X)_{c_5} \nvdash x, \forall x \in X'$. *By Definition 2.8, we have that* $Fail(CR(X'))_{c_6}$ *is the following set:*

$$Fail(CR(X'))_{c_6} = \begin{cases} \emptyset & \text{if } c_6 \text{ is satisfied ;} \\ \{S \mid S \subseteq CR(X'); \\ \text{such that } \exists x \in X', \\ S \cup Smax(\Delta_K - X)_{c_5} \vdash x, \\ \text{and } S \text{ is minimal w.r.t. } \subseteq\} & \text{otherwise.} \end{cases}$$

Each set S_i *of* $Fail(CR(X'))_{c_6}$, *for* $i = 1, \cdots, k$, *is ordered by* $\leqslant$ *as a subset of* $CR(X')$.

•

Remark 2.18 ($Smax(CR(X'))_{c_6}$)

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 2.9, we have that* $Smax(CR(X'))_{c_6}$ *is the following set:*

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

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

•

Proposition 2.7

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 2.10 ($Smax(CR(X'))_{c_6}$)

Considering Example 2.9, in which
$\Delta_K = \{b, c, d, q, q \to e, c \to f, d \wedge e \to h, h \to q\}$,
such that $b \leqslant d$, $c \leqslant d$, $d \leqslant q$, $q \leqslant q \to e$, $c \leqslant c \to f$, $d \leqslant d \wedge e \to h$, and $h \leqslant h \to q$; $X = \{b, c, q\}$; $X' = X$ 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, it 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 $R'_\alpha = R_\alpha \bigcup RMin(Fail(\Delta_K - X)_{c_5})$, 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 $Smax(CR(R'_\alpha))_{c_6} \bigcup (Smax((\Delta_K - R_\alpha)_{c_5} \cup \{\alpha\}) \bigcup P_K \nvdash \perp$.

Below we get $Fail(Smax(CR(R'_\alpha))_{c_6})_{c_7}$, which is the family of subsets of $Smax(CR(R'_\alpha))_{c_6}$ that contribute to the failure of the condition c_7.

Remark 2.19 ($Fail(Smax(CR(R'_\alpha))_{c_6})_{c_7}$)

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 2.13, such that $Smax(CR(R'_\alpha))_{c_6}$ is non-empty, and the condition $c_7 = Smax(CR(R'_\alpha))_{c_6} \bigcup (Smax((\Delta_K - R_\alpha)_{c_5} \cup \{\alpha\}) \bigcup P_K \nvdash \perp$. By Definition 2.8, we have that $Fail(Smax(CR(R'_\alpha))_{c_6})_{c_7}$ is the following set:

$$Fail(Smax(CR(R'_\alpha))_{c_6})_{c_7} = \begin{cases} \emptyset, & \textit{if } c_7 \textit{ is satisfied;} \\ \{S \mid S \subseteq Smax(CR(R'_\alpha))_{C_6}; \textit{ such that} \\ S \cup (Smax(\Delta_K - R_\alpha)_{c_5} \cup \{\alpha\}) \cup P_K \vdash \perp, \\ \textit{and } S \textit{ is minimal w.r.t. } \subseteq\}, & \textit{otherwise.} \end{cases}$$

Each set S_i *of* $Fail(Smax(CR(R'_\alpha))_{c6})_{c7}$, *for* $i = 1, \cdots, o$, *is ordered by* $\leqslant$ *as a subset of* $CR(R'_\alpha)$.

•

Remark 2.20 ($Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$)

Given a knowledge base $K = \langle \Delta_K, P_K \rangle$, *an input sentence* α, *such that it is not the case that* $\vdash \neg\alpha$, $P_K \bigcup \{\alpha\} \nvdash \bot$ *and* $K \bigcup \{\alpha\} \vdash \bot$, *a set* R_α, *as in Definition 2.13, such that* $Smax(CR(R'_\alpha))_{c6}$ *is non-empty, and the condition* c_7 *is* $Smax(CR(R'_\alpha))_{c6} \bigcup (Smax(\Delta_K - R_\alpha)_{c5} \cup \{\alpha\}) \bigcup P_K \nvdash \bot$. *By Definition 2.9, we have that* $Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$ *is the following set:*

$$Smax(Smax(CR(R'_\alpha))_{c6})_{c7} = \begin{cases} Smax(CR(R'_\alpha))_{c6}, & \textit{if } c_7 \textit{ is satisfied;} \\ Smax(CR(R'_\alpha))_{c6} - RMin(Fail(Smax(CR(R'_\alpha))_{c6})_{c7}), & \\ \textit{otherwise.} & \end{cases}$$

The set $Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$ *is ordered by* $\leqslant$, *as a subset of* $CR(R'_\alpha)$.

•

Proposition 2.8

Given a base $K = \langle \Delta_K, P_K \rangle$, *and an input sentence* α, *such that it is not the case that* $\vdash \neg\alpha$, *and a set* R_α, *as in Definition 2.13, such that* $Smax(CR(R'_\alpha)_{c6}$ *is non-empty. If* $P_K \bigcup \{\alpha\} \nvdash \bot$ *and* $K \bigcup \{\alpha\} \vdash \bot$, $Smax(Smax(CR(R'_\alpha))_{c6})_{c7} \bigcup (Smax(\Delta_K - R_\alpha)_{c5} \cup \{\alpha\}) \bigcup P_K \nvdash \bot$.

Definitions for Compromised Contraction & Revision

Below we define a general compromised contraction for our method. Notice that Definition 2.15 does not cater for the accomplishment of a compromised revision. It only specifies a compromised contraction, independent from any other operation.

Definition 2.15 (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 2.14, let* $\ominus$ *denote the compromised contraction operation, which retracts* X' *from* K, *such that* $K \ominus X'$, *is the following:*

$$K \ominus X' = \langle (Smax(\Delta_K - X)_{c5} \cup Smax(CR(X'))_{c6}), P_K \rangle.$$

□

From the definition above, we can construct the compromised contraction which is 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 2.16 (Compromised Contraction for Revision)

Definitions for the case that $P_K + \alpha \nvdash \bot$ and $K + \alpha \vdash \bot$: (Part I)

Given $K = \langle \Delta_K, P_K \rangle$, an input α, and $c_4 = \Delta_K \bigcup P_K \bigcup \{\alpha\} \nvdash \bot$.

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

$Req(\{d, u\}) = option$, where:

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

$Choice(option, \bot_\alpha) =$ chosen-set, where:

$$\text{chosen-set} = \begin{cases} RMin(\bot_\alpha) & \text{if } option = \{d\}; \\ \{x_j \mid \forall min(S_i)_j \in min(\bot_\alpha)^+, \\ \quad \exists y \in min(S_i)_j, \text{such that } x_j = y; \\ \quad \text{and } x_j \text{ is chosen by the user}\} & \text{if } option = \{u\}. \end{cases}$$

R_α = chosen-set $\cup \; min(\bot_\alpha)^1$.

Given $c_5 = (\Delta_K - X) \nvdash x, \forall x \in X$,

$$Fail(\Delta_K - X)_{c_5} = \begin{cases} \emptyset & \text{if } c_5 \text{ is satisfied ;} \\ \{S \mid S \subseteq (\Delta_K - X); \\ \quad \text{such that } \exists x \in X, \\ \quad (\Delta_K - X) \vdash x, \text{and} \\ \quad S \text{ is minimal w.r.t. } \subseteq\} & \text{otherwise.} \end{cases}$$

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

Figure 4: Summary for the case that $P_K + \alpha \nvdash \bot$ and $K + \alpha \vdash \bot$ - Part I.

Definitions for the case that $P_K + \alpha \nvdash \perp$ and $K + \alpha \vdash \perp$: (Part II)

Given $K = \langle \Delta_K, P_K \rangle$, an input α, and the set $X' = X \bigcup Min(Fail(\Delta_K - X)_{c5})$, where $X \subseteq \Delta_K$,

$$CR(X') = \{y \mid \Delta_K \vdash y \text{ and } Smax(\Delta_K - X)_{c5} \nvdash y; \text{ and } \forall x \in X', y \neq x, \text{ and } Smax(\Delta_K - X)_{c5} \cup \{y\} \nvdash x\},$$

Given $c_6 = CR(X') \bigcup Smax(\Delta_K - X)_{c5} \nvdash x, \forall x \in X$,

$$Fail(CR(X'))_{c6} = \begin{cases} \emptyset & \text{if } c_6 \text{ is satisfied ;} \\ \{S \mid S \subseteq CR(X'); \text{ such that } \exists x \in X', S \cup Smax(\Delta_K - X)_{c5} \vdash x, \text{and } S \text{ is minimal w.r.t. } \subseteq\} & \text{otherwise.} \end{cases}$$

$$Smax(CR(X'))_{c6} = \begin{cases} CR(X'), \text{ if } c_6 \text{ is satisfied;} \\ CR(X') - RMin(Fail(CR(X'))_{c6}), \text{ otherwise.} \end{cases}$$

Given $c_7 = Smax(CR(R'_\alpha))_{c6} \bigcup (Smax(\Delta_K - R_\alpha)_{c5} \cup \{\alpha\}) \bigcup P_K \nvdash \perp$,

$$Fail(Smax(CR(R'_\alpha))_{c6})_{c7} = \begin{cases} \emptyset, \text{ if } c_7 \text{ is satisfied;} \\ \{S \mid S \subseteq Smax(CR(R'_\alpha))_{C6} \text{ ; such that } S \cup (Smax(\Delta_K - R_\alpha)_{c5} \cup \{\alpha\}) \cup P_K \vdash \perp, \text{ and } S \text{ is minimal w.r.t. } \subseteq\}, \text{ otherwise.} \end{cases}$$

$$Smax(Smax(CR(R'_\alpha))_{c6})_{c7} = \begin{cases} Smax(CR(R'_\alpha))_{c6}, & \text{if } c_7 \text{ is satisfied;} \\ Smax(CR(R'_\alpha)_{c6} - RMin(Fail(Smax(CR(R'_\alpha))_{c6})_{c7}), & \text{otherwise.} \end{cases}$$

Figure 5: Summary for the case that $P_K + \alpha \nvdash \perp$ and $K + \alpha \vdash \perp$ - Part II.

Given a knowledge base $K = \langle \Delta_K, P_K \rangle$, *an input sentence* α, *and a set* R_α, *as in Definition 2.13, 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* $R'_\alpha = R_\alpha \bigcup RMin(Fail(\Delta_K - R_\alpha)_{c5})$. *So,* $[K \ominus R'_\alpha]_\alpha = \langle \Delta_{[K-R'_\alpha]_\alpha}, P_K \rangle$, *and* $\Delta_{[K-R'_\alpha]_\alpha}$ *is obtained in the following way:*

$$\Delta_{[K\ominus R'_\alpha]_\alpha} = Smax(\Delta_K - R_\alpha)_{c5} \cup Smax(Smax(CR(R'_\alpha))_{c6})_{c7}.$$

□

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 $Smax(Smax(CI(\alpha))_{c2})_{c3}$, and our compromised result is given by the knowledge base
$K \circledR \alpha = \langle (\Delta_K \bigcup Smax(Smax(CI(\alpha))_{c2})_{c3}), P_K \rangle$.
- 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 obtain the sets $\bot_\alpha$; R_α; $Smax(\Delta - R_\alpha)_{c5}$; $CR(R'_\alpha)$ and $Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$.

  Then, we make the compromised contraction $[K \ominus R'_\alpha]_\alpha = \langle \Delta_{[K-R'_\alpha]_\alpha}, P_K \rangle$, where $\Delta_{[K-R'_\alpha]_\alpha} = Smax(\Delta - R_\alpha)_{c5} \cup Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$.

  Finally, we perform the compromised revision of the base K by α, such that: $K \circledR \alpha = \langle (\Delta_{[K-R'_\alpha]_\alpha} \cup \{\alpha\}), P_K \rangle$.

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

Definition 2.17 (Compromised Revision Function)

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

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

(Case 2) *If* $K + \alpha \nvdash \bot$ [30], *then* $\Delta_{K®\alpha} = \Delta_K \bigcup \{\alpha\}$.

(Case 3) *If* $P_K + \alpha \vdash \bot$, *then* $\Delta_{K®\alpha} = \Delta_K \bigcup Smax(Smax(CI(\alpha))_{c2})_{c3}$.

(Case 4) *If* $K + \alpha \vdash \bot$ *and* $P_K + \alpha \nvdash \bot$, *then*
$\Delta_{K®\alpha} = Smax(\Delta_K - R_\alpha)_{c5} \cup Smax(Smax(CR(R'_\alpha))_{c6})_{c7} \bigcup \{\alpha\}$.

□

2.4.1 Some Properties of Compromised Revision

In this section, we present some properties of the compromised revision function, via proposition and theorem statements[31].

The proposition below states that all the consistent consequences w.r.t. K of an input sentence α, which violates some integrity contraints of a base K, and which were not orginally in K, are available in $K ® \alpha$.

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* $\vdash \neg\alpha$, *and* $CI(\alpha)$ *is non-empty, if* $P_K + \alpha \vdash \bot$, *then* $\forall x \in Smax(Smax(CI(\alpha))_{c2})_{c3}$, $K \nvdash x$ *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 $K ® \alpha$, are derivable from $K ® \alpha$.

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* $K + \alpha \vdash \bot$ *and* $P_K + \alpha \nvdash \bot$, *then* $\forall x \in Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$, $K \vdash x$ *and also* $K ® \alpha \vdash x$.

The two propositions to follow, state important conditions concerning the elements of the base $K ® \alpha$.

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* $K + \alpha \vdash \bot$ *and* $P_K + \alpha \nvdash \bot$, *then* $\forall x \in R'_\alpha$, $K ® \alpha \nvdash x$.

Proposition 2.12

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* $\beta \in K®\alpha$, *then either* $\beta \in \Delta_K$ *and* $\beta \notin R'_\alpha$; *or* $\beta \notin \Delta_K$ *and either* $\beta \in Smax(Smax(CI(\alpha))_{c2})_{c3}$, *or* $\beta = \alpha$, *or* $\beta \in Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$.

Below, we state the consistency theorem of the compromised revision $K ® \alpha$.

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

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

Definitions for Compromised Contraction and Revision:

- Compromised Contraction:

Given $K = \langle \Delta_K, P_K \rangle$, X', and $CR(X')$, such that
$X' = X \bigcup RMin(Fail(\Delta_K - X)_{c5})$, where $X \in \Delta_K$,

$$K \ominus X' = \langle (Smax(\Delta_K - X)_{c5} \cup Smax(CR(X'))_{c6}), P_K \rangle.$$

- Compromised Contraction for Revision:

Given $K = \langle \Delta_K, P_K \rangle$, α, and R'_α, where
$R'_\alpha = R_\alpha \bigcup RMin(Fail(\Delta_K - R_\alpha)_{c5})$,

$$[K \ominus R'_\alpha]_\alpha = \langle \Delta_{[K-R'_\alpha]_\alpha}, P_K \rangle,$$

$$\Delta_{[K-R'_\alpha]_\alpha} = Smax(\Delta_K - R_\alpha)_{c5} \cup Smax(Smax(CR(R'_\alpha))_{c6})_{c7}.$$

- Compromised Revision Function:

Given $K = \langle \Delta_K, P_K \rangle$, and an input sentence α,

$$K \circledR \alpha = \langle \Delta_{K \circledR \alpha}, P_K \rangle,$$

where $\Delta_{K \circledR \alpha}$ is such that one of the following conditions holds:

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

(Case 2) If $K + \alpha \nvdash \bot$, then $\Delta_{K \circledR \alpha} = \Delta_K \cup \{\alpha\}$.

(Case 3) If $P_K + \alpha \vdash \bot$, then $\Delta_{K \circledR \alpha} = \Delta_K \cup Smax(Smax(CI(\alpha))_{c2})_{c3}$.

(Case 4) If $K + \alpha \vdash \bot$ and $P_K + \alpha \nvdash \bot$, then $\Delta_{K \circledR \alpha} = Smax(\Delta_K - R_\alpha)_{c5} \cup Smax(Smax(CR(R'_\alpha))_{c6})_{c7} \cup \{\alpha\}$.

Figure 6: Summary of Definitions for Compromised Contraction and Revision.

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

Proof:

By Definition 2.17, 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, considering that $P_K + \alpha \vdash \bot$, we have that $\Delta_{K \circledR \alpha} = \Delta_K \bigcup Smax(Smax(CI(\alpha))_{c2})_{c3}$. By Proposition 2.3, we have proved that $Smax(CI(\alpha))_{c2} \bigcup \Delta_K \bigcup P_K \nvdash \bot$. Since by Remark .2.10, $Smax(Smax(CI(\alpha))_{c2})_{c3}$ is a subset of $Smax(CI(\alpha))_{c2}$, and we also have that $Smax(Smax(CI(\alpha))_{c2})_{c3} \bigcup \Delta_K \bigcup P_K \nvdash \bot$. 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 $\Delta_{K \circledR \alpha} = Smax(\Delta_K - R_\alpha)_{c5} \cup Smax(Smax(CR(R'_\alpha))_{c6})_{c7} \bigcup \{\alpha\}$. By Proposition 2.8, we have proved that $Smax(Smax(CR(R'_\alpha))_{c6})_{c7} \bigcup Smax(\Delta_K - R_\alpha)_{c5} \bigcup \{\alpha\} \bigcup P_K \nvdash \bot$. Hence, it is already guaranteed for case 4 that $K \circledR \alpha \nvdash \bot$.

In conclusion, we have that for each of the four cases of Definition 2.17, we prove that $K \circledR \alpha \nvdash \bot$.

■

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.

As we would already expect, our system does not satisfy the original notion of persistence. In our compromised approach to revision, it is not enough to say that we should be able to derive from the revised base, all the consequences of the original base which do not contradict the revised sentence α. Since we have extended the consistency notions of our system with the presence of integrity constraints in the base, we should also consider them into account at this point. We need to adjust the notion of persistence, to cater for our specific requirements of compromised revision.

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 2.17, we have that in case 1, $K \circledR \alpha = K$. And, by Definition 2.15, $(K \circledR \alpha) \ominus \{\alpha\} = K$. So, in this case it is vacuously guaranteed that $\forall x$ such that $K \vdash x$, $(K \circledR \alpha) \ominus \{\alpha\} \vdash x$. In case 2, no sentence is retracted from the base K, to accomplish the revised base $K \circledR \alpha$, since $\Delta_{K \circledR \alpha} = \Delta_K \bigcup \{\alpha\}$. And we have that by Definition 2.15, $(K \circledR \alpha) \ominus \{\alpha\} = \langle \Delta^*, P_K \rangle$, where $\Delta^* = \Delta_K \bigcup Smax(CR(\{\alpha\}'))_{c_6}$ and no element is retracted from the original base K. So, for case 2 also it is trivially guaranteed that $\forall x$ such that $K \vdash x$, $(K \circledR \alpha) \ominus \{\alpha\} \vdash x$, provided that $x \neq \alpha$. In case 3, $(K \circledR \alpha) = \langle (\Delta_K \cup Smax(Smax(CI(\alpha)_{c_2})_{c_3}), P_K \rangle$, where no element is retracted from Δ_K, and $\alpha \notin K \circledR \alpha$. So, $(K \circledR \alpha) \ominus \{\alpha\} = K \circledR \alpha$, and we guarantee that $\forall x$ such that $K \vdash x$, $(K \circledR \alpha) \ominus \{\alpha\} \vdash x$. So, for the cases 1, 2 and 3 of the compromised revision function definition, we have that compromised persistence is satisfied.[32] From case 4 of Definition 2.17 and from Definition 2.15, $(K \circledR \alpha) \ominus \{\alpha\} = \langle (Smax(\Delta_{K \circledR \alpha} - \alpha)_{c_5} \cup Smax(CR(\{\alpha\}'))_{c_6}), P_K \rangle$, where $\Delta_{K \circledR \alpha} = Smax(\Delta_K - R_\alpha)_{c_5} \cup Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7} \bigcup \{\alpha\}$. By definition, the consequences of K which are not anymore available from $(K \circledR \alpha)$ are the ones which allowed the elements of the set R_α to be derivable, that is the elements of the set R'_α where $R'_\alpha \subseteq \Delta_K$. As proved in Proposition 2.11 $\forall x \in R'_\alpha$, $K \circledR \alpha \nvdash x$. So, we have that $\forall x$ such that $(K \circledR \alpha) \ominus \{\alpha\} \vdash x$, $x \notin R'_\alpha$ since by retracting $\{\alpha\}$ from $K \circledR \alpha$ the elements of the set $Smax(CR(\{\alpha\}'))_{c_6}$ do not derive any element of R'_α. To show that $\forall x$ such that $(K \circledR \alpha) \ominus \{\alpha\} \vdash x$, $x \neq \alpha$, we have that by Definition 2.15, it is guaranteed that if α was derivable from $K \circledR \alpha$, 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.

■

Tenacity

As stated in [Han-91], the tenacity property characterizes their maxichoice internal partial meet revision. This property can be presented in two forms, namely *tenacity* and *extreme tenacity*.[33] Below, we state both poperties' presentations, considering a revision $*$.

Tenacity	$\forall x \in K$ either $x \in K * \alpha$ or $\neg x \in Cn(K * \alpha)$.
Extreme Tenacity	$\forall x \in Cn(K)$ either $x \in Cn(K * \alpha)$ or $\neg x \in Cn(K * \alpha)$.

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

[33] In the AGM model, tenacity and extreme tenacity are logically equivalent.

In compromised revision, the correspondent tenacity property: $\forall x \in K$ either $x \in K \circledR \alpha$ or $\neg x \in Cn(K \circledR \alpha)$, does not hold.

By Definition 2.17, we have that in the cases 1, 2 and 3, no sentence is retracted from the base K. So, $\forall x \in K$, $x \in K \circledR \alpha$. In case 4 of Definition 2.17, we can have that $x \in K$ and $x \notin K \circledR \alpha$, in so far $x \in R'_\alpha$. This can be shown by Proposition 2.12. Iy Proposition 2.12, we have proved that if $x \in K \circledR \alpha$, then either $x \in \Delta_K$ and $x \notin R'_\alpha$, or $x \not{i}n \Delta_K$ and $x \in Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$. In our case here, we know that $x \in \Delta_K$, hence $x \notin Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$. Then, if we assume that $x \notin K \circledR \alpha$, we can show that $x \in R'_\alpha$. And we have also proved in Proposition 2.11 that $\forall x \in R'_\alpha$, $K \circledR \alpha \nvdash x$. Hence, it is already guaranteed that if $x \in K$ and $x \in R'_\alpha$, then $x \notin K \circledR \alpha$ and $x \notin Cn(K \circledR \alpha)$. By definition, $x \in R'_\alpha$ means that one of the three cases below holds:
Case (a): $x \in R_\alpha$, and the chosen-set has only one minimal element from each set $min(S_i)$ of $min(\perp_\alpha)^+$, such that $\{x, \alpha\} \bigcup P_K \vdash \perp$.
Case (b): $x \in R_\alpha$, and the chosen-set was obtained under the default option, such that $x \in X$, for $X \subseteq R_\alpha$, and X is minimal w.r.t. $\subseteq$, such that $X \bigcup \{\alpha\} \bigcup P_K \vdash \perp$.
Case (c): $x \in RMin(Fail(\Delta_K - R_\alpha)_{c5})$, and $x \in Y$, such that $Y \vdash y$ and $y \in R_\alpha$. Hence, $Y \bigcup \{\alpha\} \bigcup P_K \vdash \perp$, where Y is minimal w.r.t. $\subseteq$.
Since we defined that the elements of P_K are sentences of the type $\bigwedge_{i=1}^{i=n} \omega_i \rightarrow \perp$, where ω_i is a proposition or its negation, we have that :
In case (a) above, when $\{x, \alpha\} \bigcup P_K \vdash \perp$, we have that $x \wedge \alpha \rightarrow \perp$ is an element in P_K. And $x \wedge \alpha \rightarrow \perp$ implies that $\alpha \rightarrow \neg x$. Hence, in this case $\neg x$ is a consequence of $K \circledR \alpha$, since when $x \in R'_\alpha$ we have that $\alpha \in K \circledR \alpha$.
In case (b), when $X \bigcup \{\alpha\} \bigcup P_\Delta \vdash \perp$, for all $x_j \in X$, for $j = 1, \cdots, m$, we have that $x_1 \wedge \cdots \wedge x_m \wedge \alpha \rightarrow \perp$ is an element in P_K. And $x_1 \wedge \cdots \wedge x_m \wedge \alpha \rightarrow \perp$ implies that $\alpha \rightarrow \neg x_1 \cdots \neg x_m$. Analogously, in case (c), when $Y \bigcup \{\alpha\} \bigcup P_K \vdash \perp$, for all $y_j \in Y$, for $j = 1, \cdots, n$, we have that $y_1 \wedge \cdots \wedge y_n \wedge \alpha \rightarrow \perp$ is an element in P_K. And $y_1 \wedge \cdots \wedge y_n \wedge \alpha \rightarrow \perp$ implies that $\alpha \rightarrow \neg y_1 \cdots \neg y_n$. For cases (b) and (c), considering $x \in X$ and $x \in Y$ respectively, we can only say that $\{\alpha\} \bigcup \Gamma \vdash \neg x$, where in (b) $\Gamma = X - \{x\}$ and in (c) $\Gamma = Y - \{x\}$. For both cases $\alpha \in K \circledR \alpha$, but since $\Gamma \subseteq R'_\alpha$ we know that $R'_\alpha \not\subseteq K \circledR \alpha$. We also have proved that $\forall x \in R'_\alpha$, $K \circledR \alpha \nvdash x$. Hence, it is already guaranteed that if $x \in R'_\alpha$, $x \notin K \circledR \alpha$ and $x \notin Cn(K \circledR \alpha)$. So, we cannot have that $\Gamma \subseteq K \circledR \alpha$, nor that $\Gamma \subseteq Cn(K \circledR \alpha)$. Hence, for cases (b) and (c) we cannot show that $\neg x$ is a consequence of $K \circledR \alpha$.

In conclusion, for our compromised revision apporach, the tenacity property cannot be satisfied for two basic reasons. First, our base K is not complete w.r.t. the language L, so that we cannot assure that either x or $\neg x \in Cn(K)$, which would allow us to conclude that if $x \notin Cn(K \circledR \alpha)$ then $\neg x \in Cn(K \circledR \alpha)$. Second, because of our notion of generating inconsistency via the violation of integrity constraints, as well as via classical contradiction, when a sentence x is retracted from the base K in order to accomplish the revision $K \circledR \alpha$, we know that $x \in R'_\alpha$ and we can prove that $x \notin Cn(K \circledR \alpha)$, as shown above, but we cannot guarantee that $\neg x \in Cn(K \circledR \alpha)$.

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 2.17. Hence, we have that the correspondence between compromised revision and contraction is given as follows.

Definition 2.18 (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:

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

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 $B = Smax(Smax(CI(\alpha))_{c2})_{c3}$.*

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

□

Notes:

It is trivial to show that the substitutions of the given sets A and B in $([K \ominus A]_\alpha) \bigcup B$, give the respective results for $K\circledR\alpha$, as stated in Definition 2.17.

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 proper substitutions for case 4, we have that $K\circledR\alpha = [K \ominus R'_\alpha]_\alpha \bigcup \{\alpha\}$.

Remark 2.21

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 Postulates and Formalization Notation:

From Postulates	From Formalization	Simplified Form
$CI(\alpha)$	$Smax(CI(\alpha))_{c2}$	$CI'(\alpha)$
$MaxCI(\alpha)$	$Smax(Smax(CI(\alpha))_{c2})_{c3}$	$SmaxCI'(\alpha)$
$CR(R_\alpha)$	$Smax(CR(R'_\alpha))_{c6}$	$CR'(R'_\alpha)$
$MaxCR(R_\alpha)$	$Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$	$SmaxCR'(R'_\alpha)$

Figure 7: Notation of Postulates x Formalization.

2.4.2 Correspondence between the Postulates and the Revision Function

In this section we show the correspondence between the compromised revision function formalization and the proposed postulates for compromised revision, via some correspondence theorems.

However, before we start with the theorem statements, some considerations concerning the differences in notation and also in concepts, between the formalization and the proposed postulates, need to be pointed out.

In the postulates (®1) to (®9), we have assumed that $CI(\alpha)$ is the largest set of consequences of α w.r.t. Δ_K, such that $\alpha \notin CI(\alpha)$; $CI(\alpha) \bigcup \Delta_K \nvdash \bot$ and $CI(\alpha) \bigcup \Delta_K \nvdash \alpha$. This set corresponds to $Smax(CI(\alpha))_{c_2}$ in our formalization. For simplicity of notation, we will refer to it as $CI'(\alpha)$, in the theorems to follow.

In the postulates, $MaxCI(\alpha)$ is a maximal subset of $CI(\alpha)$ w.r.t. K, such that $MaxCI(\alpha) \bigcup K \nvdash \bot$. In the formalization, we have adopted the safe-maximal subset notion, instead of a maximal one. So, $MaxCI(\alpha)$ is then restricted to $Smax(Smax(CI(\alpha))_{c_2})_{c_3}$. For simplicity of notation, we will refer to it as $SmaxCI'(\alpha)$.

In the postulates, we have assumed that $CR(R_\alpha)$ is the largest set of consequences of all the sentences in R_α, w.r.t. Δ_K, such that $R_\alpha \not\subseteq CR(R_\alpha)$; $\forall y \in CR(R_\alpha)$, $\Delta_K \vdash y$ and $(\Delta_K - R_\alpha) \nvdash y$; and $\forall \beta \in R_\alpha$, $CR(R_\alpha) \bigcup (\Delta_K - R_\alpha) \nvdash \beta$. In our formalization, the same notion is represented by $Smax(CR(R'_\alpha))_{c_6}$. For simplicity of notation, we will refer to it as $CR'(R'_\alpha)$.

In the postulates, $MaxCR(R_\alpha)$ is a maximal subset of $CR(R_\alpha)$, such that $MaxCR(R_\alpha) \bigcup (\Delta_K - R_\alpha) \bigcup P_K \bigcup \{\alpha\} \nvdash \bot$. The safe-maximal subset correspondent to this one in our formalization is $Smax(Smax(CR(R'_\alpha))_{c_6})_{c_7}$. For simplicity of notation, we will refer to it as $SmaxCR'(R'_\alpha)$.

Also in the postulates, D_{CR} and D_{CI} were assumed to denote the differences $D_{CR} = CR(R_\alpha) - MaxCR(R_\alpha)$ and $D_{CI} = CI(\alpha) - MaxCI(\alpha)$. For our formalization, we assume that they correspond to: $D'_{CR} = CR'(R'_\alpha) - SmaxCR'(R'_\alpha)$ and $D'_{CI} = CI'(\alpha) - SmaxCI'(\alpha)$, respectively.

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 (®9). Therefore, the theorems below will only refer to the postulates (®1) to (®8), to make the correspondence with the revision function.

Concerning the set of postulates for compromised revision, we have that postulates (®1), (®6), (®7) and (®8) have to be satisfied by all the four cases of the revision function, whereas postulates (®2), (®3), (®4) and (®5), correspond to the specific cases 1, 2, 4 and 3, respectively, of the revision function, for obtaining the set $\Delta_{K®\alpha}$.

Theorem 2.2

If a contraction function is defined as a compromised contraction $\ominus$ in Definition 2.16, then the revision function ® obtained from Definition 2.17 satisfies the postulates (®1) to (®8).

Proof:

Considering the safe-maximality restrictions, by Definition 2.17, we have that:

In case 1, $\Delta_{K®\alpha} = \Delta_K$ and $K®\alpha = K$. So, postulate (®1) is trivially guaranteed. Postulate (®2) is directly satisfied by case 1 of the revision function definition. Postulates (®3), (®4) and (®5) do not apply for case 1. Postulate (®6) is satisfied, since $K®\alpha = K$ and $P_K \subseteq K®\alpha$. Postulate (®7) holds for case 1, as proved by Theorem 2.4.1. And Postulate (®8) trivially follows from case 1, such that if $\alpha \in \Delta_K$ or if $\vdash \neg\alpha$, and $\alpha \leftrightarrow \beta$, then it is guaranteed that $Cn(K ® \alpha) = Cn(K ® \beta) = Cn(K)$.

In case 2, $\Delta_{K®\alpha} = \Delta_K \cup \{\alpha\}$. So, postulate (®1) is guaranteed, since $K ® \alpha = \langle \Delta_{K®\alpha}, P_K \rangle$. Postulate (®3) follows from case 2, where $\Delta_{K®\alpha} = \Delta_K \bigcup \{\alpha\}$, in the case that $K + \alpha \nvdash \bot$. So, it is guaranteed that $K \circledast \alpha \subseteq K ® \alpha$ in this case. Postulates (®2), (®4) and (®5) do not apply for case 2. Postulate (®6) is satisfied, since $P_K \subseteq K®\alpha$. Postulate (®7) holds for case 2, as proved by Theorem 2.4.1. And Postulate (®8) follows from case 2, such that if $\Delta_{K®\alpha} = \Delta_K \cup \{\alpha\}$ and $\alpha \leftrightarrow \beta$, then it is guaranteed that $Cn(K ® \alpha) = Cn(K ® \beta)$.

In case 3, $\Delta_{K®\alpha} = \Delta_K \cup SmaxCI'(\alpha)$. So, postulate (®1) is guaranteed, since $K ® \alpha = \langle \Delta_{K®\alpha}, P_K \rangle$. Postulate (®5) is satisfied directly by case 3, where $\Delta_{K®\alpha} = \Delta_K \cup SmaxCI'(\alpha)$. Postulates (®2), (®3) and (®4) do not apply for case 3. Postulate (®6) is satisfied, since $P_K \subseteq K®\alpha$. Postulate (®7) holds for case 3, as proved by Theorem 2.4.1. And Postulate (®8) follows from case 3, such that if $\alpha \leftrightarrow \beta$, then the consistent consequences introduced by $SmaxCI'(\alpha)$ in $\Delta_{K®\alpha}$ will correspond to the ones in $SmaxCI'(\beta)$, and it is guaranteed that $Cn(K ® \alpha) = Cn(K ® \beta)$.

In case 4, $\Delta_{K®\alpha} = Smax(\Delta_K - R_\alpha)_{c_5} \cup SmaxCR'(R'_\alpha) \cup \{\alpha\}$. So, postulate (®1) is guaranteed, since $K ® \alpha = \langle \Delta_{K®\alpha}, P_K \rangle$. Postulate (®4) is satisfied by case 4, from where it follows directly that $\alpha \in \Delta_{K®\alpha}$ and that $\Delta_{K®\alpha} = Smax(\Delta_K - R_\alpha)_{c_5} \cup SmaxCR'(R'_\alpha)$ is defined according to this postulate, considering the restriction that $Smax(\Delta_K - R_\alpha)_{c_5}$ corresponds to $(\Delta_K - R_\alpha)$, under safe-maximality and condition c_5. Postulates (®2), (®3) and (®5) do not apply for case 3. Postulate (®6) is satisfied, since $P_K \subseteq K®\alpha$. Postulate (®7) holds also for case 4, as proved by Theorem 2.4.1. And Postulate (®8) follows from case 4, such that if $\alpha \leftrightarrow \beta$, then safe-maximality is considered as the only choice. That is, provided that R_α and R_β are guaranteed to have the same chosen-set.

Hence, we have shown that the revision function ® satisfies the postulates (®1) to (®8).

■

Theorem 2.3

Given a compromised contraction function $\ominus$, as in Definition 2.16, for any knowledge base $K = \langle \Delta, P_K \rangle$, and any sentence α, a revision function is a compromised revision ®, as in definition 2.17, if and only if it satisfies the postulates (®1) to (®8).

Proof:

In Theorem 2.2, we have shown that the revision function ® satisfies the postulates (®1) to (®8). This proves the if-case of this theorem statement. For the only-if-case, we need to show that in the cases that the safe-maximality restrictions apply, if the postulates (®1) to (®8) are satisfied, then the revision function is a compromised revision ®. Considering the safe-maximality restrictions, and the four cases of Definition 2.17, we have that: Postulate (®1) is trivially guaranteed. Postulate (®2) is directly satisfied by case 1 of the revision function definition. Postulate (®3) follows from case 2, where $\Delta_{K®\alpha} = \Delta_K \bigcup \{\alpha\}$, in the case that $K + \alpha \nvdash \bot$. So, it is guaranteed that $K \oplus \alpha \subseteq K ® \alpha$ in this case. Postulate (®4) is satisfied by case 4, from where it follows directly that $\alpha \in \Delta_{K®\alpha}$ and that $\Delta_{K®\alpha} = Smax(\Delta_K - R_\alpha)_{c_5} \cup SmaxCR'(R'_\alpha)$ is defined according to this postulate, considering the restriction that $Smax(\Delta_K - R_\alpha)_{c_5}$ corresponds to $(\Delta_K - R_\alpha)$, under safe-maximality and condition c_5. Postulate (®5) is satisfied directly by case 3, where $\Delta_{K®\alpha} = \Delta_K \cup SmaxCI'(\alpha)$. Postulate (®6) follows from the general definition of compromised revision for all the four cases, where $K ® \alpha = \langle \Delta_{K®\alpha}, P_K \rangle$. Postulate (®7) is proved to be satisfied by Theorem 2.4.1. Postulate (®8) follows from cases 1, 2 and 3, and for case 4 it also holds, when safe-maximality is considered as the only choice. That is, in the case that R_α and R_β are guaranteed to have the same chosen-set. This ends our proof.

■

2.4.3 Summary

We have presented our compromised revision approach within a belief revision perspective, via restricting existing revision specifications to our requirements. We have specified a base-theory change operator ® 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 propositions established relevant relationships among functions, postulates and some relations of this characterization.

We re-state that the compromised revision ® presented here, does not stand as an alternative method to the existing belief revision approaches, since it does not propose another general belief revision model. Instead, ® 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.

2.5 Discussions

In this section, we have reviewed the basic notions of belief revision and introduced a compromised characterization to revision, within the philosophy of our approach.

The main results of this chapter are concentrated on the propositions and theorems of section 2.4.2. There, we have shown that compromised revision preserves consistency, and we have established the correspondences between compromised revision and contraction, as well as between the revision function and the proposed postulates for compromised revision.

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,[34] and on an impartial solution, via the notion of safe-maximality, for the other retraction choices among conflicting data. In most cases, those conflicting data arise when generating candidates of compromised consequences to be added to the final revised base.[35]

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

In the case that the sentences on the base Δ_K are not comparable by the partial ordering, according to the application's requirements, 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.[36] Instead, we are offering a combined choice between the function and the application-oriented option, hoping for a more adequate result.

The high persistence results 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 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

[34] See Definitions 2.11, 2.12 and 2.13.

[35] See Definitions 2.9, and Remarks 2.6, 2.8, 2.10, 2.15, 2.18 and 2.20.

[36] In some cases where weak contraction functions, like the full meet for instance, are specified, the result can be very restrictive.

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 adopted a more impartial solution, which gets rid of the minimals which are involved in the failure of the condition in question, by identifying the common elements among the subsets of minimal elements.

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. Postulate (®9) would also be satisfied in this case.

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 could discard many more sentences than the minimal number needed from the set of compromised consequences, to satisfy the condition under question. 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 only the number of compromised consequences to be added to the revised base. In fact, this is the special feature that our approach proposes as a compromised solution, 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 chapter. Among those points, we can cite the problem of the computational tractability of compromised revisions.

3 Final Remarks

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

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 conceive 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 authors. In [Dar-96b], Dargam introduces a realization of the CIU system, using labelled databases, based on the framework of Labelled Deductive Systems [Gab-94]. That work states formally all the concepts defined for the compromised policies of the approach. Another formalization also studied by the author,[37] 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 [Dar-96d], we made a brief comparison with Truth Maintenance Systems. But a more detailed study on their differences, advantages and limitations, is also a planned future work.

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 some existing approaches to base revision, 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

During the development of this work, Fátima C. C. Dargam was supported by CNPq, Brazil, under the grant 202078/90.6. The authors are very grateful to Odinaldo Rodrigues, Sanjay Modgil and Hans Juergen Ohlbach, for important comments and suggestions on earlier versions of this work.

A Auxiliary Proofs

This appendix presents proofs of some propositions cited in section 2 of this paper.

Proposition A.1 (Proposition 2.1)

Given a base $K = \langle \Delta_K, P_K \rangle$, and an input sentence α, such that $\Delta_K \bigcup \{\alpha\} \vdash \bot$, and it is not the case that $\vdash \neg\alpha$, and the set $Fail(\Delta_K)_{c_1} = \{S_1, S_2, \cdots, S_l\}$, it is sufficient to retract

[37] See [Dar-96a].

one element from each $S_i \subseteq Fail(\Delta_K)_{c_1}$*, such that* $(\Delta_K - \mathcal{S}(Fail(\Delta_K)_{c_1})) \bigcup \{\alpha\} \nvdash \perp$*, where* $\mathcal{S}(Fail(\Delta_K)_{c_1}) = \{x_i \mid \forall S_i \in Fail(\Delta_K)_{c_1}, \exists y \in S_i, \text{such that } x_i = y\}$.

Proof:

By Remark 2.5, each set $S_i \in Fail(\Delta_K)_{c_1}$ is minimal w.r.t. $\subseteq$, such that $S_i \bigcup \{\alpha\} \vdash \perp$. Then, for any element $y \in S_i$, $(S_i - \{y\}) \bigcup \{\alpha\} \nvdash \perp$. This is true for all $S_i \in Fail(\Delta_K)_{c_1}$. $\mathcal{S}(Fail(\Delta_K)_{c_1})$ is a set that contains one element of each $S_i \in Fail(\Delta_K)_{c_1}$, for $i = 1, \cdots, l$. Therefore, it is guaranteed that $(\Delta_K - \mathcal{S}(Fail(\Delta_K)_{c_1})) \bigcup \{\alpha\} \nvdash \perp$.

■

Proposition A.2 (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* $\vdash \neg\alpha$*,* $Smax(\Delta_K)_{c_1} + \alpha \nvdash \perp$.

Proof:

By Definition 2.9 and Remark 2.6, if $Smax(\Delta_K)_{c_1} = \Delta_K$, the statement of the proposition is already guaranteed by condition c_1. In the case that $Smax(\Delta_K)_{c_1} = \Delta_K - RMin(Fail(\Delta_K)_{c_1})$, we have that in Proposition 2.1, we have proved that $(\Delta_K - \mathcal{S}(Fail(\Delta_K)_{c_1})) \bigcup \{\alpha\} \nvdash \perp$, where $\mathcal{S}(Fail(\Delta_K)_{c_1})$ is any set which includes one element of each set $S_i \in Fail(\Delta_K)_{c_1}$, for $i = 1, \cdots, l$. By Definition 2.7, $RMin(Fail(\Delta_K)_{c_1})$ contains at least one minimal element w.r.t. $\leqslant$, of each set S_i of $Fail(\Delta_K)_{c_1}$. Hence, there exists a set $\mathcal{S}(Fail(\Delta_K)_{c_1})$, such that $\mathcal{S}(Fail(\Delta_K)_{c_1}) \subseteq RMin(Fail(\Delta_K)_{c_1})$. Then, by Proposition 2.1, we have that $(\Delta_K - RMin(Fail(\Delta_K)_{c_1})) \bigcup \{\alpha\} \nvdash \perp$. Hence, also when we have that $Smax(\Delta_K)_{c_1} = \Delta_K - RMin(Fail(\Delta_K)_{c_1})$, it is guaranteed that $Smax(\Delta_K)_{c_1} \bigcup \{\alpha\} \nvdash \perp$.

■

Proposition A.3 (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* $\vdash \neg\alpha$*,* $Smax(CI(\alpha))_{c_2} \bigcup \Delta_K \bigcup P_K \nvdash \perp$.

Proof:

By Definition 2.9 and Remark 2.8, in the case that $Smax(CI(\alpha))_{c_2} = CI(\alpha)$, the statement of the proposition is already guaranteed by condition c_2. In the case that $Smax(CI(\alpha))_{c_2} = CI(\alpha) - RMin(Fail(CI(\alpha))_{c_2})$, we have by Definition 2.8 and Remark 2.7, that each set $S_i \in Fail(CI(\alpha))_{c_2}$ is a minimal subset of $(CI(\alpha)$, such that $S_i \bigcup \Delta_K \vdash \perp$. Then, for any element $y \in S_i$, $(S_i - \{y\}) \bigcup \Delta_K \nvdash \perp$. This is true for all $S_i \in Fail(CI(\alpha))_{c_2}$. By Definition 2.7, $RMin(Fail(CI(\alpha))_{c_2})$ contains at least one minimal element w.r.t. $\leqslant$, of each set S_i of $Fail(CI(\alpha))_{c_2}$, for $i = 1, \cdots, m$.

Therefore, it is guaranteed that $(CI(\alpha) - RMin(Fail(CI(\alpha))_{c2})) \bigcup \Delta_K \bigcup P_K \nvdash \bot$, Hence, also when $Smax(CI(\alpha))_{c2} = CI(\alpha) - RMin(Fail(CI(\alpha))_{c2})$, it is guaranteed that $Smax(CI(\alpha))_{c2} \bigcup \Delta_K \bigcup P_K \nvdash \bot$.

■

Proposition A.4 (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, and it is not the case that $\vdash \neg\alpha$, $Smax(Smax(CI(\alpha))_{c2})_{c3} \bigcup \Delta_K \nvdash \alpha$.

Proof:

By Remark 2.10, in the case that $Smax(Smax(CI(\alpha))_{c2})_{c3} = Smax(CI(\alpha))_{c2}$, the statement of the proposition is already guaranteed by condition c_3. In the case that $Smax(Smax(CI(\alpha))_{c2})_{c3} = Smax(CI(\alpha))_{c2} - RMin(Fail(Smax(CI(\alpha))_{c2})_{c3})$, we have that by Definition 2.8, and Remark 2.9, each set $S_i \in Fail(Smax(CI(\alpha))_{c2})_{c3}$ is a minimal subset of $Smax(CI(\alpha))_{c2}$, such that $S_i \bigcup \Delta_K \vdash \alpha$. Then, for any element $y \in S_i$, $(S_i - \{y\}) \bigcup \Delta_K \nvdash \alpha$. This is true for all $S_i \in Fail(Smax(CI(\alpha))_{c2})_{c3}$. By Definition 2.7, $RMin(Fail(Smax(CI(\alpha))_{c2})_{c3})$ contains at least one minimal element w.r.t. $\leqslant$, of each set S_i of $Fail(Smax(CI(\alpha))_{c2})_{c3}$, for $i = 1, \cdots, n$. Therefore, it is guaranteed that $Smax(CI(\alpha))_{c2} - RMin(Fail(Smax(CI(\alpha))_{c2})_{c3}) \bigcup \Delta_K \nvdash \alpha$. Hence, also when we have that $Smax(Smax(CI(\alpha))_{c2})_{c3} = Smax(CI(\alpha))_{c2} - RMin(Fail(Smax(CI(\alpha))_{c2})_{c3})$, it is guaranteed that $Smax(Smax(CI(\alpha))_{c2})_{c3} \bigcup \Delta_K \nvdash \alpha$.

■

Proposition A.5 (Proposition 2.5)

Given a base $K = \langle \Delta_K, P_K \rangle$, and an input sentence α, such that it is not the case that $\vdash \neg\alpha$ nor that $\{\alpha\} \bigcup P_K \vdash \bot$. If $\{\alpha\} \bigcup K \vdash \bot$, then given a set $\bot_\alpha = \{S_1, S_2, \cdots, S_n\}$, it is sufficient to retract one element of each set $S_i \in \bot_\alpha$, for $i = 1, \cdots, n$, from $\Delta_K \cup \{\alpha\}$ such that $((\Delta_K \cup \{\alpha\}) - S(\bot_\alpha)) \cup P_K \nvdash \bot$, where $S(\bot_\alpha) = \{x_i \mid \forall S_i \in \bot_\alpha, \exists y \in S_i$, such that $x_i = y\}$.

Proof:

By Definition 2.8 and Remark 2.11, each $S_i \in \bot_\alpha$ is a minimal subset of Δ_K, such that $S_i \bigcup \{\alpha\} \bigcup P_K \vdash \bot$. Then, for any element $y \in S_i$, if y is retracted from S_i, we have that $(S_i - \{y\}) \bigcup \{\alpha\} \bigcup P_K \nvdash \bot$. This is true for all $S_i \in \bot_\alpha$. $S(\bot_\alpha)$ is a set which contains one element of each $S_i \in \bot_\alpha$. Therefore, it is guaranteed that $(\Delta_K - S(\bot_\alpha)) \bigcup \{\alpha\} \bigcup P_K \nvdash \bot$.

■

Proposition A.6 (Proposition 2.6)

Given a base $K = \langle \Delta_K, P_K \rangle$*, for any set* $X \subseteq \Delta_K$*, we have that* $Smax(\Delta_K - X)_{c5} \nvdash x, \forall x \in X$.

Proof:

By Definition 2.9 and Remark 2.15, in the case that $Smax(\Delta_K - X)_{c5} = (\Delta_K - X)$, the statement of the proposition is already guaranteed by condition c_5. In the case that $Smax(\Delta_K - X)_{c5} = (\Delta_K - X) - RMin(Fail(\Delta_K - X)_{c5})$, we have that by Definition 2.8 and by Remark 2.14, each set $S_i \in Fail(\Delta_K - X)_{c5}$ is a minimal subset of $(\Delta_K - X)$, satisfying the condition $(\Delta_K - X) \vdash x$, for any $x \in X$. Then, for any element $y \in S_i$, $(S_i - \{y\}) \nvdash x$, $\forall x \in X$. This is true for all $S_i \in Fail(\Delta_K - X)_{c5}$. By Definition 2.3, the set $min(Fail(\Delta_K - X)_{c5})$ contains at least one element of each $S_i \in Fail(\Delta_K - X)_{c5}, for$ i $= 1, \cdots, k$. And by Definition 2.7, $RMin(Fail(\Delta_K - X)_{c5})$ contains at least one element of each set $min(S_i)$ of $min(Fail(\Delta_K - X)_{c5})$. Hence, also when $Smax(\Delta_K - X)_{c5} = (\Delta_K - X) - RMin(Fail(\Delta_K - X)_{c5})$, it is guaranteed that $Smax(\Delta_K - X)_{c5} \nvdash x$, $\forall x \in X$.

■

Proposition A.7 (Proposition 2.7)

Given a base $K = \langle \Delta_K, P_K \rangle$*, for any set* $X' \subseteq \Delta_K$*, such that* $CR(X')$ *is non-empty,* $Smax(CR(X'))_{c6} \cup Smax(\Delta_K - X)_{c5} \nvdash x, \forall x \in X'$.

Proof:

By Definition 2.9 and Remark 2.18, if $Smax(CR(X'))_{c6} = CR(X')$, the statement of the proposition is already guaranteed by condition c_6. In the case that $Smax(CR(X'))_{c6} = CR(X') - RMin(Fail(CR(X'))_{c6})$, we have that by Definition 2.8 and Remark 2.17, each set $S_i \in Fail(CR(X'))_{c6}$ is a minimal subset of $CR(X')$, satisfying the condition $S_i \bigcup Smax(\Delta_K - X)_{c5} \vdash x$, for any $x \in X'$. Then, for any element $y \in S_i$, $(S_i - \{y\}) \bigcup Smax(\Delta_K - X)_{c5} \nvdash x$, $\forall x \in X'$. This is true for all $S_i \in Fail(CR(X'))_{c6}$. By Definition 2.3, the set $min(Fail(CR(X'))_{c6})$ contains at least one element of each $S_i \in Fail(CR(X'))_{c6}$, for $i = 1, \cdots, k$. By Definition 2.6, $RMin(Fail(CR(X'))_{c6})$ contains at least one element of each set $min(S_i)$ of $min(Fail(CR(X'))_{c6})$. Hence, also when $Smax(CR(X'))_{c6} = CR(X') - RMin(Fail(CR(X'))_{c6})$, it is guaranteed that $Smax(CR(X'))_{c6} \cup Smax(\Delta_K - X)_{c5} \nvdash x$, $\forall x \in X'$.

■

Proposition A.8 (Proposition 2.8)

Given a base $K = \langle \Delta_K, P_K \rangle$*, and an input sentence* α*, such that it is not the case that* $\vdash \neg\alpha$*, and a set* R_α*, as in Definition 2.13, such that* $Smax(CR(R'_\alpha)_{c6}$ *is non-empty. If* $P_K \bigcup \{\alpha\} \nvdash \bot$ *and* $K \bigcup \{\alpha\} \vdash \bot$*,* $Smax(Smax(CR(R'_\alpha))_{c6})_{c7} \bigcup (Smax(\Delta_K - R_\alpha)_{c5} \cup \{\alpha\}) \bigcup P_K \nvdash \bot$.

Proof:

By Definition 2.20, in the case that $Smax(Smax(CR(R'_\alpha))_{c6})_{c7} = Smax(CR(R'_\alpha))_{c6}$, the statement of the proposition is already guaranteed by condition c_7. When $Smax(Smax(CR(R'_\alpha))_{c6})_{c7} = Smax(CR(R'_\alpha))_{c6} - RMin(Fail(Smax(CR(R'_\alpha))_{c6})_{c7})$, we have that by Remark 2.19, each set $S_i \in Fail(Smax(CR(R'_\alpha))_{c6})_{c7}$ is minimal w.r.t. $\subseteq$, satisfying the condition $S_i \bigcup (Smax(\Delta_K - R_\alpha)_{c5} \cup \{\alpha\}) \bigcup P_K \vdash \bot$. Then, for any $y \in S_i$, $(S_i - \{y\}) \bigcup (Smax(\Delta_K - R_\alpha)_{c5} \cup \{\alpha\}) \bigcup P_K \nvdash \bot$. This is true for all $S_i \in Fail(Smax(CR(R'_\alpha))_{c6})_{c7}$. By Definition 2.3, the set $min(Fail(Smax(CR(R'_\alpha))_{c6})_{c7})$ contains at least one element of each of the sets $S_i \in Fail(Smax(CR(R'_\alpha))_{c6})_{c7}$, for $i = 1, \cdots, o$.
And by Definition 2.7, $RMin(Fail(Smax(CR(R'_\alpha))_{c6})_{c7})$ contains at least one element of each set $min(S_i)$ of minimal elements in $min(Fail(Smax(CR(R'_\alpha))_{c6})_{c7})$. Hence, also in the case that $Smax(Smax(CR(R'_\alpha))_{c6})_{c7} = Smax(CR(R'_\alpha))_{c6} - RMin(Fail(Smax(CR(R'_\alpha))_{c6})_{c7})$, it is guaranteed that
$Smax(Smax(CR(R'_\alpha))_{c6})_{c7} \bigcup (Smax(\Delta_K - R_\alpha)_{c5} \cup \{\alpha\}) \bigcup P_K \nvdash \bot$.

∎

Proposition A.9 (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 $\vdash \neg\alpha$, and $CI(\alpha)$ is non-empty, if $P_K + \alpha \vdash \bot$, then $\forall x \in Smax(Smax(CI(\alpha))_{c2})_{c3}$, $K \nvdash x$ and $K ® \alpha \vdash x$.

Proof:

By Definition 2.10, we guarantee that all the elements in $CI(\alpha)$ are not originally present or derived from K. By Definition 2.9 and Remark 2.10, we have that $Smax(Smax(CI(\alpha))_{c2})_{c3} \subseteq CI(\alpha)$. And by Definition 2.17, we state that in the case that $P_K + \alpha \vdash \bot$, $K ® \alpha = \langle \Delta_{K®\alpha}, P_K \rangle$, where $Smax(Smax(CI(\alpha))_{c2})_{c3} \subseteq \Delta_{K®\alpha}$. Hence, $\forall x \in Smax(Smax(CI(\alpha))_{c2})_{c3}$, $K \nvdash x$, and $K ® \alpha \vdash x$.

∎

Proposition A.10 (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 $K + \alpha \vdash \bot$ and $P_K + \alpha \nvdash \bot$, then $\forall x \in Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$, $K \vdash x$ and also $K ® \alpha \vdash x$.

Proof:

By Definition 2.9 and Remark 2.20, $Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$ is a safe maximal subset of $CR(R'_\alpha)$, relative to conditions c_6 and c_7. So, $\forall y \in Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$, $K \vdash y$, since $CR(R'_\alpha)$, is a set of consequences of R'_α and by Definition 2.13 and Remark 2.11, $R_\alpha \subseteq \bot_\alpha$, and $\bot_\alpha \subseteq \Delta_K$. By Definition 2.17, in the case that $K + \alpha \vdash \bot$ and $P_K + \alpha \nvdash \bot$, we state that $K ® \alpha = \langle \Delta_{K®\alpha}, P_K \rangle$, where $Smax(Smax(CR(R'_\alpha))_{c6})_{c7} \subset \Delta_{K®\alpha}$. Hence,

$\forall x \in Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$, $K ® \alpha \vdash x$. Therefore, we guarantee that $\forall x \in Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$, both K $\vdash x$ and $K ® \alpha \vdash x$.

■

Proposition A.11 (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 $K + \alpha \vdash \bot$ and $P_K + \alpha \nvdash \bot$, then $\forall x \in R'_\alpha$, $K ® \alpha \nvdash x$.

Proof:

By Definition 2.14, $R'_\alpha = R_\alpha \bigcup RMin(Fail(\Delta_K - R_\alpha)_{c5})$, where $RMin(Fail(\Delta_K - R_\alpha)_{c5})$ is a subset of $\Delta_K - R_\alpha$ and $R_\alpha \nsubseteq \Delta_K - R_\alpha$. By Definition 2.17, in the case that $K + \alpha \vdash \bot$ and $P_K + \alpha \nvdash \bot$, we state that $K ® \alpha = \langle \Delta_{K®\alpha}, P_K \rangle$, where $\Delta_{K®\alpha} = Smax(\Delta_K - R_\alpha)_{c5} \cup Smax(Smax(CR(R'_\alpha))_{c6})_{c7} \bigcup \{\alpha\}$. By Definition 2.9 and Remark 2.15, $R_\alpha \nsubseteq Smax(\Delta_K - R_\alpha)_{c5}$ and it is shown in Proposition 2.6 that $Smax(\Delta_K - R_\alpha)_{c5} \nvdash x$, $\forall x \in R_\alpha$. By Definition 2.9 and Remark 2.20, $Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$ is a subset of $Smax(CR(R'_\alpha))_{c6}$, and Proposition 2.7 has proved that $Smax(CR(R'_\alpha))_{c6} \cup Smax(\Delta_K - R_\alpha)_{c5} \nvdash x$, $\forall x \in R'_\alpha$. Since R_α contains the elements that have to be retracted from Δ_K in order to accomplish the insertion of α and keep consistency, it is then also guaranteed that $\Delta_{K®\alpha} \nvdash x$, $\forall x \in R_\alpha$. Hence, $K®\alpha \nvdash x$, $\forall x \in R'_\alpha$.

■

Proposition A.12 (Proposition 2.12)

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 $\beta \in K®\alpha$, then either $\beta \in \Delta_K$ and $\beta \notin R'_\alpha$; or $\beta \notin \Delta_K$ and either $\beta \in Smax(Smax(CI(\alpha))_{c2})_{c3}$, or $\beta = \alpha$, or $\beta \in Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$.

Proof:

By Definition 2.17, we state that $K®\alpha = \langle \Delta_{K®\alpha}, P_K \rangle$, where in case 1, $\Delta_{K®\alpha} = \Delta_K$. This means that $\forall \beta \in K®\alpha$, $\beta \in \Delta_K$. For case 2, $\Delta_{K®\alpha} = \Delta_K \bigcup \{\alpha\}$. In this case, $\forall \beta \in K®\alpha$, either $\beta \in \Delta_K$ or $\beta = \alpha$. In case 3, $\Delta_{K®\alpha} = \Delta_K \bigcup Smax(Smax(CI(\alpha))_{c2})_{c3}$. This means that $\forall \beta \in K®\alpha$, either $\beta \in \Delta_K$ or $\beta \in Smax(Smax(CI(\alpha))_{c2})_{c3}$. Finally, for case 4, $\Delta_{K®\alpha} = Smax(\Delta_K - R_\alpha)_{c5} \cup Smax(Smax(CR(R'_\alpha))_{c6})_{c7} \bigcup \{\alpha\}$. By Definition 2.9 and Remark 2.15, $Smax(\Delta_K - R_\alpha)_{c5}$ is equivalent to $\Delta_K - R'_\alpha$, since $R'_\alpha = R_\alpha \bigcup RMin(Fail(\Delta_K - R_\alpha)_{c5})$. Then, $\forall \beta \in K®\alpha$, either $\beta \in \Delta_K$ and $\beta \notin R'_\alpha$, or $\beta \in Smax(Smax(CR(R'_\alpha))_{c6})_{c7}$. Hence, all the four cases of Definition 2.17 satisfy the proposition statement above.

■

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.

[Ban-90] R. B. Banerji, ed., *"Formal Techniques in Artificial Intelligence : a sourcebook"*, 1990.

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

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

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

[Bre-91] G. Brewka, *"Non-Monotonic Reasoning: Logical Foundations of Common Sense"*, Cambridge University Press, 1991.

[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 Proceedings of AI: Methodology, Systems and Applications, 1990.

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

[Dar-89] F.C.C. Dargam, *"Sistema Especialista Aplicado a Guerra Eletronica"*, (An Expert System applied to Electronic Warfare), Masters Thesis, IME-RJ SE/9, Instituto Militar de Engenharia, Rio de Janeiro, Brasil, March 1989.

[Dar-91] F.C.C. Dargam; E.L. Passos; F.R. Pantoja, *"Decision Support Systems for Military Applications"*, European Journal of Operational Research (EJOR), (special issue on DSS), 1991.

[Dar-93] F.C.C. Dargam, *"Resolving Conflicting Actions and Updates"*, Transfer Report, Department of Computing, Imperial College, August 1993, (available by anonymous ftp to *theory.doc.ic.ac.uk*, in the file *papers/dargam/tr-rep93.ps*).

[Dar-96a] F.C.C. Dargam, *"Compromised Updates in Deductive Databases"*, Research Report (available by anonymous ftp to *theory.doc.ic.ac.uk*, in the file *papers/Dargam/dar96a.ps*).

[Dar-96b] F.C.C. Dargam, *"On Compromising Updates in Labelled Databases"*, Imperial College Research Report DoC 96/1, (available by anonymous ftp to *theory.doc.ic.ac.uk*, in the file *papers/Dargam/dar96b.ps*).

[Dar-96d] 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), in Proceedings of the Compulog Net Meeting on Knowledge Representation and Reasoning CNKRR'93, Lisbon, July 1993.

[Doy-79] J. Doyle, *"A Truth Maintenance System"*, Artificial Intelligence 12, pp 231-272, North Holland Pub. Co., 1979.

[Elk-90] C. Elkan, *"A Rational Reconstruction of Non-monotonic Truth Maintenance Systems"*, Artificial Intelligence 43(2), pp 219-234, North Holland Pub. Co., 1990.

[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 vs. 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 Proceedings of 2nd Conference on Theorectical Aspects of Reasoning about Knowledge, Edited by M. Vardi, Morgan Kaufmann, 1988.

[GäRo-95] P. Gärdenfors, H. Rott, *"Belief Revision"*, in the Handbook of Logic in Artificial Intelligence and Logic Programming, Vol IV, Epistemic and Temporal Reasoning, Oxford University Press, 1995.

[GGMO-94] D. Gabbay; L. Giordano; A. Martelli; N. Olivetti, *"Conditional Logic Programming"*, in Proceedings of the ICLP'94, International Conference of Logic Programming, 1994.

[GiMa-90] L. Giordano; A. Martelli, *"An Abductive Characterization of the TMS"*, in LNAI 515, J.P. Martins & M. Reinfrank (Eds.), TMS, ECAI-90 Workshop Proceedings, 1990.

[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.

[Han-91] S. Hansson, *"In Defense of Base Contraction"*, Synthese, 1991.

[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-89] H. Katsuno, A.O. Mendelzon, *"A Unified View of Propositional Knowledge Base Updates"*, In Proceedings of the 11th. International Joint Conference on Artificial Intelligence, 1989.

[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.

[Kle-86a] J.d. Kleer, *"An Assumption-based TMS"*, Artificial Intelligence 28, pp 127-162, North Holland Pub. Co., 1986.

[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-85] D. Makinson, *"How to give it up"*, Synthese 62, 1985.

[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.

[MaWa-87] S. Manchanda; D.S. Warren, *"A Logic-based Language for Database Updates"*, in Foundations of Deductive Databases and Logic Programming, J. Minker, editor, Morgan-Kaufman, Palo ALto, CA, 1987.

[BaSu-92] J. Minker; C. Baral; S. Kraus; V. S. Subrahmanian, *"Combining Knowledge Bases consisting of First Order Theories"*, Computational Intelligence, 8(1), 1992.

[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.

[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 Proceedings of the 1st. 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.

[Neb-91] B. Nebel, *"Belief Revision and Default Reasoning: Syntax-Based Approaches"*, in Proceedings of the 2nd. International Conference on Principles of Knowledge Representation and Reasoning, 1991.

[Nil-80] N. J. Nilsson, *"Principles of Artificial Intelligence"*, Tioga, Palo Alto, CA.,1980.

[PaDe-90] D. Partridge; Y. Wilks, ed., *"The Foundations of Artificial Intelligence : a sourcebook"*, 1990.

[PeWo-92] P. Peppas, W. Wobcke, *"On the use of Epistemic Entrenchment in Reasoning about Actions"* in Proc. of the 10th. ECAI, ECAI-92, edited by B. Neumann, 1992.

[PiCu-89] *"A Truth Maintenance System Based on Stable Models"*, S. G. Pimentel; J. L. Cuadrado, in Proceedingsof the North American Conference on Logic Programming NACLP, Cleveland, pp 274-290, 1989.

[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.

[Ric-89] M. Richer, ed., *"AI Tools and Techniques"*, 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.

[RoPi-91] *"A Nonmonotonic Assumption-Based TMS using Stable Bases"*, W. Rodi; S. Pimentel, in Proceedings of KR'91, pp 485-495, 1991.

[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.

[SaIw-91] K. Satoh; N. Iwayama, *"Computing Abduction by using TMS"*, April 1991.

[Seg-94] K. Segerberg, *"Belief Revision from the Point of View of Doxastic Logic"*, Bulletin of IGPL, Vol. 2, n.3, pp. 3-21, 1994.

[Sha-92] S. Shapiro, ed., *"Encyclopedia of Artificial Intelligence"*, 2nd. edition, 1992.

[Sri-93] S. Sripada, *"A Temporal Approach to Belief Revision in Knowledge Bases"*, in Proceedings of the 9th. IEEE Conference on AI for Applications, CAIA-93, Florida, March 1993.

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

[Wag-94] G. Wagner, *"Vivid Logic"*, Lecture Notes in Artificial Intelligence, LNAI 764, Springer Verlag, 1994.

[WaCh-91] X. Wang; H. Chen, *"On the Semantics of TMS"*, in Proceedings of the IJCAI '91, pp 306-309, 1991.

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

[Win-92] P. Winston, *"Artificial Intelligence"*, 3rd edition, 1992.