Stellingen

behorend bij het proefschrift

A General Framework for Compositional Reasoning with Uncertainty

Bob Goedhart

28 oktober 1994

1. Compositioneel redeneren met onzekerheid vereist een definitie van *contradictie* die niet rechtstreeks kan worden afgeleid uit de onderliggende driewaardige logica. (Hoofdstuk 3 van dit proefschrift)

2. De door Shortliffe en Buchanan voorgestelde definitie van de *certainty factor* combineert het nadeel van de strenge randvoorwaarden die een probabilistische onzekerheidsmaat oplegt aan het gebruik van de propagatiefuncties, met het ontbreken van een heldere semantiek. (zie Definities 4.3, 4.4 en 4.5 in dit proefschrift)

3. Het in convergerende en divergerende verklaringen gebruikte principe voor onzekerheidspropagatie dient te worden gebruikt bij het ontwikkelen van flexibele oplossingsstrategieën voor zoekproblemen (Hoofdstuk 5 van dit proefschrift)

4. Taalconstructies zoals deze binnen Delfi3 zijn ontwikkeld dienen te worden opgenomen in bestaande vierde-generatie programmeertalen. (Hoofdstuk 6 van dit proefschrift)

5. Kennissystemen dienen in de toekomst niet alleen gebruikt te worden om besluitvormingsprocedures ten aanzien van complexe technologische vraagstukken te ondersteunen, maar vooral ook om gemaakte keuzes achteraf te kunnen evalueren.

6. De term *Informatica* in de naam van een faculteit kan niet worden gelegitimeerd door voldoende eigen wetenschappelijke theorievorming.

7. Het veelgeprezen ontbreken van enige vorm van centraal gezag binnen Internet verhindert schaalvergroting van de gebruikersgroep.

8. Het door alle politieke partijen zo gewenste maatschappelijk debat over normen en waarden in de samenleving dient te worden voorafgegaan door een politiek debat waarin het afschaffen van de militaire dienstplicht wordt gekoppeld aan de invoering van een sociale dienstplicht.

9. Het gebruik van het begrip *fuzzy* in de omschrijving van consumentenelektronica slaat vooralsnog voornamelijk op de gebruiksaanwijzing.

10. Het grootste probleem voor arbiters van waterpolowedstrijden is niet zozeer de interpretatie van een spelregel als wel de *matching* van de juiste regelpremisse met de gegeven spelsituatie.

11. In plaats van maatschappijkritische stellingen betreffende het milieu te poneren, doet een promovendus er beter aan het proefschrift te laten drukken op kringlooppapier.

A General Framework

for

Compositional Reasoning with Uncertainty

Bob Goedhart

A General Framework

for

Compositional Reasoning with Uncertainty

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College van Dekanen aangewezen,
op vrijdag 28 oktober 1994 te 10.30 uur
door

Bob Goedhart

wiskundig ingenieur,

geboren te Delft.

Dit proefschrift is goedgekeurd door de promotor: Prof.dr.ir. E. Backer,
en de toegevoegd promotor: Dr.ir. J.C.A. van der Lubbe.

Promotiecommissie:
Rector Magnificus
Prof.dr.ir. E. Backer (promotor)
Dr.ir. J.C.A. van der Lubbe (toegevoegd promotor)
Prof.dr. H.J. van den Herik (Rijksuniversiteit Limburg)
Prof.dr. H. Koppelaar (Technische Universiteit Delft)
Prof.dr. J-J.Ch. Meyer (Rijksuniversiteit Utrecht)
Dr.ir. H. de Swaan Arons (Erasmus Universiteit Rotterdam)
Prof.ir. H.B. Verbruggen (Technische Universiteit Delft)

CIP-DATA KONINKLIJKE BIBLIOTHEEK DEN HAAG

Goedhart, Bob

A general framework for compositional reasoning with
uncertainty / Bob Goedhart. - Delft : Technische
Universiteit Delft, Faculteit der Elektrotechniek
Thesis Technische Universiteit Delft. - With ref. -
With summary in Dutch.
ISBN 90-5326-019-6
Subject headings: reasoning systems; uncertainty

Summary

Over the past few decades, Artificial Intelligence (AI) has evolved into a serious field of scientific research. A topic that has received a considerable amount of attention within AI research is the development of general-purpose reasoning techniques that can be used in various fields of application. The emphasis of this thesis is on *compositional reasoning*, a general-purpose reasoning technique in which elementary reasoning steps (the compositional reasoning primitives) are combined into complex reasoning processes. Compositional reasoning is the most widely used strategy in existing knowledge-based systems.

The representation and propagation (or *management*) of *uncertainty* in compositional reasoning is the central theme in this thesis. Throughout the years, the development of real-world AI applications has shown the necessity for the ability to cope with uncertainty in knowledge-based systems. Uncertainty has been used to model different aspects of knowledge, since knowledge can be incomplete, imprecise, invalid and unreliable. This thesis describes a general framework for reasoning with these aspects of uncertainty in a compositional setting, which satisfies a set of design constraints formulated for this type of reasoning.

The general framework

The development of a general framework for compositional reasoning with uncertainty is an evidential first step in the process. The framework is general in the sense that no assumptions are made with respect to the reasoning type, the uncertainty model and the field of application. A three-valued logic, of *true*, *false* and *unknown*, is used as the underlying mathematical theory, allowing the explicit representation (label unknown) of the incompleteness aspect. Clearly, classical two-valued logic does not support this capability. Since other aspects of uncertainty do not necessitate a conceptual change to a higher-order logic, the use of a three-valued logic suffices. However, many-valued logics can be used in the framework when they are considered to model an uncertainty measurement.

Formal definitions are provided for domain, inference, reason maintenance and uncertainty model (the basic structures of a knowledge-based system). The domain model is concerned with the representation aspect of compositional reasoning. A hierarchical approach is used, consisting of an expression, an inference (containing the compositional reasoning primitive) and a control level. The inference model describes the reasoning strategy for this formalism. The reason maintenance model contains information on previously executed reasoning patterns. According to the literature, these patterns can be used to support the inference model in avoiding the rediscovering of similar patterns and early detection of conflicts (invalid reasoning patterns).

Uncertainty is introduced into the framework by extending the three-valued labelling to an uncertainty type. The specification of this uncertainty type, accompanied by a set of propagation functions for the compositional reasoning primitives, is characteristic for each individual uncertainty model. Embedding this uncertainty type into the three-valued logic provides a fundamental basis for uncertainty representation and guarantees the correctness of uncertainty propagation in the boundary situations. The presence of a reason maintenance model in the framework enables a structural approach towards situation-specific propagation of uncertainty.

When the framework uses a deductive interpretation for compositional reasoning primitives, all existing deductive uncertainty models can be shown to be a special case of the framework. In a broader sense, it is argued that the framework offers the most general formal description of uncertain reasoning in a compositional setting.

The model of belief and disbelief

The certainty factor model is the most well-known and frequently used model for compositional uncertain reasoning when using a deductive reasoning strategy. Former analyses concerning the certainty factor model have been definite in their rejection of the model. According to these analyses, the main reason for this rejection is the violation of the suggested probabilistic foundation by the propagation functions of the model. However, the certainty factor model has attractive characteristics, for instance its computationally simple propagation functions, and has produced satisfactory results throughout the years in various applications. As a preliminary exercise, a part of the model is analysed again, thus clarifying the causes of its misbehaviour in specific situations.

Some results of this analysis have been used in the development of a new uncertainty model for compositional deductive reasoning, called the model of belief and disbelief. Uncertainty is represented by a two-dimensional vector that reflects both the evidence in favour of and against a domain variable. A set of propagation functions is provided for these belief vectors, which uses operators that adapt to the situation they are used in. The model was developed in isolation first, and was embedded into the general framework afterwards.

The information stored in a reason maintenance model not only enables the enhancement of the inference model, it can also be used to construct more advanced uncertainty propagation functions. This idea has been used in the design of a set of enhanced propagation functions for the model of belief and disbelief that are sensitive to the situation in which they are used. These functions are able to handle a type of problem that cannot be solved within existing deductive uncertainty models. The computational complexity of this approach is, although exponential in nature, acceptable for practical applications.

The appropriateness of implementing this model of belief and disbelief is demonstrated by adding it to an existing environment for knowledge-based system development, Delfi3. The knowledge representation formalism of this environment was mapped onto the frameworks domain and inference model. Further, a reason maintenance model was developed for Delfi3, which enables the implementation of the uncertainty propagation functions. This reason maintenance model has also been used to increase the performance of the inference model of Delfi3. The implementation underlines the practical profits derived from the developed theory.

Explanatory reasoning

Existing compositional architectures use a *deductive* problem-solving strategy. However, the use of *abductive* reasoning is increasingly acknowledged in the AI research community. Abductive reasoning is represented by the following scheme.

$$\begin{array}{ll} \text{Fact:} & B' \\ \text{Relation:} & A \rightarrow B \\ \hline \text{Explanation:} & A' \end{array}$$

It has been acknowledged that this kind of reasoning is often used in the human reasoning process. Abductive reasoning enables more flexibility in reasoning by assuming explanations that can be retracted in later stages of the reasoning process. A number of formalisms have been developed to support this type of reasoning, however, none of them uses a compositional architecture. The extent to which an abductive strategy can be used in compositional uncertain reasoning has been investigated, this thesis reports on the results of this investigation.

Because of its assumptive characterisation, an abductive reasoning strategy is more difficult to integrate into the framework than a truth-preserving, deductive strategy. Informal definitions of abductive reasoning mention the generation of the best or most plausible explanation(s) for the observations. In this thesis, a restricted abductive interpretation of the reasoning primitives is introduced for the compositional reasoning environment. Within the scope of this thesis, this interpretation is called *explanatory reasoning*.

An explanatory interpretation, which states that a set of explanations is assumed to have caused some given observation, is defined for the compositional reasoning primitive. Two types of explanations are distinguished: *converging* and *diverging* explanations. In a converging explanation the assumed explanations must all be valid at the same time to explain the observation, where in a diverging explanation the validity of at least one explanation suffices.

Using uncertainty makes explanatory reasoning even more interesting. In this, the explanatory interpretation of a compositional reasoning primitive provides an initial distribution of belief for an explanation. When additional information for this explanation is derived by the reasoning process, the type of the explanation is used to update this initial distribution. Suppose, in a diverging explanation, an increase in belief is observed for an explanation, then it seems plausible that belief in the remaining explanations decreases. This behaviour of diverging explanations is called *explain away*. Similarly, an increase in belief for one of the converging explanations will increase the belief in the remaining explanations as well. Being contrary to the concept of explain away, this behaviour is called *joint explain*.

This, more or less informal, description of explanatory reasoning has been formally described in terms of the developed framework. Further, the model of belief and disbelief has been used to illustrate the development of a set of uncertainty propagation functions for explanatory reasoning. The notions of explain away and joint explain are clearly reflected by these propagation functions. To prevent the computational complexity of these propagation functions from becoming exponential (one of the main merits of compositional reasoning), restrictions have been added to the concept of explanatory reasoning. The majority of these restrictions are not severe and must be satisfied by the application area. However, the most important restriction, the modelling of bidirectional influences, decreases the capabilities of explanatory reasoning with respect to modelling abduction.

In conclusion, with explanatory reasoning, a formalism has been developed that has attractive reasoning characteristics (explain away and joint explain) and propagation functions that have simple computational complexity. However, compositional reasoning fundamentally lacks the possibility of modelling bidirectional reasoning, and therefore it is concluded that supporting a general form of abductive reasoning is beyond the limits of the compositional reasoning architecture.

Using compositional reasoning in problem solving

Comparable to other reasoning formalisms, the compositional reasoning architecture has its strong and weak points. Advantages of the compositional approach are that domain models can be constructed easily, that inference is well-defined and that the computational complexity of uncertainty propagation is simple. Practical applications based on (deductive) compositional formalisms have proved the applicability of such an approach.

Limitations of compositional reasoning are imposed by the restriction on handling bidirectional influences and some demands on the problem area in which it is used. However, it is concluded that compositional reasoning remains an attractive formalism for modelling (parts of) an intelligent problem-solving system. This thesis clearly outlines both theoretical and practical constraints of this modelling potential.

Table of Contents

Chapter 4

Chapter 5

Notations and Abbreviations

In this thesis, the following notations and abbreviations are frequently used.

A	The set of Atomic propositions.
CF(a,e)	Certainty Factor of proposition a given environment e.
$C\mathcal{F}$(c)	Control Function, evaluation function for control steps c.
CS	The set of Control Steps.
d_a	The derivation of atomic proposition a.
D	The set of Derivations.
E	The set of Expressions.
$\{f_\neg,f_\wedge,f_\vee,f_\Rightarrow,f_\parallel\}$	The uncertainty propagation functions for deductive reasoning.
$\{f_\neg,f_{conv},f_{div},f_\Leftarrow,f_\parallel\}$	The uncertainty propagation functions for explanatory reasoning.
$I\mathcal{F}(e \Rightarrow a)$	Inference Function for the primitive inference steps $e \Rightarrow a$.
IS	The set of Inference Steps.
MB(a,d_a)	Measurement of increased Belief for proposition a, given d_a.
MD(a,d_a)	Measurement of increased Disbelief for proposition a, given d_a.
NAF	Negation As Failure inference strategy.
SLD	Selection function for Linear resolution of Definite clauses.
T_l	Basic type of the three-valued labelling $\{t,u,f\}$.
T_{ul}	Basic type of an uncertainty labelling.
UW(e)	Evaluation function for expression e resulting in label of type T_{ul}.
W(e)	Evaluation function for expression e resulting in label of type T_l.
$e \Rightarrow a$	Primitive inference step, expression e implies proposition a.
$\dagger,\perp$	A conflict in reasoning.

Chapter 1

Introduction

Artificial Intelligence (AI) has become a serious field of research over the past few decades. By no means will we try to provide a definition of AI here, since numerous (informal) definitions can be found in the literature. The majority of these definitions reflects the connection with human intelligence or behaviour. In combination with its mathematical nature, this positions AI between the technical and the cognitive sciences.

For some time, it has been the conviction of the AI research community that a General Problem Solver [Newell, 1972] could be developed that would be able to solve a great variety of problems. Feigenbaum was the first to acknowledge that this idea had to be abandoned and replaced by research, which still concentrated on general purpose reasoning, but applied to restricted problem areas [Bar, 1981]. Since then, AI research has been carried out in a great diversity of application fields, from theorem proving to speech recognition, and from medical diagnosis to robotics. The focus of this study is on the general field of expert systems or, as we prefer to call it, the field of knowledge-based systems.

1.1 Knowledge-based systems

In this section, we very briefly discuss the constructing components of a general knowledge-based system.

Three main components can be distinguished in a traditional knowledge-based system. The knowledge base, or *domain model*, contains a description of the knowledge that can be used to solve the problem at hand. The problem solver, or *inference model*, is a set of reasoning techniques that use the static knowledge from the knowledge base and dynamic knowledge from the situation at hand to actually find a solution. The *interface model* takes care of all interfacing with the environment of the knowledge-based system. This can be the user, an external data base or external computing facilities. The system components and their communication relationships are illustrated in Figure 1.1.

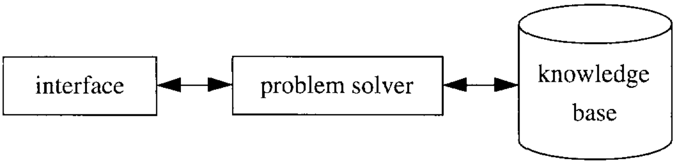

Figure 1.1 Knowledge-based system.

This structure of a knowledge-based system allows knowledge in the domain model to be made explicit and the communication with the environment to be defined independently. However, the significant part of the work of the system, the reasoning part, is still carried out by just one component of the system, the problem solver. A *reason maintenance model* was introduced into knowledge-based systems to lighten the burden of the problem solver and to increase its reasoning performance. The reason maintenance model dynamically builds its own knowledge base, that we will call the reason base, to support the inferencing by the problem solver. The reason base consists of previously made inferences and detected conflict situations, it can be viewed as a meta knowledge base. This reason base contains knowledge concerning the domain model in combination with the situation that it currently obtains (a problem instance or case). Besides run-time use, the reason maintenance model can also be used in analysing knowledge bases or implementing case-based reasoning. This is realised by inspecting the recorded reason bases of a variety of problem instances. The depiction of a knowledge-based system, extended by a reason maintenance model, is given in Figure 1.2.

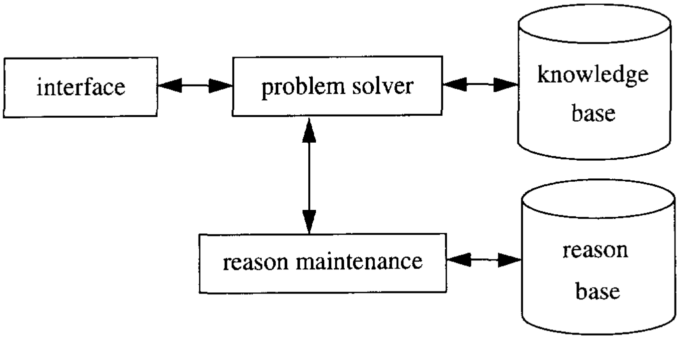

Figure 1.2 Knowledge-based system extended by reason maintenance.

In the remainder of this thesis, the subject of attention is the problem solver (inference model) and reason maintenance model of a knowledge-based system. The interface capabilities and knowledge representation techniques are given less attention.

1.2 Research objectives

In this section, we formulate our research objectives and discuss the motivations. For this purpose we first give descriptions of terms frequently used throughout the thesis.

When we consider the problem solver of a knowledge-based system we assume that it uses an inference model that combines small reasoning steps to solve the overall problem. This strategy is called *compositional* reasoning [Hájek, 1992]. We claim that the vast majority of existing knowledge-based systems are of this category.

Generally speaking, three major *reasoning types* are distinguished: deduction, induction and abduction. For a long time, *deductive reasoning* (from cause to consequent) was considered to be the most important strategy in both human and knowledge-based reasoning. However, the importance of *abductive reasoning* (explaining consequences by assuming causes) is increasingly acknowledged. Finally, *inductive reasoning* (generating relations between causes and consequences) is concerned with the learning aspect of reasoning.

A second notion that plays a key role in this thesis is *uncertainty*. It has been widely accepted that for the development of real-world AI applications, it is inevitable that knowledge-based systems are able to cope with the fact that knowledge and information is incomplete, imprecise, invalid and unreliable. These four conceptually different notions are often represented by just one term: uncertainty. In the closing chapter of the thesis, we comment on the way in which we have incorporated these notions into the developed theory.

When uncertainty is used in compositional reasoning, propagation functions are necessary to assign uncertainty measurements to newly derived information. This propagation process is called the *management* of uncertainty. In correspondence with the domain, inference and interface and reason maintenance model of a knowledge-based system, the term *uncertainty model* is often used for the representation and propagation of uncertainty. The main advantage of the compositional treatment of uncertainty in knowledge-based systems is its simple computational complexity.

A survey of the literature on compositional reasoning with uncertainty makes it clear that the existing models all use assumptions that restrict their use in general situations. In this study, we report on the analysis of these restrictions and the development of an uncertainty model that can do with fewer restrictions than the above. However, an even more important observation is the fact that all existing models are suited for deductive reasoning only. This brings us to the main issue of this study: Can a formalism for abductive reasoning under uncertainty be realised in a compositional setting?

If we wish to achieve acceptable results, we have to create a sound theoretical basis first [Kyburg, 1991]. Therefore, we develop a *general framework* for compositional reasoning with uncertainty, consisting of a domain, inference, reason maintenance and uncertainty model. The framework is general in the sense that no assumptions are made with respect to the type of reasoning, the uncertainty model and the application area that will be used. The appropriateness of this framework is demonstrated by showing that all the existing deductive uncertainty models are a special case of the framework. Based on this framework, an uncertainty model is developed that uses the structures from the reason maintenance model to eliminate the main restrictions of existing deductive approaches.

We develop an abductive strategy for compositional reasoning under uncertainty, that is called *explanatory reasoning*. We emphasise that the explanatory reasoning strategy is the result of incorporating an abductive reasoning strategy in a compositional environment. This does not mean that we defend the standpoint that abduction must be, or can best be, represented by a compositional system. It might well be that restrictions demanded by explanatory reasoning make it an awkward choice in a specific application. The concept of explanatory reasoning must therefore be compared to other abductive formalisms.

Further, uncertainty propagation functions are developed for explanatory reasoning. In this, special attention is paid to the computational complexity of these propagation functions. Recall that when simple computational complexity is lost, one of the main advantages of the compositional approach is lost, and other formalisms might be more appealing.

The development of theories in knowledge-based system research can best be demonstrated by implementing these theories in applicable environments. We have acknowledged this *implementation aspect* and here discuss implementing a part of the developed theory in an existing knowledge-engineering environment.

1.3 Benchmark problems

Some specific problems frequently appear in discussions on the management of uncertainty in compositional reasoning. We quote two of these problems from the literature that we think can be discarded as benchmark problems for these specific problems.

The first type of problem is the situation in which uncertain information which is based on a common cause for some fact is combined. The problem is that propagation functions in compositional reasoning systems do not take the history of some piece of information into account. We use the *news report example* from [Henrion, 1987, p. 111] as our benchmark example for this kind of problem.

> "The first radio news bulletin you hear on the accident at the Chernobyl nuclear power plant reports that the release of radioactive materials may have already killed several thousand people. Initially you place small credence in this, but as you start to hear similar reports from other radio and tv stations, and in the newspapers, you believe it more strongly. A couple of days later, you discover that the news reports were all based on the same wire-service report based on a single unconfirmed telephone interview."

The second type of problem is the situation in which the uncertainty of a piece of information has to be reconsidered when the uncertainty of some other piece of information with which it causes a common symptom is changed. The problem is that no propagation functions are available in most uncertainty models to cope with this kind of relationship between pieces of information. We use the *alarm call example* from Kim and Pearl [1983, p. 190] as our benchmark example for this kind of problem.

> "Mr. Holmes received a telephone call from his neighbor notifying him that she heard a burglar alarm sound from the direction of his home. As he was preparing to rush home, Mr. Holmes recalled that last time the alarm had been triggered by an earthquake. On his way driving home, he heard a radio newscast reporting an earthquake 200 miles away."

1.4 Scope of the thesis

After a brief overview of the literature (Chapter 2), both the theoretical aspects (Chapters 3, 4 and 5) and the practical aspects (Chapter 6) of the research objectives are covered and discussed (Chapter 7).

Chapter 2 contains an overview of important literature on AI. Knowledge representation and reasoning strategies are described in a nutshell, the types of reasoning and existing uncertainty models and reason maintenance systems are outlined here. In short, the overview describes the background setting of the research work. This chapter is concluded with the formulation of a set of design constraints for the theory developed in the following chapters.

In Chapter 3, a general framework for uncertain compositional reasoning is given. Starting from scratch, we carefully build up the foundation of our theory. In this chapter, a deductive reasoning strategy is added to the framework. Further, the framework is extended by the addition of a reason maintenance model and an embedding facility for uncertainty. The chapter is concluded by an evaluation of the framework, using both the design constraints given in Chapter 2 and the most well-known deductive uncertainty models for compositional reasoning.

Chapter 4 is dedicated to a specific uncertainty model for compositional deductive reasoning. The model is developed in isolation first, and embedded into the general framework (described in Chapter 3) afterwards. Further, the propagation of uncertainty is enhanced by exploiting information stored in the reason maintenance model. The enhancement yields a solution for the first benchmark problem.

Chapter 5 contains the description of an abductive reasoning strategy for the framework, called explanatory reasoning. The reasons for the chosen approach are discussed. The uncertainty model given in Chapter 4, is equipped with propagation functions for this type of reasoning. As a result of this, a solution for the second benchmark problem can be proposed. Finally, explanatory reasoning is analysed in comparison to other abductive approaches.

In Chapter 6, the knowledge engineering environment Delfi3 is described. We make clear that the representation language of this environment is a specialisation of the representation formalism of the general framework (recall Chapter 3). The implementation of an uncertainty model (as developed in Chapter 4) and a reason maintenance model within Delfi3, according to the theory from the preceding chapters, is the main topic here.

A discussion of the obtained results is given in Chapter 7. The research objectives, which have been described in Section 2, are the guidelines in this discussion. Further, we propose further investigations on some subjects that could not be covered within the scope of this thesis.

Chapter 2

Fundamentals of Knowledge Representation and Reasoning

A brief overview is given of the field of knowledge representation and reasoning in knowledge-based systems. Four major topics are distinguished: knowledge representation formalisms, inference models, uncertainty models and reason maintenance systems. We provide a global description of the fundamentals of these subjects as can be found in literature. The objective of the overview is to obtain some insight into the terminology, formalisms and techniques used in the field. The chapter is concluded with a discussion on the use of the described formalisms in the remainder of the thesis.

Each topic is considered to cover a specific aspect of knowledge representation and reasoning. In Section 1, knowledge representation formalisms are discussed. Within these formalisms the *domain model* of a knowledge-based system can be specified. The strategy for reasoning with this knowledge is specified in the *inference model.* In Section 2, the main characteristics of inference models are described, control of the reasoning process and coping with negation and incompleteness. Uncertainty of knowledge and information is another important aspect with respect to knowledge-based systems. An *uncertainty model* should support both representation and reasoning aspects, Section 3 contains an overview of current approaches. In Section 4, we discuss the different types of reasoning that can be distinguished. Section 5 is concerned with a learning aspect of knowledge-based systems. *Reason maintenance models* have been developed to administer the reasoning process and to use this information in future inferences. Finally, Section 6 contains some closing remarks on the presented aspects, and the specification of a set of design constraints.

2.1 Knowledge representation formalisms

Knowledge representation has a long history in AI. In the early days, production rules were the most frequently used representation structure. The strength of production rules is in the representation of heuristic knowledge [Davis, 1977; Clancey, 1983]. It was noted, however, that representation formalisms based on production rules have only limited expressive power, while in complex applications it is necessary to have more advanced formalisms to represent the domain model of the problem area. The idea of representing this kind of knowledge within objects and relations traces back to two fundamental representation formalisms: *semantic networks* and *frames.* The advantages of both formalisms have now been merged into a formalism called *structural inheritance networks.*

Semantic networks

The introduction of semantic networks is attributed to Quillian [1966]. In his paper he describes the basic components of semantic networks: nodes (to represent the entities in a world) and links (to represent the relationships between these entities). Later, inheritance was introduced into the basic concepts of semantic networks [Collins, 1970]. The main reasons for the success of semantic networks are the intuitive manner of knowledge representation and the ease of implementation. However, as pointed out by Woods [1975], there is fundamental criticism of the original definition of a semantic network. This criticism concerns the lack of coherent semantics for semantic networks, i.e. fundamentally different aspects of knowledge representation must be represented by the same construct (the link between nodes).

Frames

The idea of representing and structuring knowledge in frames is described by Minsky [1975]. Frames are supposed to describe a situation by clustering all information relevant to that situation in slots and fillers (the principle of locality). Further, frames that stand for similar concepts are able to inherit information from each other through the *is-a* link. A frame system contains a collection of frames and new concepts are identified by matching them against the frames already available in the system [Fikes, 1985]. Two main inference strategies are distinguished in frame systems: *inheritance* and *classification*. Inheritance collects slots and fillers from other frames to which there is an *is-a* link. Classification tries to insert a new frame in the right position in a given frame hierarchy. The main advantage of frames is their efficient representation of structural knowledge, the main weaknesses are the limited inference capabilities and the restricted possibilities in representing relational knowledge.

Structural inheritance networks

When trying to combine the advantages of frames and semantic networks, some important observations were made. Firstly, the need for a distinction between knowledge on a definition or generic level and on an individual level was argued by Levesque [1979]. A second observation concerned the distinction that should be made between declarative knowledge and procedural knowledge [Winograd, 1975]. Declarative knowledge is expressed in the domain model, whereas procedural knowledge is represented in the inference model, by means of expressions active in the domain model. Finally, a knowledge representation formalism should have a well-defined formal semantics [McDermott, 1978].

The term structural inheritance networks [Brachman, 1979] is used to describe the new category of representation formalisms. In short, the domain model of a structural inheritance network is a semantic network with frames as nodes and a number of (in a semantical sense) different links. A topic that has received much attention here is the technique to deal with multiple inheritance and the modelling of exceptions [Sandewall, 1986; Touretzky, 1986].

2.2 Inference models

As stated in the introduction, an inference model is used to specify and use procedural knowledge. Because of its strong theoretical foundations, logic was considered to be the most suitable candidate. The idea of using first-order predicate logic, or at least a subset thereof, like Horn clauses, as a programming language stems from Kowalski [1974] and is based on the principle of resolution [Robinson, 1965], an inference strategy for first-order logic well suited to be handled by computers. Two main problems to be taken into account concerning logic programming are control and negation [Lloyd, 1984].

Control and negation

The necessity to make a distinction between the logic and the control part of a logic program was pointed out by Kowalski [1979]. He stated that the programmer should only be concerned with the logic needed when solving a problem, while the control of the reasoning process should be handled by the reasoning process itself. However, the most well-known control strategy for the resolution of logic programs, SLD-resolution (Selection function for Linear resolution of Definite clauses), does not completely fulfil this demand. The programmer can influence the control of the program by (dis-) ordering the clauses or by using meta-control operators.

Various ways to cope with negation have been proposed within the framework of logic programming. The negation as failure (NAF) inference rule [Clark, 1978], however, is certainly the most well-known strategy. This inference rule is a weakened consequence of the closed world assumption which states that negative facts may be inferred from the absence of their positive counterparts [Reiter, 1978]. Thus, strictly speaking, the negation as failure rule is not a real negation in terms of classical logic.

The frame problem and non-monotonic logics

Although classical first-order logic is descriptively universal, it is not effective in dealing with problems as described by the*frame* problem [Hayes, 1969]. In short, the problem comes down to the question of which states of the world should change as a result of some action. Non-monotonic logics all emerged as an extension of a first-order logic, as a result of the various attempts to find a formal solution to this problem.

The main difference between a non-monotonic logic and first-order logic is that in the former assumptions are made about the state of unknown predicates. These assumptions, and all conclusions derived from them, must be withdrawn when they are proved invalid at some later time. Traditional deductive inference in first-order logic is not capable of doing this. As a starting point for a more detailed introduction one is referred to [Lukaszewicz, 1991].

The closed world assumption underlies database completion and predicate *circumscription*, as introduced by McCarthy [1980]. Predicate circumscription is based on finding the set of minimal models for some predicate given a theory base. Later on, McCarthy generalised this theory to formula circumscription [McCarthy, 1986]. The theory of *default logic*, as described in [Reiter, 1980], is based on an extension of first-order predicate logic with a set of inference rules, called default rules. A default theory consists of a set of premises (first-order formulas) and a set of default rules. A set of formulas that is consistent with the default theory is called an extension. In general, there is no description for the deduction process (or proof theory) for the extensions of a default theory, as there is for normal predicate logic. Instead, the extensions of a default theory are defined by a fixpoint construction. A model-theoretic semantics for both circumscription and default logic can be found in [Etherington, 1988].

Autoepistemic logic [Moore, 1985] was introduced as a propositional modal logic. The ideas were based on earlier work on non-monotonic logic by McDermott [1980; 1982]. In autoepistemic logic an explicit distinction is made between a proposition and the belief that the proposition holds. A stable belief set is defined to be a set in which all logical consequences of its beliefs are known. An autoepistemic extension is a stable belief set grounded in the set of premises. It is possible to translate the sentences of autoepistemic logic into the rules of a propositional default theory and vice versa, such that the autoepistemic extensions are equivalent to the extensions of the default theory [Konolige, 1988].

2.3 Uncertainty models

As argued in the introduction, the ability to cope with uncertain knowledge and data is crucial to the acceptance of knowledge-based systems. Over the last decades a number of approaches have been proposed. A distinction is often made in the literature between probabilistic and possibilistic approaches to uncertainty. A third class that is distinguished is the heuristic approach, taking the theory of endorsement [Cohen, 1985] as its main exponent. This theory however is more concerned with the uncertainty of the reasoning process itself, it differs fundamentally from the probabilistic and possibilistic approaches.

2.3.1 Probabilistic approaches

Since the early days of uncertainty in AI, it has been the conviction of many researchers that probability theory should be the foundation for the handling of uncertainty. In this view, uncertainty should be modelled through a conditional probability distribution function on a set of variables. The strength of the dependency of some hypothesis on some evidence should be determined by prior observation (objective interpretation) or by some degree of gamble based on the actual truth of the hypothesis (subjective interpretation) [Bonissone, 1991a].

Bayes' theorem, concerning the conditional probability of some hypothesis on some piece of evidence, was the starting point for the first uncertainty models. However, quite soon it became clear that there were some difficulties with the applicability of this theorem in knowledge-based systems. The theorem requires a disjunct set of all hypotheses and conditional independence of the pieces of evidence for each hypothesis. Furthermore, an enormous number of prior and conditional chances must be known beforehand. Finally, Bayes' theorem does not support any explicit propagation functions for the updating of uncertainty through the logical operators like conjunction and disjunction. The first uncertainty models therefore concentrated on finding solutions to these problems.

The subjective Bayesian methods

In the subjective Bayesian method [Duda, 1976], the problem of the availability of prior and conditional chances is solved by allowing them to be estimated by experts. In general, these estimations are inconsistent. In the subjective Bayesian method, a consistent combination function is introduced that is based on an interpolation between the expert estimations and the boundary conditions of probability theory. Further, propagation functions are introduced to handle the uncertainty of combined pieces of evidence with respect to the logical operators. The subjective Bayesian method, however, does not present a solution for one of the main problems of probabilistic reasoning: the demand for the independence of pieces of evidence used in the propagation functions.

Certainty factor model

The certainty factor model of Shortliffe [1975] and Buchanan [1984] was introduced for the handling of uncertainty in rule-based systems. The model is based on the idea that pieces of evidence can not only increase but also decrease the belief in some hypothesis. These measures of belief and disbelief are defined in terms of probability theory. Further, a set of propagation functions is specified to perform the updating of the belief and disbelief measures. The certainty factor is based on both measures and the propagation functions for certainty factors are derived from the propagation functions for the individual measures. The problem of prior and conditional chances is, like in the subjective Bayesian method, solved by expert estimations of the certainty factors. Because of its simplicity in use and its intuitively appealing semantics, it is by far the most often used model in practice. However, to establish this, the model has undergone some pragmatic changes that have weakened its theoretical foundations [van der Gaag, 1990].

Bayesian belief networks

Bayesian belief networks were introduced in [Pearl, 1982]. A Bayesian belief network consists of two components: a dependency graph and a propagation scheme.

The dependency graph of a Bayesian belief network is a directed acyclic graph, consisting of nodes and arcs, in which mutual dependencies between real-world entities are modelled. A probability matrix specifies the degree of dependency between these entities. The direction of the arcs in the graph is from cause to consequent. The dependency graph has to be acyclic because otherwise propagation cannot end.

The propagation scheme of a Bayesian belief network consists of updating functions and message passing. The propagation scheme is activated by the introduction of new evidence to one of the nodes of the network and aims to propagate this piece of evidence through the dependency graph until a new equilibrium is reached. Local updating concerns the updating of uncertainty in a node, due to incoming messages from its direct neighbours. Message passing is used to pass the uncertainty of a node to its neighbours and is done through λ-messages (in the direction of an arc) and π-messages (against the direction of an arc).

An efficient propagation scheme for belief in tree-typed networks was originally proposed in [Pearl, 1988]. However, as the complexity increases when going from tree-typed networks to general graphs, the computational complexity becomes exponential [Cooper, 1990]. To handle such complexities, techniques were introduced to decompose general graphs into less complex structures (clique trees), as proposed by Lauritzen [1988].

Dempster-Shafer theory

The Dempster-Shafer theory is based on a mathematical theory of uncertainty [Dempster, 1967] that was further developed in [Shafer, 1976]. The theory is considered to be a generalisation of probability theory and was not specifically developed for the handling of uncertainty in knowledge-based systems.

The theory is based on a basic probability assignment, assigning a measure of uncertainty to all subsets of the frame of discernment. This basic probability assignment is transformed into an uncertainty interval through the Mobius-transform. The lower bound of this interval is called the belief of a proposition, whereas the upper bound is referred to as the plausibility of the proposition. This Mobius-transform is the static part of the theory and is common in all methods based on the Dempster-Shafer theory.

The dynamic part of the Dempster-Shafer theory consists of a combination function for the updating of the bounds of the uncertainty interval. In the form described by Shafer, this function is of exponential complexity. Thus, there are two problems to solve when using the Dempster-Shafer theory in knowledge-based systems: defining a complete set of propagation functions and reducing the complexity of the calculations. Over the last decade, a number of approaches have been developed to tackle these problems [Shenoy, 1990; Voorbraak, 1989].

2.3.2 Possibilistic approaches

In the possibilistic approach, uncertainty is not based on the frequency or the randomness of the membership of an element of a well-defined set as it is in probabilistic approaches. Instead, uncertainty is based on the partial membership of an element of an imprecisely defined set. Zadeh is considered the first to develop this idea into a possibilistic approach which is based on his theory of fuzzy sets.

Fuzzy set theory

The development of fuzzy set theory [Zadeh, 1965] opened up the possibility for a totally different approach toward uncertainty. Fuzzy set theory is an extension of classical set theory in the sense that an element is allowed to be a partial member of some set. A membership function, or fuzzy set, is introduced to define the degree of satisfiability. All traditional set operations like union, intersection and Cartesian product are also defined in fuzzy set theory.

Fuzzy set theory has been used as the foundation for fuzzy logic [Zadeh, 1975]. Fuzzy sets are used to express linguistic notions, and fuzzy operators, as defined in fuzzy logic, are used to express the meaning of linguistic operations. On the choice of these fuzzy operators, however, there is much disagreement.

A second important use of fuzzy set theory is found in possibility theory [Zadeh, 1978; 1983]. In this paper, Zadeh made a distinction between meaning and measure of information. He argued that the possibilistic notion of the meaning of information is best described by a (fuzzy) membership function, for instance in modelling someone being *old* or *tall*. Contrary to this notion of possibility, the probabilistic notion of the measure of information is the result of modelling the frequency of occurrence of some characteristic. Clearly, two fundamentally different notions underlay the two approaches.

Two fuzzy functions are available for the propagation of uncertainty. The *generalised modus ponens* is used for the propagation of uncertainty over an implication, the *compositional rule of inference* is used for a fuzzy relation. Since the introduction of fuzzy set theory and its use in fuzzy logic and possibility theory, there has been enormous attention from both the theoretical and the practical side. Although there is still fundamental criticism of the theory, it has become a major research field.

Triangular norms and conorms

The theory of triangular norms and conorms (T-norms and S-norms) was not developed in the context of the management of uncertainty in knowledge-based systems, it is part of the general mathematical theory on metric spaces. Triangular norms and their dual conorms are the most general families of binary functions that satisfy the requirements of the conjunction and disjunction [Schweizer, 1983]. Both norms are monotonic, commutative and associative and their boundary conditions satisfy the two-valued truth tables of the *and* and *or* operators. The objective of the introduction of these norms was to establish a theoretical basis for a syntactic and semantic definition of a subset of uncertainty calculi. An uncertainty representation form is given by a term set of linguistic statements of likelihood [Bonissone, 1986].

Reasoning with uncertainty, as described in [Bonissone, 1987], describes the implementation of a possibilistic propagation technique. Uncertainty of facts is represented by an interval in which the lower bound represents the minimal degree of confirmation, the upper bound the maximal degree of disconfirmation and the width of the interval represents the amount of ignorance. Uncertainty of dependencies is represented by a degree of sufficiency, indicating the strength with which the antecedent implies the consequent, and a degree of necessity, indicating the strength to which a failed antecedent implies a negated consequent. The uncertainty intervals are propagated and aggregated by triangular norm- and conorm-based calculi.

2.4 Types of reasoning

In the literature, three fundamentally different types of reasoning are distinguished: deduction, induction and abduction.

Deduction is characterised by the derivation of a conclusion, which is based on the presence of the premise of some causal relation. An important characteristic of this kind of reasoning is that it is impossible to derive false conclusions from true premises. There is not much disagreement on the definition of deduction, the notions of the other types of reasoning, induction and abduction, are less well-defined; we will explain what we mean by these terms. The notion of deduction can be illustrated by the following scheme.[1]

Example 2.1 (deduction)

Relation:	$A \rightarrow B$
Premise:	A'
Conclusion:	B'

Induction is the process of extracting *knowledge* which is represented by relations between the entities in some domain, following from a number of observations of or experiments made on that domain. In [Thagard, 1992] this process is called *generalisation.* In fact, deduction and abduction often use relations that have been developed through induction. However, the nature of induction is rather different from both deduction and abduction (as we use it), and is therefore not covered in this thesis. The scheme of induction is given in the following example, in which A_i and B_i represent multiple observations concerning A and B.

Example 2.2 (induction)

Facts:	A_i'
Facts:	B_i'
Relation:	$A \rightarrow B$

Abduction is probably the less well-defined type of reasoning. Informal definitions concerning abduction are, among others, given in [Pople, 1973, 1977; Josephson, 1987] as *the generation of hypotheses which, if true, would explain some collection of observed facts.* This definition implicitly assumes that a relationship is available between explanations and observations. However, abduction is also used in the context of forming theories that explain observed phenomena. Using abduction this way is called *rule abduction* by Thagard [1992] or *abductive rule generation* by van der Lubbe and Backer [1993]. This form of abduction is based on abduction as introduced by Peirce [1958], where it was raised in the context of the development of knowledge theory. Schemes representing both types of abduction are given in Examples 2.3 (pure abduction) and 2.4 (theory abduction).

[1] The use of A' and B' in the reasoning schemes implies that some form of uncertainty is allowed concerning the presence of characteristics A and B.

Example 2.3 (pure abduction)

Fact: B’
Relation: $A \rightarrow B$

Explanation: A’

Example 2.4 (theory abduction)

Observation: B’

Relation: $A \rightarrow B$
Explanation: A’

In the remainder of this thesis, we use a definition of abduction in which is assumed that the relation between observations and explanations is known beforehand. This form of abduction is called *abductive rule inference* in [van der Lubbe, 1993]. The conviction that this form of abductive reasoning is used more frequently in the human reasoning process than was generally recognised, and allows more flexibility in reasoning, has gained support in recent AI research. We remark that in rule-based systems, abductive reasoning has often been modelled through a combination of deductive reasoning and a backward chaining control strategy. We do not use such an approach, since it ignores an important aspect of abductive reasoning, namely the possibility to model a dependency relation *between* explanations or parts of an explanation. These dependency relations are especially interesting when uncertainty is used in combination with abductive reasoning.

2.4.1 Formalisms enabling abductive reasoning

We discuss three formalisms from the literature that have incorporated some form of abductive reasoning.

Parsimonious covering theory

Peng and Reggia [1990] extended the description of abduction with the necessity of a method to discriminate between explanations, based on their plausibility. Their definition of abduction is given by *inferring the most plausible explanation(s) for a given set of observations.* They developed a framework for diagnostic problem solving, using a combination of both abductive and deductive reasoning. The abductive component is represented by the *parsimonious covering theory*, in which plausible sets of *explanations* (covers) are generated for the observed *manifestations.* These covers are ordered by using deductive reasoning steps (causations), that are labelled with a probabilistic uncertainty measurement. A branch and bound type search method is used to select the most plausible set of explanations from the available covers.

Bayesian belief networks

Bayesian belief networks naturally integrate the combination of deductive and abductive reasoning. The importance of a uniform approach was, among others, acknowledged by Pearl [1988], it recognised that people do not use two separate rules to express both causal and diagnostic reasoning steps, i.e. *fire implies smoke* and *smoke makes fire more credible* can be represented by the same dependency relation. As we saw in the previous section, Bayesian belief networks are also able to cope with uncertainty.

Abductive Horn clauses

In [Poole, 1991; 1993], a description is given of an extension of Horn clause logic programming towards abduction. This is done by introducing a construction, the *assumable*, in which a set of possible hypotheses is assumed, each associated with a prior probability. To guarantee the correctness of probabilistic propagation, some assumptions are formulated for this construction. The possible hypotheses must be independent (both logically and probabilistically), disjoint (or mutually exclusive) and covering (i.e. be sufficient to explain some observation). Traditional (deductive) Horn clauses are used to discriminate between possible hypotheses, a strategy that is also used by Peng and Reggia [1990].

2.5 Reason maintenance systems

A reason maintenance system[2] serves as a support system for the problem solver (inference model) of a knowledge-based system (recall Figure 1.2). A reason maintenance system consists of a dependency graph (the reason base) and a labelling algorithm (reason maintenance). The dependency graph consists of nodes, representing entities, and links, representing justifications. The labelling algorithm assigns labels, representing facts, to the nodes. The intended goal of a reason maintenance system is to maintain consistency between the labelled facts. Further, the system allows learning from previously made inferences within one problem instance. For a more detailed description of reason maintenance theory, the reader is referred to [McDermott, 1991] and [Reinfrank, 1989].

Two important characteristics of reason maintenance systems are that no contradictory facts may be held simultaneously and that all facts must have some well-defined support. An observation that can be made on reason maintenance systems is that they are meant to work independently from the knowledge representation formalism and inference model used. Although this is a pleasant characteristic, it introduces the problem of *encoding*: What information must be stored in a node and how should information be transferred between the inference model and the reason maintenance system?

Through the years, a number of reason maintenance systems have been developed. A distinction can be made between two fundamentally different views: justification-based and assumption-based reason maintenance.

Justification-based reason maintenance

Justification-based reason maintenance was first described by Doyle [1979] and it is the basis of many reason maintenance systems that have been developed since. Justification-based

[2] We use the term *reason* maintenance system consistently, instead of switching between reason and *truth* maintenance system [Doyle, 1979].

reason maintenance works with a labelling algorithm that labels the nodes either *in* or *out*, depending on the node being in or out of the current set of facts. The labelling algorithm provides two basic techniques: reason maintenance and dependency-directed backtracking. Reason maintenance is used when a node with label *out* is given a valid justification. Dependency-directed backtracking, the term was introduced in [Stallman, 1977], is used when a valid justification is found for a node representing a contradiction, a *nogood* node.

A correct strategy to deal with nogood nodes was described by Elkan [1989]. However, this strategy introduces a non-monotonic cycle in the dependency graph, which causes the complexity of the labelling algorithm to become exponential. The introduction of three-valued reason maintenance, with labels *in*, *out* and *unknown*, enables a strategy to construct partial labellings of dependency graphs with non-monotonic cycles in quadratic time [Witteveen, 1991; 1993].

Assumption-based reason maintenance

A different view of reason maintenance was developed by de Kleer [1986]. His *assumption-based truth maintenance system* maintains for each node the set of (minimal) environments in which it holds, justifications are used to construct these environments. The environment of a node consists of a number of assumption nodes, nodes representing (retractable) assumptions made by the problem solver. De Kleer further extended his reason maintenance system with respect to default reasoning, disjunctions of assumptions and non-monotonic justifications. He tackled the encoding problem with a consumer strategy. In this, most attention is paid to the control mechanism used for the communication between the inference model and the reason maintenance system. Advantages of his approach are that the labelling of a node can be done efficiently by inspecting its environments, that the consequences of retracting an assumption can easily be dealt with, and that it is possible to reason with multiple contexts. The construction of the environments, however, has been proved to be NP-hard [Provan, 1988].

Other views on reason maintenance

There are two other conflicting views on reason maintenance: an implementation-based versus a logic-based approach and a single-context versus a multiple-context approach. In logic-based reason maintenance, the emphasis is put on increasing the expressiveness of the formalism, whereas in implementation-based reason maintenance, the emphasis is on increasing the efficiency of the labelling algorithm. In the single-context approach only one consistent set of facts is available at a time, whereas in a multiple-context approach several consistent sets may be available at the same time [Martins, 1988]. Although the properties are in theory independent of each other, it should be mentioned that justification-based reason maintenance is strongly related to the logic-based and single-context approaches and that assumption-based reason maintenance is strongly related to the implementation-based and multiple-context approaches [Kelleher, 1988].

2.6 Closing remarks

Section 1 is concerned with *knowledge representation formalisms*. With the introduction of structural inheritance networks, a powerful formalism has come up. Striking problems on the subject, like exception modelling and multiple inheritance, has been sufficiently solved. No further attention is paid to research on this subject in the remainder of this work.

As argued in Section 2, first-order predicate logic is considered to be the best descriptive formalism for the *inference model* of a knowledge-based system. However, the lack of an efficient proof theory is known to be a severe objection. Research on this subject is still going on, but is not covered here. From a practical point of view, it is more convenient to use a subset of first-order logic, like Horn clauses, for the inference model.

An observation that can be made on *uncertainty models*, as described in Section 3, is that despite their fundamentally different approaches toward uncertainty, the probabilistic and possibilistic approaches should be regarded as being complementary rather than competitive [Bonissone, 1991a]. The consequence of this view is that no choice between the two approaches is necessary and integration of both approaches makes more sense. This view, the use of both probabilistic and possibilistic approaches as complementary formalisms in a multi-type uncertainty model, is also advocated in [van der Lubbe, 1990].

As outlined in Chapter 1, one of the major problems in probabilistic reasoning is the demand for the independence of different pieces of evidence. In most uncertainty models, this is solved by making strong assumptions concerning the mutual and conditional independency relations of the variables at hand. Most approaches create an environment that shields a variable from other variables to which no dependency is presumed. A reason maintenance system contains information concerning all dependencies used in the reasoning process. Extending the Boolean label of a node to some measurement of uncertainty as label for a node opens a possibility to use dependency information during the propagation of the uncertainty stored in the labels.

From Section 5 we can conclude that the importance of *reason maintenance* is twofold. Firstly, the information stored in a reason maintenance system can be used to optimise the inference model, see for instance [Dechter, 1989]. Secondly, the use of a reason maintenance system is considered to be critical to the construction of efficient implementations of non-monotonic formalisms. It is therefore obvious that a reason maintenance system should be part of a knowledge-based system.

2.6.1 Specification of design constraints

As outlined in Chapter 1, the global objective of this work is the study of existing models and the development of new models for the management of uncertainty in compositional reasoning systems. In Chapter 3, we report on the development of a *general framework for compositional uncertain reasoning* that can serve as a basic framework for the development of these new *uncertainty models*. In this section, we formulate a number of *design constraints* that can be used to evaluate this general framework, and specific uncertainty models based on it, and compare them with other approaches.

Specifying constraints is also done in [Bonissone, 1987], where the *reasoning with uncertainty module* (RUM) is evaluated using a set of *desiderata.* The reason for mentioning this specific work here is the fact that besides a number of constraints concerning the uncertainty model, some additional constraints are formulated that are related to the inference model. When developing our general framework, it is necessary to specify a set of constraints not only for the uncertainty model, but for the inference and even the reason maintenance model as well. In the following, we specify such a set of constraints, the set is called the *design constraints,* accompanied by the underlying arguments. Similarities with the desiderata of Bonissone (given in Appendix A) are explained as well.

In addition, we state that some enhancements can be made to uncertainty models by using the combined sources of information of inference, uncertainty and reason maintenance models. The possibilities of these *uncertainty model enhancements* must be analysed for each choice of inference and uncertainty model.

Design constraints on the inference model

Two design constraints are formulated for the inference model. The first constraint is concerned with the expressiveness of inference models, i.e. the types of reasoning that must be supported. The second constraint is concerned with the representation of inconsistencies in the inference model.

1. The inference model should be able to support a deductive as well as an abductive reasoning strategy.
   When we observe the reasoning types deduction, induction and abduction, we distinguish a fundamental difference between the types. Induction is concerned with the learning or construction of knowledge bases, where deduction and abduction are concerned with reasoning. Since we are primarily interested in a framework for reasoning, in the scope of this thesis we limit ourselves to abduction and deduction only. Inference models for both reasoning strategies have to be developed within the context of the framework. Clearly, there is no equivalent demand in the desiderata for this constraint, Bonissone assumes the use of a deductive reasoning strategy.
2. There should be an explicit representation of inconsistency in reasoning.
   Different aspects of uncertainty were outlined in Chapter 1, inconsistency in reasoning is one of them. The demand for explicit modelling of this characteristic provides a first separation of these aspects. A comparable demand is formulated by Bonissone (see demand 4). Explicitly defining inconsistency at the inference level eases the definition of inconsistency in specific uncertainty models.

Design constraints on the uncertainty model

The design constraints that are used for the uncertainty model are very general, since we do not want to disqualify uncertainty models when we construct a general framework.

3. The representation type of the uncertainty model may not restrict the classes of potential uncertainty models.
   The argument for this demand is trivial, it means that the general framework may not demand the uncertainty representation type to be, for instance, numerical or symbolic. The demand is a generalisation of Bonissone's first, in which is stated that the representation form should allow explicit representation of evidence supporting, as well as refuting, some hypothesis.

4. The propagation functions of the uncertainty model should not assume independency relations between pieces of evidence, or exclusiveness of hypotheses.
   Again, from the viewpoint of developing a set of design constraints, this is a trivial demand. However, uncertainty models that do make these assumptions must be accepted equally well by the general framework. Comparable demands are formulated by Bonissone (see demands 7 and 8).

Design constraints on the reason maintenance model

We formulate two general design constraints on the reason maintenance model that are often mentioned in the literature on reason maintenance but rarely used in the specifications of a knowledge-based system. Clearly, no constraints are formulated by Bonissone on the reason maintenance model.

5. The reason maintenance model structures must be independent of the characteristics of the field of application, as specified in the domain model.
   It is clear that reasoning processes in a specific field of application influence the *contents* of structures in the reason maintenance model. However, in order to develop a *general* framework for compositional reasoning, it is obvious that the representation formalism of the model is application independent.

6. The reason maintenance model must work independently of the inference model used by the problem solver.
   This demand prescribes that a well-defined solution must be offered by the framework for the encoding problem (see Section 5). The advantage of a clear distinction between inference and reason maintenance model and a unique communication protocol between them is found in the independent development and implementation of both models.

Model enhancements

The advantage of having a reason maintenance model in the general framework can be used to enhance the inference model and the handling of uncertainty. For instance, when we have defined a specific uncertainty model, we can tune the propagation functions to the situation they are used in, when information concerning this situation is available in the reason maintenance model. Bonissone specified these kinds of enhancements in the desiderata as well. However, we prefer to call them model enhancements, instead of constraints.

1. The inference model can be enhanced, the enhancement is based on information offered by the reason maintenance model.
   Examples of enhancements of this type can be the early detection of coming inconsistencies or the skipping of inferences that have previously been made. This is a generalisation of Bonissone's demands 11 and 12, stating that the reasoning process must be tractable, i.e. there should be an explicit representation of supporting evidence of the hypotheses.
2. The propagation functions of the uncertainty model can be enhanced, the enhancement is based on information offered by the reason maintenance model.
   Enhancements of this type are made by enabling the uncertainty propagation functions to use a better-specified representation of the environment in which the propagation takes place. This is a generalisation of Bonissone's demands 10 and 15, which express that the reasoning process must be capable of selecting the most appropriate uncertainty propagation rule.

In Chapters 4 through 6, we give some illustrations of these enhancements, as applied in a specific inference and uncertainty model.

Chapter 3

A General Framework for Compositional Uncertain Reasoning

In this chapter, we develop a general model for compositional reasoning, including reason maintenance and uncertainty management. The model will function as a general framework for the techniques and formalisms to be developed in the remaining part of this thesis.

To start with, we have to get some more insight into the notion of compositional reasoning. An informal definition of compositional reasoning is given by constructing complex reasoning processes by combining small, well-defined reasoning steps. A set of operators is used to construct the reasoning process out of these primitive reasoning steps. Mathematical logic is certainty the best known underlying theory that can be found for this principle. We choose a three-valued logic as the basis of our framework. The choice for the three-valued logic was supported by the possibility it offers for explicitly representing the lack of knowledge, thus enabling a method for representing the incompleteness aspect of uncertainty. Introducing uncertainty into this provides us with a strong fundamental basis and guarantees the correctness of uncertainty propagation in the boundary situations. In this chapter, a deductive reasoning strategy for the reasoning primitive is added to the framework.

In Section 1, the representation formalism is introduced for the domain model of the general framework, where Section 2 contains the description of the reasoning model. In Section 3, a deductive inference strategy is specified for the primitive reasoning step. The integration of uncertainty into the general framework is the main topic of Section 4. Finally, in Section 5 the resulting framework is discussed.

3.1 A domain model for compositional reasoning

In this section, the specification formalism for the domain model of the general framework is defined. We use a layered approach, consisting of three levels. The expression level is the bottom level, in which the basic structures for reasoning, like propositions and expressions are introduced. The middle level, the inference level, contains the primitive reasoning structure. The top level, the control level, contains structures to enable the construction of problem-solving strategies based on the inference level.

3.1.1 The expression level

In this section, the specification formalism for the expression level is defined. First, the definition of the set of atomic propositions is given.

Definition 3.1
An *atomic proposition* is a literal.
The set of atomic propositions is called A.

Expressions are defined using the set of atomic propositions and the logical operators for the negation ($\neg$), conjunction ($\wedge$), and disjunction ($\vee$). The expressions are defined in a standard recursive manner [Hájek, 1992].

Definition 3.2
An *expression* is defined by
(atomic) each atomic proposition $a \in A$,
(negation) $\neg e_1$ when e_1 is an expression,
(conjunction) $e_1 \wedge e_2$ when e_1 and e_2 are expressions,
(disjunction) $e_1 \vee e_2$ when e_1 and e_2 are expressions.
The set of expressions is called E.

As argued in the introduction, we use a three-valued labelling T_l of true (*t*), false (*f*) and unknown (*u*) for the set of atomic propositions.

Definition 3.3
A *three-valued labelling* L is defined as the assignment of a label from { *t,u,f* } to each element of A.
In functional notation, $L: A \rightarrow T_l$.

The combination of an atomic proposition and a label is used frequently throughout the remainder of this chapter, this combination is called a binding.

Definition 3.4
A *binding* is defined as a 2-tuple consisting of an element from A and label from T_l.
Notation: $<a,l>$ where $a \in A$ and $l \in T_l$.
A *set of bindings* is denoted by $\{<a_i,l_i>\}$, the empty set by { }.

Note that a set of bindings that contains bindings for all atomic propositions from A can be considered equal to a labelling.

According to the three-valued labelling, an evaluation function for the set of expressions is defined. For compound expressions (expressions containing a logical operator), the evaluation function is defined using Kleene's truth tables [Turner, 1985].

Definition 3.5
An *evaluation function* W is defined as the assignment of a label from { *t,u,f* } to each element $e \in E$, where
(atomic) $W(e) = L(a)$, the label of the atomic proposition $a \in A$,
(negation) $W(e) = W_\neg(e_1)$ according to the truth table of $\neg$,
(conjunction) $W(e) = W_\wedge(e_1,e_2)$ according to the truth table of $\wedge$,
(disjunction) $W(e) = W_\vee(e_1,e_2)$ according to the truth table of $\vee$.
In functional notation, $W: E \rightarrow T_l$.

Tables 3.1 through 3.3 contain the truth tables for the evaluation of compound expressions that are used in the functions $W_{\neg}$, $W_{\wedge}$ and $W_{\vee}$.

Table 3.1 Negation

$\neg$	
t	f
u	u
f	t

Table 3.2 Conjunction

$\wedge$	t	u	f
t	t	u	f
u	u	u	f
f	f	f	f

Table 3.3 Disjunction

$\vee$	t	u	f
t	t	t	t
u	t	u	u
f	t	u	f

3.1.2 The inference level

In compositional reasoning, basic inference steps are combined into complex reasoning tasks. From Chapter 2, it has become clear that the basic inference steps for both deduction and abduction use the same relation; to model this relation we provide Definition 3.6. Whether this relation is used either deductively or abductively is prescribed by the reasoning type.

> Definition 3.6
> An *inference step* is defined by $e \Rightarrow a$, where $e \in E$ and $a \in A$.
> The set of inference steps is called IS.

The inference level is the middle layer of the framework, it is concerned with both the problem solving by the control level and the labelling of the propositions at the expression level. As a result of this, two functions are needed, one on each level. For the specification of the influence of the inference level on the expression level, an additional evaluation function is needed to update the labelling of propositions by executing inference steps. Further, an inference function is needed to inform the control level whether the execution of an inference step was successful or not.

The additional evaluation function for updating the label of a proposition is represented by the combination function $W_{||}$, it is used to combine two labels of an atomic proposition. The truth table of function $W_{||}$ is given in Table 3.4.

Table 3.4 Combination

$\|\|$	t	u	f
t	t	t	†
u	t	u	f
f	†	f	f

In the function for the combination $W_{||}$, a special symbol (†) is used when the labels true (*t*) and false (*f*) are combined. This situation is called a *conflict*, the reasoning process should be designed to avoid this situation. The definition of a conflict is given here in terms of the three-valued labelling.

Definition 3.7
A *conflict* (†) is defined to be the situation in which the label true (*t*) is assigned to an atomic proposition labelled false (*f*) or vice versa.

The definition of the combination function can be given by using the notion of a conflict.

Definition 3.8
The *evaluation function for the combination* $W_{||}$: $T_l,T_l \rightarrow T_l$ is defined for each element pair $a_1,a_2 \in A$, where $W_{||}(a_1,a_2)$ is specified in Table 3.4.

With the definition of the combination function $W_{||}$, we have completed the set of evaluation functions for the framework. We call the combination of the three-valued labelling L and the set of functions ($W_{\neg}$, $W_{\wedge}$,$W_{\vee}$,$W_{||}$), a *three-valued model* for the set of atomic propositions. This three-valued model is used as the fundamental structure when generalising towards the definition of a general uncertainty model.

Definition 3.9
The *three-valued model* M for the set of atomic propositions is defined by a three-valued labelling L, together with the set of four functions ($W_{\neg}$,$W_{\wedge}$,$W_{\vee}$,$W_{||}$) on L,
where $W_{\neg}$: $T_l \rightarrow T_l$ and $W_{\wedge}$,$W_{\vee}$,$W_{||}$: $T_l,T_l \rightarrow T_l$.

Now, an inference function (called $I\mathcal{F}$) can be defined in which the evaluation of the primitive inference steps is described. This inference function specifies the reasoning strategy (either deductive or abductive) for the primitive inference step. The result of the inference function is an element from the set $\{\mathcal{T},\mathcal{F}\}$, where $\mathcal{T}$ should be interpreted as *true* (the inference step is evaluated with success) and $\mathcal{F}$ as *false* (the inference step has led to failure). To avoid confusion with the labelling of atomic propositions and the evaluation function for expressions, different symbols are used here. In the course of this chapter we define a deductive inference function, an abductive inference function is introduced in Chapter 5.

3.1.3 The control level

The structures of the control level are introduced in this subsection. We have chosen three general-purpose control steps. The *choice* step indicates that the inference model can choose between several alternative inference steps. The *concatenate* step indicates that the inference model should concatenate several steps. The *denial* step indicates that the inference model cannot continue when a given inference step has succeeded. Choice, concatenate and denial steps are used to model a reasoning strategy, i.e. how to combine primitive inference steps into reasoning tasks. Control steps are also defined in a recursive way.

Definition 3.10
A *control step* is defined by
(primitive) each inference step $s \in IS$,
(choice) choose$\{c_1,..,c_n\}$ when $c_1,..,c_n$ are control steps,
(concatenate) concat$\{c_1,..,c_n\}$ when $c_1,..,c_n$ are control steps,
(denial) deny(c_1) when c_1 is a control step.
The set of control steps is called CS.

A control function is needed to specify the evaluation result of control steps. As already described, a label from the set $\{\mathcal{T},\mathcal{F}\}$ is used to denote the outcome. Obviously, for primitive control steps, the control function is equal to the result of the inference function ($\mathcal{IF}$) for that inference step. The control functions for the *choice* and *concatenate* steps are modelled by the existential and universal operators, the *denial* step is modelled by a negation. An actual execution procedure for control steps is given by the inference model.

Definition 3.11
An *control function* $\mathcal{CF}$ is defined as the assignment of a label from $\{\mathcal{T},\mathcal{F}\}$ to each element $c \in CS$, where
(primitive) $\mathcal{CF}(c) = \mathcal{IF}(s)$,
(choice) $\mathcal{CF}(c) = \mathcal{T}$ when $\exists_i\ \mathcal{CF}(c_i) = \mathcal{T}$, otherwise $\mathcal{CF}(c) = \mathcal{F}$,
(concatenate) $\mathcal{CF}(c) = \mathcal{T}$ when $\forall_i\ \mathcal{CF}(c_i) = \mathcal{T}$, otherwise $\mathcal{CF}(c) = \mathcal{F}$,
(denial) $\mathcal{CF}(c) = \mathcal{T}$ when $\mathcal{CF}(c_1) = \mathcal{F}$, otherwise $\mathcal{CF}(c) = \mathcal{F}$.
In functional notation, $\mathcal{CF}$: $CS \rightarrow \{\mathcal{T},\mathcal{F}\}$.

Now, the structure of a program can be defined, which is the equivalent of what others call the knowledge base or domain model. A program consists of a set of constraints C_0, a subset of the set of all possible control steps CS, a query Q from CS and an initial labelling L_0 over the set of propositions A.

Definition 3.12
A *program* P is defined as a triple (C_0,Q,L_0) consisting of a subset $C_0 \subset CS$, a query $Q \in CS$ and an initial labelling L_0 over A.

Now we have completely defined the specification formalism of the domain model, a definition of the desired outcome remains. With respect to a program P, we are interested in labellings L_Q over A that are consistent with constraint set C_0 and that fulfil a posed query Q.

Definition 3.13
An *extension* X for a program P is defined as a labelling L_Q over A with $\mathcal{CF}(Q) = \mathcal{T}$.

3.2 An inference model description

In this section, we describe an inference model for the structure of a program, as defined in the previous section. The description is preceded by the incremental definition of the process state, the structure that contains all information needed by the inference model.

Definition 3.14
A *trail* $T = <T_1, \dots ,T_m>$ is a finite set of control steps ($T_i \in CS$) that have been made in the execution process so far.

Definition 3.15
A *course* $C = <C_n, \dots ,C_1>$ is a finite set of control steps ($C_i \in CS$) that still must be made in the execution process.

Definition 3.16
A *process state* PS is a description of an intermediate state during the execution of a program P, given by [T,C,L], where T is a trail, C is a course and L is a labelling.

A process state is used to describe any situation during the execution, according to the set of control steps that have been and still have to be made at that particular situation. Its semantics can be compared to that of a trace in [Mellish, 1987].

Definition 3.17
A *process state stack* PSS is a description of the state of the reasoning process based on all available alternative process states. It is denoted by
$[PS_k]: \dots :[PS_1] =$
$= [T_k,C_k,L_k]: \dots :[T_1,C_1,L_1] =$
$= [<T_k^1, \dots ,T_k^m>,<C_k^n, \dots ,C_k^1>,L_k]: \dots :[<T_1^1, \dots ,T_1^m>,<C_1^n, \dots ,C_1^1>,L_1].$

The process state stack is a structure that contains all information necessary to execute a program. Each stack element contains an alternative reasoning path, which is characterised by a trail, a course and a labelling. Now, a deterministic procedure can be given for the inference model according to the four different sorts of control steps. The effect of each situation is described by changing the general form of the process state stack, given by

$$[<T_k^1, \dots ,T_k^m>,<C_k^n, \dots ,C_k^1>,L_k]: \dots :[<T_1^1, \dots ,T_1^m>,<C_1^n, \dots ,C_1^1>,L_1].$$

In this notation, the current process state is PS_k, and the control step to be executed is thus C_k^n. This current control step is always removed after execution. Now consider the effect of the four different types of control steps.

A primitive inference step

The successful execution of a primitive inference step has the following influence on the process state: The used inference step is added to the trail T_k and the result of this inference step is added to labelling L_k. The type of reasoning used to execute the inference step is not important here, its result is used to update the current labelling. The resulting process state stack has the following form

$$[<T_k^1, \dots ,T_k^m,T_k^{m+1}>,<C_k^{n-1}, \dots ,C_k^1>,L_k+]: \dots :[T_1,C_1,L_1]$$

$T_k^{m+1} = C_k^n$ is the inference step that has been executed,

L_k+ is L_k extended with the result of the inference step.

The effect of a primitive control step leading to failure is that the current process state PS_k is removed from the process state stack. This means that the current reasoning path has led to failure and that the inference model has to continue with the most recently created alternative reasoning path. The resulting process state stack is

$$[T_{k-1},C_{k-1},L_{k-1}]: \dots :[T_1,C_1,L_1].$$

A choice step

The effect of a choice step $C_k^n = \text{choose}\{s_1,..,s_n\}$ is that alternative process states PS_{k+1} to PS_{k+n} are added to the process state stack. The current process state is therefore copied a number of times and extended with each alternative. The resulting process state stack is

$$[T_{k+n},C_{k+n},L_{k+n}]: \dots :[T_k,<C_k^{n-1}, \dots ,C_k^1>,L_k]: \dots :[T_1,C_1,L_1].$$

T_{k+1} till T_{k+n} are copies of T_k,

C_{k+1} till C_{k+n} are equal to s_1 till s_n respectively,

L_{k+1} till L_{k+n} are copies of L_k.

A concatenate step

The effect of a concatenate step $C_k^n = \text{concat}\{s_1,..,s_n\}$ is that new concurrent reasoning paths are added to the course of the current process state, while the rest of the process state stack remains unchanged

$$[T_k,<C_k^{n+i-1}, \dots ,C_k^n, \dots ,C_k^1>,L_k]: \dots :[T_1,C_1,L_1],$$

C_k^n till C_k^{n+i-1} are equal to s_1 till s_n respectively.

A denial step

The effect of a denial step $C_k^n = \text{deny}(s_1)$ is that the inference step s_1 is evaluated by the inference model in isolation, by using process state stack $[<>,<s_1>,L_k]$.

When no extensions can be found for this program, the denial step as a whole succeeds, and the same action is taken as with a successfully made primitive control step, except for the fact that the labelling L_k is unchanged

$$[<T_k^1, \dots ,T_k^m,T_k^{m+1}>,<C_k^{n-1}, \dots ,C_k^1>,L_k]: \dots :[T_1,C_1,L_1].$$

When one or more extensions can be found for this program, the denial step as a whole fails, and the same action is taken as with a failed primitive control step, yielding

$$[T_{k-1},C_{k-1},L_{k-1}]: \dots :[T_1,C_1,L_1].$$

Initial and final state

The initial process state stack of program P = (S_0,Q,L_0) is given by [T_0,Q,L_0], where T_0 is the empty trail <> by definition, Q is the query and L_0 is the initial labelling. The function and semantics of this initial trail T_0 are made clear in the following section.

Two types of final states can be distinguished for the program state stack. A final process state is characterised by an empty course <>, for instance [T_k,<>,L_k]. In this, L_k is an extension for Q, derived through the trail T_k. When the process state stack is empty [], this means that no (more) extensions for the given query can be found, and execution can be terminated.

3.3 Deductive inference

The general framework, as defined in the preceding sections, does not depend on a specific type of reasoning, since we have not given the reasoning strategy for the primitive inference step. In this section we introduce a deductive inference strategy. Further, a reason maintenance model is outlined for this deductive inference strategy, which enables the proof of correctness of the inference model (given in Section 2). Other advantages of using a reason maintenance model are explained in later sections and chapters.

3.3.1 The deductive inference function

The primitive inference step consists of a relation (⇒), with an expression (e) as its premise and a proposition (a) as its conclusion. The deductive strategy for this primitive inference step is straightforward: successful evaluation of the premise results in a new or additional label for the proposition in the conclusion, and adds it to the current labelling. This strategy is formalised by the definition of a deductive inference function.

The inference function for an inference step (e ⇒ a), using *deductive* inference is based on two criteria, in which both the new label for the conclusion (say $L_{\Rightarrow}$) and the propagation function for the combination ($W_{||}$) are used. The first criterion demands that the evaluation of the premise of the inference step results in true (W(e)=*t*), this is a trivial demand. The second criterion demands that the application of the inference step may not give rise to a conflict (†). This can be done by checking the outcome of the combination function $W_{||}$ on the existing label for the proposition, L(a), and the new label, which is given by $L_{\Rightarrow}$.

As explained, the result of the inference function $\mathcal{IF}$ is an element from the set {$\mathcal{T}$,$\mathcal{F}$}, where $\mathcal{T}$ should be interpreted as *true* and $\mathcal{F}$ as *false*.

Definition 3.18

A *deductive inference function* $\mathcal{IF}$ is defined as the assignment of a label from {$\mathcal{T}$,$\mathcal{F}$} to each element s ∈ IS, where

$\mathcal{IF}$(s) = $\mathcal{T}$ when W(e) = *t* and $W_{||}(L(a),L_{\Rightarrow}) \neq$ †, otherwise $\mathcal{IF}$(s) = $\mathcal{F}$.

In functional notation, $\mathcal{IF}$: IS → {$\mathcal{T}$,$\mathcal{F}$}.

3.3.2 Reason maintenance

According to the prototype of a knowledge-based system, as given in Chapter 1, a reason maintenance system can be introduced to support the inference model. We develop some basic reason maintenance theory here that will serve several purposes: in developing a proof for the inference model and in supporting relabelling and uncertainty propagation.

Justifications, contradictions and dependency networks are the basic structures in a reason maintenance model. We define these terms in the context of the framework, and the deductive inference strategy for the primitive inference step. The definitions given here are based on the approach found in [Witteveen, 1990].

In general, a justification reflects the dependency of a new label assignment for some proposition on a set of already labelled propositions. We use a justification to record a successfully made deductive inference step by the inference model. The propositions that have been used in the evaluation of the premise expression are recorded in the antecedents of the justification as a set of bindings $\{<a_i,l_i>\}$. The conclusion of the deductive inference step, which is a single binding <a,l>, is recorded in the consequent of the justification.

> Definition 3.19
> A *justification* is a structure of the form $\{<a_i,l_i>\} \rightarrow <a,l>$, where $a,a_i \in A$, $l,l_i \in T_l$.
> The set of bindings $\{<a_i,l_i>\}$ is called the antecedent (Ante) of the justification, binding <a,l> is called the consequent (Cons) of the justification.
> When the antecedent is equal to the empty binding set { }, the justification is called a *premise justification.*

A contradiction is used to store a set of conflicting labels (sometimes called a *nogood set* in the literature). We use a contradiction to record failed deductive inference steps made by the inference model. The propositions that were used in the evaluation of the premise expression are recorded in the antecedents of the contradiction as a set of bindings $\{<a_i,l_i>\}$. The $\perp$ symbol is used to denote the failure.

> Definition 3.20
> A *contradiction* is a structure of the form $\{<a_i,l_i>\} \rightarrow \perp$, where $a_i \in A$, $l_i \in T_l$.
> The set of bindings $\{<a_i,l_i>\}$ is called the antecedent (Ante) of the contradiction.

A dependency network can be constructed when the set of atomic propositions is combined with a set of justifications. Since a deductive inference strategy is used, this is called a deductive dependency network.

> Definition 3.21
> A *deductive dependency network* consists of the set of atomic propositions A and a set of justifications J.

From Chapter 2, it is concluded that the task of a reason maintenance model is to support the correctness of the belief set of a knowledge-based system. In terms of the general framework, the belief set is represented by the current labelling of the set of atomic propositions. Therefore, we have to introduce a definition for correctness or validness of a labelling.

Definition 3.22
A labelling L over the set of atomic propositions A *satisfies a justification* j when $\text{Ante}(j) \subset L$ implies $\text{Cons}(j) \in L$.

Definition 3.23
A labelling L over the set of atomic propositions A is *valid* for a deductive dependency network consisting of the set of atomic propositions A and a set of justifications J, when L satisfies each justification in J.

The task of a reason maintenance model is to support or maintain valid labellings of a deductive dependency network. A common problem with the construction of valid labellings for these kinds of networks is the presence of *cycles*. We give an example of a deductive dependency network, containing a cycle.

Example 3.1
Consider $A=\{a,b\}$ and $J=\{\{<a,t>\} \rightarrow <b,t>,\{<b,t>\} \rightarrow <a,t>\}$, then $L=\{<a,t>,<b,t>\}$ is a valid labelling. However, the supporting justifications for propositions *a* and *b* are circular and therefore violate our intuitive notion of a *proof*.

To overcome these problems a *grounded* labelling, that will not accept circular proofs is defined. In order to define a grounded labelling, the notion of a monotonic proof is defined first, see also [Witteveen, 1990].

Definition 3.24
Let A be the set of atomic propositions, J be a set of justifications and L a labelling. Now a *monotonic proof* mp(a) for proposition $a \in A$ is a finite enumeration $<a_1,l_1>, \ldots ,<a_n,l_n>$, where $a_i \in A$ and $l_i \in T_l$, with
(i) $\forall_i <a_i,l_i> \in L$,
(ii) $a_n = a$,
(iii) $\forall_i \exists_{j \in J}$ with $\text{Cons}(j) = <a_i,l_i>$ and $\text{Ante}(j) \subset \{<a_{i-1},l_{i-1}>\}$.

The demand we propose for a grounded labelling of a deductive dependency network is that each atomic proposition *a* with label *l* from $\{t,f\}$ is supported by a justification, having $<a,l>$ as its consequent, that is satisfied by that labelling.

Definition 3.25
A labelling L over the set of atomic propositions A is *grounded* with respect to a set of justifications J when each atomic proposition $a \in A$, labelled true (t) or false (f) in L, has a monotonic proof mp(a) in L given J.

It follows from this definition that only the subset of A, propositions that are labelled *true* or *false*, needs a monotonic proof. The remaining atomic propositions, labelled *unknown*, do not need such a proof, which is intuitively correct when the semantics of *unknown* are considered.

A dependency network with a grounded labelling can still contain cycles, but they are balanced [Goodwin, 1987]. This means that at least one atomic proposition in a cycle has at least one justification that does not belong to that cycle and can therefore be used in constructing a monotonic proof for that proposition.

Example 3.2
If the premise justification { } → <a,t> is added to justification set J from Example 3.1, the labelling L={<a,t>,<b,t>} is a grounded labelling over A, since the cycle $a \rightarrow b \rightarrow a$ is balanced at *a*.

We conclude this section with the proof of correctness of the inference model described in the previous section.

3.3.3 Correctness of the inference model

Correctness of the inference model given in Section 2 is proved by developing a mapping of the inference model onto a reason maintenance model and using a theorem developed by Lloyd [1984] on the *immediate consequence operator*. This operator updates a model M (equal to what we call a labelling L), given a set of justifications J, by applying all justifications from J that can be applied given M. This yields a new model M' that is equal to M, updated with the consequents of the used justifications. Repeated application of the Lloyd operator results in a monotonic update of the models. When a model has not changed after applying this Lloyd operator, the fixpoint of the operator is found.

The mapping of the inference model structures onto the reason maintenance structures is straightforward. A trail T, used in the inference model, can be mapped directly onto a set of justifications J in the reason maintenance model. This is done by taking all atomic propositions used in the evaluation of the premise of a successful implication and their current labels as the antecedents of a justification and the conclusion of that implication as the consequent of the justification. A failed implication can be mapped onto a contradiction in a similar way.

Now, in some program P = (S_0,Q,L_0) with initial process state $[T_0,Q,L_0]$, none of the atoms from labelling L_0 can have a monotonic proof, since T_0 (and therefore J) is empty at the start of the inference process. Therefore, J is filled with premise justifications of the form *{} → <a,l>* for each atomic proposition *a* in L_0, with an initial label *l* equal to *true* or *false*. This is a common strategy to define a set of premises in reason maintenance literature.

With the following theorem, we can prove the correctness of the extensions derived by the inference model given in Section 2.

Theorem 3.1
Given the inference state $[T_i,C_i,L_i]$.
The labelling L_i is a grounded labelling over A given a set of justifications J, where J is the mapping of T_i.

Proof

By induction.

(i) In the case of L_0, the theorem holds by definition, because of the introduced set of premise justifications.

(ii) Now, consider L_i given $[T_i,C_i,L_i]$ and suppose labelling L_{i-1} is a grounded labelling over A given (the mapping of) T_{i-1}. Obviously, only the successful application of an inference step can have changed L_{i-1} into L_i. In this case, L_i is derived from L_{i-1} by adding *the conclusion of this inference step*, some binding *<a,l>*. This inference step can only be successfully applied, however, when *a* has a monotonic proof *mp(a)* in L_{i-1}, and therefore L_i is a grounded labelling over A given (the mapping of) T_i.

The enumeration of labelings $L_0,...,L_k$ can be compared with the fixpoint characterisation $M_0,...,M_n$ of the *immediate consequence operator* as defined in [Lloyd, 1984]. Clearly, the labelling L_0 is equal to the starting model M_0 of the Lloyd operator and the final labelling L_k is equal to the fixpoint, say M_n, of the Lloyd operator. Since in our model only one justification is used to come from one labelling to another, it is obvious that $n \leq k$. In the following example we demonstrate this, and also the fact that not all intermediate labellings generated by the fixpoint construction have equivalent labellings constructed by the inference model.

Example 3.3

Let A = {a,b,c,d,e,g} and L_0 = M_0 = {<a,t>,<b,f>},
resulting in two premise justifications
{ } → <a,t> and { } → <b,f>.
Now, suppose the following inference steps are generated
a ⇒ <c,t>, c ⇒ <e,f>, ¬b ⇒ <d,t> and a∧d ⇒ <g,t>,
resulting in the following justifications
{<a,t>} → <c,t>, {<c,t>} → <e,f>, {<b,f>} → <d,t> and {<a,t>,<d,t>} → <g,t>.
Then the sequence of labellings found by the inference model is
L_0 = {<a,t>,<b,f>},
L_1 = {<a,t>,<b,f>,<c,t>},
L_2 = {<a,t>,<b,f>,<c,t>,<e,f>},
L_3 = {<a,t>,<b,f>,<c,t>,<d,t>,<e,f>},
L_4 = {<a,t>,<b,f>,<c,t>,<d,t>,<e,f>,<g,t>}.
The immediate consequence operator gives
M_0 = {<a,t>,<b,f>},
M_1 = {<a,t>,<b,f>,<c,t>,<d,t>},
M_2 = {<a,t>,<b,f>,<c,t>,<d,t>,<e,f>,<g,t>}.
Now, L_0 = M_0 by definition. M_2 is the fixpoint of the immediate consequence operator and thus equal to the final labelling L_4. However, M_1 does not have an equivalent labelling in the enumeration due to the order in which the inference steps are executed by the inference model.

3.4 Uncertainty management

The labelling of atomic propositions used so far has been limited to the three labels *true*, *false* and *unknown*. In this section, we extend this labelling to some measurement of uncertainty in such a way that only minor changes are necessary to the theory developed for the rest of the framework. We define only a general model for reasoning with uncertainty, this means that no choice for a specific uncertainty model is made. Not even the type of the model, probabilistic or possibilistic, limits our framework.

Therefore, as the uncertainty measure, an unspecified type T_{ul} is introduced that is used to replace the definition of the three-valued labelling L by the definition of an uncertainty labelling UL. The specification of this uncertainty type T_{ul} is part of the definition of any specific uncertainty model.

Definition 3.26
An *uncertainty labelling* UL is defined as the assignment of a label from type T_{ul} to each element of A.
In functional notation, UL: $A \rightarrow T_{ul}$.

As a result of this, the definition of a binding, as given in Definition 3.4, can be extended in a straightforward manner to an uncertainty binding.

Definition 3.27
An *uncertainty binding* is defined as a 2-tuple consisting of an element from A and an uncertainty label from T_{ul}.
Notation: <a,ul> where $a \in A$ and $ul \in T_{ul}$.
A *set of uncertainty bindings* is denoted by $\{<a_i,ul_i>\}$, the empty set by { }.

As a result of the definition of an uncertainty labelling, the evaluation function for expressions W, see Definition 3.5, should be changed accordingly. The functions $(W_{\neg},W_{\wedge},W_{\vee})$ are replaced by a set of uncertainty propagation functions $(f_{\neg},f_{\wedge},f_{\vee})$ that again have to be specified in the definition of an uncertainty model. Then, the uncertainty evaluation function can be defined using these uncertainty propagation functions.

Definition 3.28
An *uncertainty evaluation function* UW is defined as the assignment of a label from T_{ul} to each element $e \in E$, where
(atomic) $UW(e) = UL(a)$,
(negation) $UW(e) = f_{\neg}(UW(e_1))$,
(conjunction) $UW(e) = f_{\wedge}(UW(e_1),UW(e_2))$,
(disjunction) $UW(e) = f_{\vee}(UW(e_1),UW(e_2))$.
In functional notation, UW: $E \rightarrow T_{ul}$.

Similar to the three-valued situation, an additional uncertainty propagation function, $f_{||}$, is necessary to support the updating of uncertainty labels, in case the successful application of an inference step derives a new uncertainty label for a proposition.

When working with uncertainty, it is of interest to assign a measurement of uncertainty to the used inference step as well. When doing this, an additional uncertainty propagation function, $f_{\Rightarrow}$, is necessary to support the impact of this uncertainty label[1]. This uncertainty propagation function has two arguments: one for the strength of the relation and one for the uncertainty label of the premise given the situation it is used in.

Comparable to the definition of the three-valued model, the uncertainty model is defined consisting of an uncertainty labelling UL and a set of uncertainty propagation functions.

> Definition 3.29
> An *uncertainty model* UM for the set of atomic propositions is defined by an uncertainty labelling UL, together with a set of five uncertainty propagation functions $(f_{\neg},f_{\wedge},f_{\vee},f_{\Rightarrow},f_{\|})$ on UL, where $f_{\neg}$: T_{ul}: $\rightarrow T_{ul}$ and $f_{\wedge},f_{\vee},f_{\Rightarrow},f_{\|}$: $T_{ul},T_{ul} \rightarrow T_{ul}$.

This definition of the uncertainty model, consisting of a representation and a propagation part, exactly covers the intuitive description of such a model. The representation aspect is specified by uncertainty labelling UL of type T_{ul} over A. The propagation aspect is covered by a set of five uncertainty propagation functions $(f_{\neg},f_{\wedge},f_{\vee},f_{\Rightarrow},f_{\|})$ for the update of uncertainty in the basic reasoning steps. The fill-in of both the uncertainty type T_{ul} and the set of propagation functions determines the character of an uncertainty model.

3.4.1 Embedding the three-valued model

We have extended the three-valued labelling of the set atomic propositions to an uncertainty labelling. However, it would be attractive if the uncertainty model could be considered as a generalisation of the three-valued model. To achieve this, we have to define a transition function from the three-valued model into the uncertainty model, such a transition function is called an embedding.

> Definition 3.30
> An *embedding* B of the three-valued model M in an uncertainty model UM is defined as a transformation function from the elements of the three-valued labelling L from M into elements of the uncertainty labelling UL from UM.
> In functional notation, B: $T_l \rightarrow T_{ul}$.

When we wish to maintain the properties that have been set out in the preceding sections, we must guarantee that the embedding satisfies the *boundary constraints* of a three-valued model. The following definition describes the impact of these boundary conditions on an embedding, if they are satisfied the embedding is called a *correct* embedding.

[1] Using a function $W_{\Rightarrow}$ in the three-valued model (see Definition 3.9) for the same purpose can be done straightforwardly. However, this function is meaningless in the three-valued case, since the impact of the premise can either be *true* (assign a new label to the conclusion) or *unknown/false* (the inference step is ignored).

Definition 3.31

An embedding B of the three-valued model M in an uncertainty model UM is *correct* when the following conditions hold for all $l_1,l_2 \in L$:

(i) $f_{\neg}(B(l_1)) = B(W_{\neg}(l_1))$,

(ii) $f_{\wedge}(B(l_1),B(l_2)) = B(W_{\wedge}(l_1,l_2))$,

(iii) $f_{\vee}(B(l_1),B(l_2)) = B(W_{\vee}(l_1,l_2))$,

(iv) $f_{||}(B(l_1),B(l_2)) = B(W_{||}(l_1,l_2))$.

In the following example, the uncertainty type T_{ul} and a correct embedding are given for the Dempster-Shafer theory.[2]

Example 3.4

The *uncertainty type* T_{ul} of the Dempster-Shafer theory is an interval [Bel(x), Pl(x)] consisting of the belief (Bel) and plausibility (Pl) measurements of this theory. Both measurements are defined on the interval [0..1].

A *correct embedding* B of the three-valued model into the Dempster-Shafer theory can be defined through $B(\mathit{t})$=[1,1], $B(\mathit{u})$=[0,1] and $B(\mathit{f})$=[0,0].

When the embedding of the three-valued model in a specific uncertainty model has been specified, only the definition of an inference function is needed to complete the framework for compositional reasoning with uncertainty. For deductive models, this definition can be derived straight from the inference function $\mathcal{IF}$, given in Definition 3.18. In this definition, the notion of a conflict in reasoning was used to distinguish between valid and invalid inference steps. Because of this, the definition of a conflict is also necessary in the context of an uncertainty model. To conclude, the definition of an inference function is specific to the type of reasoning as well as to the uncertainty model it is used in.

In Example 3.5, an embedding is given of the certainty factor model of Shortliffe-Buchanan[3], followed by the definition of an inference function for this model. The deductive strategy for the primitive inference step (given by $e \Rightarrow a$, with strength $CF_{\Rightarrow}$) uses two model specific characteristics. First, the premise of the inference step must be evaluated and, secondly, certainty factors +1 and -1 cannot be combined (the definition of the uncertainty conflict). In evaluating the certainty factor of the premise, the threshold 0.2, which is also used in most known implementations of the model, is used.

Example 3.5

The *uncertainty type* T_{ul} of the certainty factor model is equal to a real number (CF) from the interval [-1..+1].

The *propagation functions* $(f_{\neg},f_{\wedge},f_{\vee},f_{\Rightarrow},f_{||})$ of the certainty factor model are:

$f_{\neg}(x)= -x$, $f_{\wedge}(x,y)=\min\{x,y\}$, $f_{\vee}(x,y)=\max\{x,y\}$, $f_{\Rightarrow}(x,y)=x*y$,

[2] For a detailed description of this example, we refer to Appendix B.

[3] In Chapter 4, we analyse this model in more detail.

$$f_{||}(x,y)= \begin{array}{ll} x+y - x*y & \text{when } x,y > 0, \\ x*y \,/\, (1-\min\{|x|,|y|\}) & \text{when } -1 < x*y \leq 0, \\ x+y + x*y & \text{when } x,y < 0, \\ \dagger & \text{when } x*y= -1. \end{array}$$

A *correct embedding* B of the three-valued model in the certainty factor model can be defined through $B(t)=+1$, $B(u)=0$ and $B(f)=-1$.

An *inference function* $I\mathcal{F}$ for the certainty factor model is defined by
$I\mathcal{F}(s) = \mathcal{T}$ when $CF(e) \geq 0.2$ and $f_{||}(CF(a),f_{\Rightarrow}(CF(e),CF_{\Rightarrow})) \neq \dagger$, otherwise $I\mathcal{F}(s) = \mathcal{F}$.

The extension from a three-valued labelling to an uncertainty labelling as described here does not influence the reason maintenance model and the theory developed in Section 3. Of course, the forms of justification, contradiction and deductive dependency network are changed according to the introduction of the uncertainty labelling, but this has no influence on the inference model or the described reason maintenance techniques.

The introduction of uncertainty can be incorporated in the definitions of a program and an extension, as can be seen from the following definitions. This concludes the integration of uncertainty into the general framework.

Definition 3.32
An *uncertainty program* UP is defined as a triple (S_0,Q,UL_0) consisting of a subset $S_0 \subset CS$, a query $Q \in CS$ and an initial uncertainty labelling UL_0 over A.

Definition 3.33
An *uncertainty extension* UX for an uncertainty program UP is defined as an uncertainty labelling UL_Q over A with $C\mathcal{F}(Q) = \mathcal{T}$.

3.4.2 The problem of relabelling

Changing from a three-valued labelling to an uncertainty labelling introduces the problem of relabelling. This problem is characterised by an update in the uncertainty label of a proposition due to the combination function. The question is, should uncertainty labels of other propositions that are based on this proposition, also be updated? In the three-valued case, this situation could not occur, since the only change in the labelling of a proposition is from *unknown* to *true* or *false*, and no labelling of other propositions could have been based on the *unknown* label of this proposition. When using an uncertainty labelling, changes can be made to the uncertainty label of a proposition that has already been used in the derivation of other propositions.

When using the deductive strategy, the relabelling process is modelled straightforwardly when one realises that dependencies between propositions are stored in the justifications of the reason maintenance model. The relabelling process will act similarly to the proof of Theorem 3.1, each time all justifications that use the relabelled proposition in their antecedents are checked for their consequent to be relabelled. As a result of this, the relabelled proposition from the consequent can activate other relabelling processes.

In Appendix C, a short description is given of the relabelling algorithm, which consists of two steps: a marking step and a relabelling step. In the marking step, all propositions that have to be relabelled are marked and placed in an order in which they can be relabelled consecutively. In the relabelling step, the propositions are actually relabelled. The computational complexity of this relabelling process is of $O(|A||J|)$, in which $|A|$ is the number of atomic propositions and $|J|$ is the number of justifications in the dependency network. Here, we illustrate the relabelling on a simple dependency network.

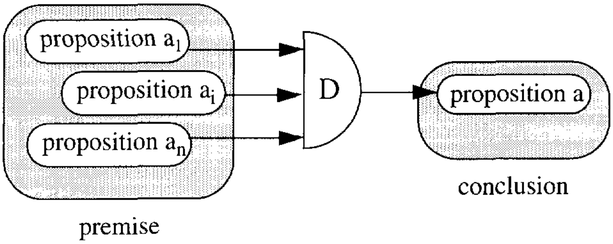

Figure 3.1 Symbol for deductive inference.

First, we define a symbol for an executed deductive inference step (justification). This symbol was introduced in [Goodwin, 1987] and reflects the relation between the input (the premise) and the output (the conclusion) of a deductive inference step. Capital D denotes that the deductive inference strategy is used on the primitive inference steps. In Figure 3.1, the general form of this symbol is depicted.

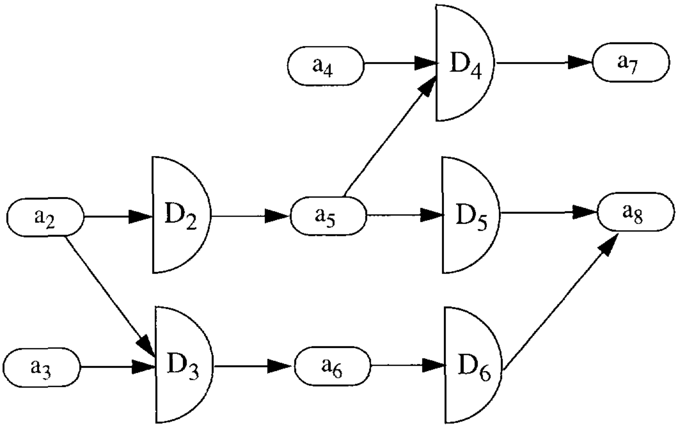

Figure 3.2 An example dependency network.

In Figure 3.2, an example dependency network is given. Depicted inference steps are $a_2 \Rightarrow a_5$ (D_2), $a_2 \vee a_3 \Rightarrow a_6$ (D_3), $a_4 \wedge a_5 \Rightarrow a_7$ (D_4), $a_5 \Rightarrow a_8$ (D_5) and $a_6 \Rightarrow a_8$ (D_6). The impact of executing inference step $a_1 \Rightarrow a_5$ and adding it to the network in the form of D_1 is demonstrated.

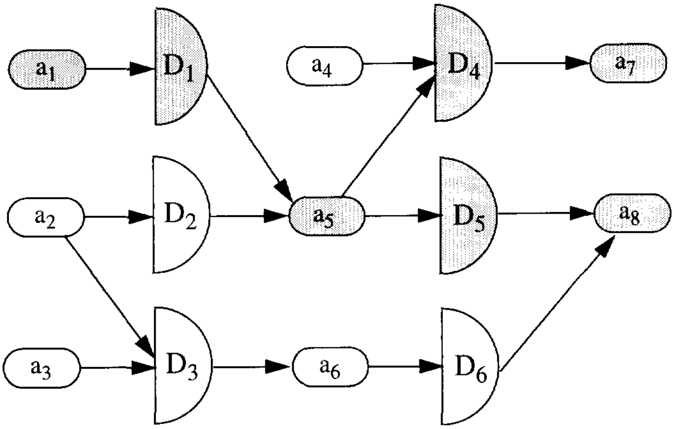

Figure 3.3 Relabelling in a dependency network.

A relabelling process has to be started at the moment that inference step $a_1 \Rightarrow a_5$ is executed (resulting in D_1) and added to the dependency network. Obviously, proposition a_5 has to be relabelled. Further, the consequences of relabelling a_5 are the relabelling of propositions a_7 and a_8. In Figure 3.3, the propositions and inference steps that are visited by the relabelling process resulting from the inference step $a_1 \Rightarrow a_5$ are marked.

3.5 Discussing the framework

In this chapter, we have defined a general framework for compositional uncertain reasoning, supported by a deductive reasoning strategy and a reason maintenance model. The framework was designed to create a general, coherent basis for the management of uncertainty in compositional reasoning. In this section, we conclude with an analysis of the developed framework.

First, a brief analysis is given of the inference and reason maintenance models provided by the framework. Next, a number of well-known uncertainty models, as described in Chapter 2, are examined in relation to the framework. This gives us some idea of the classes of uncertainty models that can be defined upon the framework. Finally, we describe the characteristics of the potential classes of uncertainty models that can be developed, starting from the framework. The design constraints, formulated at the end of Chapter 2, are used in the evaluation of the general framework.

3.5.1 The inference model

According to the specification formalism, as described in Section 1, we can see that in fact two levels of inference can be distinguished. There is inference on the level of expressions and inference on the level of inference steps. The inference of expressions is described by the evaluation function W (see Definition 3.5), and is based directly on Kleene's truth tables. This type of inference can better be denoted by the term *expression evaluation*, in the remainder, the term inference is used for the inference function of the set of control steps only.

The first design constraint (*1. The inference model should be able to support deductive as well as abductive reasoning*) is only half satisfied. A deductive reasoning strategy can easily be used with the framework, as outlined in this chapter. A discussion concerning abductive reasoning has been postponed until Chapter 5, where an abductive inference strategy is developed for the primitive inference step. The definition of a conflict in reasoning (see Definition 3.7) fulfils the second design constraint (*2. There should be an explicit representation of an inconsistency in reasoning*). As expected when formulating the constraint, the explicit definition translates into the notion of an uncertainty conflict (see Example 3.5).

We briefly discuss the position of the inference model in two well-known reasoning formalisms for compositional systems, production rules and logic programming.

Production rule systems

It is obvious that production rule systems can be modelled through the general framework. A production rule is modelled by the *deductive* strategy for a primitive inference step. The order in which the production rules are evaluated, i.e. using backward or forward reasoning strategies, can easily be modelled by using the *choice* and *concatenate* control steps. As expected, production rule systems are a special case of the framework.

Logic programming

Consider the definition of a logic program given in [Lloyd, 1984], as a set of clauses of the form $C \leftarrow B_1,...,B_n$, where C is some atomic formula in a first-order language and $B_1,...,B_n$ is a conjunction of, possibly negated, atomic formulas. Such a program clause is comparable to the inference step $B_1 \wedge ... \wedge B_n \Rightarrow C$. We can construct a mapping from the set of control steps CS (Definition 3.10) into the set of clauses of a logic program in which the primitive control step is considered as the atomic formula. Then, the *concatenate* step models the conjunction of atomic formulas and the *denial* step can be used to model negated formulas. The *choice* step can be used to model a set of formulas sharing the same head clause. The set of control steps constructed by the *choice*, *concatenate* and *denial* control steps is sufficient to model a set of clauses as used in the definition of a logic program. The semantics for the set of control steps CS, given by the control function $C\mathcal{F}$(see Definition 3.11) and the introduction of a grounded labelling (see Definition 3.25), are equal to the grounded model semantics of a logic program. It should be noted that SLD-resolution and the negation as failure (NAF) strategy (see Chapter 2, the subsection on *Control and negation* for a description of these strategies) are used to control the inference model.

3.5.2 The reason maintenance model

We can be brief in our analysis of the reason maintenance model used. Both design constraints are satisfied by the reason maintenance model, as defined in Section 3. No application-specific information is used in justifications and contradictions (*5. The reason maintenance model structures must be independent of the characteristics of the field of application, as specified in the domain model*). Since justifications only record the primitive inference steps, the reason maintenance model has no influence on the reasoning strategy (*6. The reason maintenance model must work independently of the reasoning strategy used by the inference model*). The model enhancements for the reason maintenance model discussed in Chapter 2 are further examined in Chapters 4 and 6.

3.5.3 Existing uncertainty models

In Chapter 2, a brief overview is given of the most used uncertainty models. Here, we discuss those uncertainty models that support a deductive reasoning type, and are therefore considered special cases of the general framework as developed in this chapter.

The Bayesian and subjective Bayesian methods

The (subjective) Bayesian method [Duda, 1976], is based directly on probability theory, especially the use of Bayes' rule. In the following example, we describe the type T_{ul} and the set of propagation functions of the (subjective) Bayesian model, followed by a correct embedding of the three-valued model.

> Example 3.6
>
> The *uncertainty type* T_{ul} of the (subjective) Bayesian model is equal to a probability number from the interval [0..1].
>
> The *propagation functions* $(f_{\neg},f_{\wedge},f_{\vee},f_{\Rightarrow},f_{||})$ of the (subjective) Bayesian model are:
> $f_{\neg}(x)=1-x$,
> $f_{\wedge}(x,y)=\min\{x,y\}$,
> $f_{\vee}(x,y)=\max\{x,y\}$,
> $f_{||}(x,y)=x*y \,/\, (x*y+(1-x)*(1-y))$,
> $f_{\Rightarrow}(x,y)$ is defined by (subjective) Bayes' rule propagation.
>
> A *correct embedding* B of the three-valued model into the (subjective) Bayesian model can be defined through B(*t*)=1, B(*u*)=0.5 and B(*f*)=0.

The choice for using B(*u*)=0.5 as the embedding of the *unknown* label is caused by the choice of the propagation function for the negation $f_{\neg}(x)=1-x$. Although this choice yields a correct embedding, from a semantic viewpoint one might ask whether a probability of 0.5 can be considered to model the label *unknown*. Another approach that can be used here is to consider all probabilities between 0 (*false*) and 1 (*true*) as *unknown*, and allow the specification an embedding term like B(*u*) $\in$ (0..1), i.e. any number between 0 and 1 is considered to represent label *unknown*.

Certainty factor model

The uncertainty model developed by Shortliffe [1975] and Buchanan [1984], the certainty factor model, can be shown to be a special case of the general framework. In Example 3.5, the uncertainty type T_{ul}, the set of propagation functions and a correct embedding of the three-valued model were already specified. Further, an inference function was given for the primitive inference step that completes the specialisation of the general framework into the certainty factor model.

Dempster-Shafer theory

The uncertainty model developed by Dempster [1967] and Shafer [1976] can be shown to be a special case of the general framework. In Example 3.4, the uncertainty type T_{ul} and a correct embedding of the three-valued model were specified. In Appendix B, the set of propagation functions and an inference function can be found.

Triangular norms and conorms

From Chapter 2, we know that families of T-norms and S-norms are the most general families of functions that satisfy a set of conditions for the conjunction and disjunction operators [Schweizer, 1983]. We use definitions here given in [Bonissone, 1986].

Definition 3.34
A *triangular norm* T is a function that satisfies
(boundary) $T(0,0) = 0$,
(boundary) $T(a,1) = T(1,a) = a$,
(commutativity) $T(a,b) = T(b,a)$,
(associativity) $T(a,T(b,c)) = T(T(a,b),c)$,
(monotonicity) $T(a,b) \leq T(c,d)$ if $a \leq c$ and $b \leq d$.
In functional notation, T: $[0..1],[0..1] \rightarrow [0..1]$.

Definition 3.35
A *triangular conorm* S is a function that satisfies
(boundary) $S(1,1) = 1$,
(boundary) $S(a,0) = S(0,a) = a$,
(commutativity) $S(a,b) = S(b,a)$,
(associativity) $S(a,S(b,c)) = S(S(a,b),c)$,
(monotonicity) $S(a,b) \leq S(c,d)$ if $a \leq c$ and $b \leq d$.
In functional notation, S: $[0..1],[0..1] \rightarrow [0..1]$.

Definition 3.36
A *negation norm* N is a function that satisfies
(boundary) $N(0) = 1$,
(boundary) $N(1) = 0$,
(involution) $N(N(a)) = a$,
(monotonicity) $N(a) > N(c)$ if $a < c$.
In functional notation, N: $[0..1] \rightarrow [0..1]$.

Additionally, a family of functions can be defined that satisfy a set of conditions for the negation operator [Bonissone, 1986].

This set of conditions does not completely determine the propagation function for the negation, but we restrict the discussion here to the function that is used in almost all implementations, N(a) = 1-a. With this propagation function for the negation, the following relation holds for T-norms and S-norms, T(a,b) = N(S(N(a),N(b))). Now pairs of T- and S-norms can be defined.

<u>Example 3.7</u>

$T_1(a,b) = \max\{0,a+b-1\}$	$S_1(a,b) = \min\{1,a+b\}$,
$T_2(a,b) = a*b$	$S_2(a,b) = a+b-a*b$,
$T_3(a,b) = \min\{a,b\}$	$S_3(a,b) = \max\{a,b\}$.

Given Definitions 3.32, 3.33 and 3.34, it is obvious that *1* should be interpreted as *true* and *0* as *false*. In this case, the boundary conditions of these definitions are equal to the boundary conditions resulting from the definition of a correct embedding of the three-valued model. Now, one problem remains, the definition of the *unknown* label. Assume we have chosen some arbitrary $u \in (0..1)$ for B(*u*). From Definition 3.32, it is easy to prove that T(u,1)=T(1,u)=u and T(u,0)=T(0,u)=0, from Definition 3.33 it follows that S(u,1)=S(1,u)=1 and S(u,0)=S(0,u)=u. Still, three boundary conditions remain to be satisfied: T(u,u)=u, S(u,u)=u and N(u)=u.

Given N(a)=1-a, there is only one possibility for u, and that is u=0.5. Thus, the only pair of T-norm and S-norm given in Example 3.7, which satisfies the remaining conditions, is (T_3,S_3). Other pairs of T- and S-norms violate the conditions T(u,u)=u and S(u,u)=u for this choice of u. This means that only a subset of the families of T-norms and S-norms can be embedded in the three-valued model of the general framework.

The reason for this, on first sight, disappointing conclusion is simple, the families of T- and S-norms are developed from classical logic, i.e. a two-valued system. Therefore, if we wish to construct families of T-norms and S-norms that enable a correct embedding, there are two distinguishable approaches. Firstly, we can loosen the definition of a correct embedding, for instance by dropping the constraints on *unknown* or by allowing B(*u*) to be modelled by an interval (as suggested for the (subjective) Bayesian method). Secondly, we can change the proposed conditions of the families of T-norms and conorms by starting from a three-valued system. In Chapter 4, we use this latter approach, without losing the characteristics of the theory developed on triangular norms and conorms.

Fuzzy set theory

The handling of negation, conjunction and disjunction in fuzzy set theory [Zadeh, 1965], is closely related to the theory of T-norms and S-norms, see [Weber, 1983] for an overview. For instance, the propagation functions for conjunction and disjunction that Zadeh suggested are equal to the T_3 and S_3 norms (see Example 3.7). We do not repeat this discussion here.

In fuzzy set theory, the attention has been focused on the development of propagation functions for the implication operator (the *generalised modus ponens*) and for the fuzzy relation (the *compositional rule of inference*). Since the primitive inference step of the general framework is the deductive implication, the generalised modus ponens seems appropriate to model an uncertainty propagation function for the *implication.* Numerous examples of these functions have been developed, a few examples of them are given here, see also [Weber, 1983].

Example 3.8
$f_{\Rightarrow}(a,b) = a*b$,
$f_{\Rightarrow}(a,b) = \min\{1,1-a+b\}$,
$f_{\Rightarrow}(a,b) = \max\{1-a,b\}$,
$f_{\Rightarrow}(a,b) = \max\{1-a,\min\{a,b\}\}$.

Now, what are the restrictions on the fuzzy implication function that are imposed by the embedding of the three-valued model. Obviously, when the evaluation function of the premise evaluates to *true*, the outcome of the propagation function $f_{\Rightarrow}$ should be equal to the strength of the relation. For the fuzzy implication, this results in the condition $f_{\Rightarrow}(B(t),b)=b$, where *b* is the strength of the relation and *B* is an embedding of the three-valued model.

Again, it is obvious that *1* can be interpreted as *true* and *0* as *false*. Regardless of the choice of the embedding result of *unknown*, we can see that this condition is fulfilled for each of the fuzzy implication functions from Example 3.8. This means that these functions can be used to model the uncertainty propagation function in a fuzzy extension of the general framework. Therefore, no problems are present in finding a fuzzy implication function to produce a correct embedding of the three-valued model.

3.5.4 Potential classes of supported uncertainty models

We analyse the framework by trying to describe characteristics of the uncertainty models that are supported by our framework. We pursue this analysis along the traditional lines of the representation and propagation aspects, and relate the results to the two design constraints for uncertainty models.

Concerning the *representation* aspect of potential uncertainty models, a three-valued labelling was chosen as the underlying logical labelling. Preferring a three-valued labelling to a two-valued one is the result of the possibility to explicitly represent unknown in a three-valued labelling. The unknown label is conceptually different from both *true* and *false* and cannot be expressed in any combination of these labels. The possibility to use any other N-valued logic (N>3) is encapsulated by the uncertainty type T_{ul}.

We state that the definition of an uncertainty labelling, based on uncertainty type T_{ul}, is general enough to leave open the choice between probabilistic and possibilistic typed models. In fact, concerning the *type of uncertainty*, we can go even further. There is no reason why type T_{ul} should be restricted to either of these approaches. Assuming that it is accompanied by a model-specific set of propagation functions, any T_{ul} will be supported equally well.

A second observation on the type of uncertainty T_{ul} that can be made is that it is not restricted by the framework to one-dimensional measurements. The introduction of an N-dimensional measurement, an *uncertainty vector*, consisting of equally or differently typed elements is allowed by the framework. Again, the set of propagation functions of the uncertainty model must be able to cope with these uncertainty vectors. This extension from one-dimensional to N-dimensional measurements is useful when the use of several types of uncertainty (imprecision vs. vagueness vs. incompleteness) is demanded at the same time (recall the approach that is proposed in [van der Lubbe, 1990]).

These two observations on the representation aspect of uncertainty reflect the spirit of the first design constraint on uncertainty models *(3. The representation type of the uncertainty model may not restrict the classes of potential uncertainty models).*

Concerning the *propagation* aspect of potential uncertainty models, we state that the introduction of the set of five uncertainty propagation functions ($f_{\neg}$,$f_{\wedge}$,$f_{\vee}$,$f_{\Rightarrow}$,$f_{\|}$) is only subject to the boundary conditions specified in the definition of a correct embedding (see Definition 3.28). Other conditions on propagation functions, like commutativity, associativity and montonicity, can be formulated but are no part of the general framework itself. The boundary conditions construct a minimal set of constraints that is necessary to fulfil the axioms of the underlying three-valued model. Although in this chapter a deductive strategy has been used, the framework does not impose restrictions on the reasoning type. In Chapter 5, the framework is used in combination with an abductive strategy, and is still able to use the basic notions of uncertainty propagation that have been described in this chapter.

It is clear that no conditions are formulated on independencies between propositions that are used by the five propagation functions, and therefore the second design constraint is also satisfied *(4. The propagation functions of the uncertainty model should not assume independency relations between pieces of evidence, or exclusiveness of hypotheses).* The use of independency relations in the uncertainty propagation functions of a specific uncertainty model is of course still allowed.

Chapter 4

An Uncertainty Model for Compositional Reasoning

In this chapter, we develop an uncertainty model to be used with the general framework and the deductive reasoning strategy. According to Chapter 2, several models are available. The certainty factor model is the most well-known model for compositional uncertain reasoning using a deductive control strategy. Former analyses concerning the certainty factor model have been definite in their disapproval, especially mentioning the fact that the suggested probabilistic foundations of the model are not respected by its propagation functions. Today, research on probabilistic typed models is mainly concentrated on two models: Bayesian belief networks and Dempster-Shafer theory. The arguments for our decision to analyse certainty factor theory first are given below. In the second part of the chapter, we concentrate on the development of a new model for compositional uncertain reasoning, based on results from this analysis. With this new model we introduce a set of propagation functions that take into account the situation in which they are used. The flexibility that is thus created enables a solution to the first of the benchmark problems, as described in Chapter 1.

Management of uncertainty through the use of *Bayesian belief networks* is the best method to employ when a method to construct a pure probabilistic model is demanded. However, we can also formulate some objections against their application. Firstly, there is the problem that the propagation functions used are of exponential complexity and that their effect on a local change in belief can propagate through the whole network (which can work counterintuitively, as compared to human reasoning). Furthermore, the basic measurement of Bayesian belief networks, a probability number, does not allow separate independent support both in favour of and against the same hypothesis (one of the objectives of our model). Finally, in practical applications, the model demands that a huge number of conditional probabilities is available.

The aspect of exponential computational complexity is also a strong argument against the use of the *Dempster-Shafer theory*. Furthermore, the theory lacks the propagation functions for the logical operators and even for the implication operator (although there is common agreement on one such a function). However, the basic uncertainty measurement of the theory, the belief interval consisting of a belief and a plausibility factor, is suited to be embedded in our framework as developed in Chapter 3. Both aspects are outlined in Appendix B.

Our decision to analyse parts of *certainty factor theory* is based on the following arguments. Firstly, the model has attractive characteristics, the reasoning strategy is deductive, its measures are intuitively clear and its propagation functions are computationally simple. Further, the results obtained with the model have been satisfactorily throughout the years, this should have some rationale. Since the model has been and still is applied in a great number of applications, it is important to know what are its limitations and demands with respect to specific application areas. Finally, we feel that some arguments used to reject the model, although they demonstrate important shortcomings, do not validate its complete rejection. We analyse and clarify the exact causes of this misbehaviour and specify the limitations of the model for potential application areas.

The first part of this chapter uses results that can be found in [van der Gaag, 1990]. In this work, a detailed analysis of the propagation functions of certainty factor theory is given. However, we feel that some conclusions based on this analysis need further study and that other conclusions can be drawn with respect to parts of the model. Therefore, we use a quite similar notation of the basic measurements and adopt some of the proofs.

In Section 1, some basic definitions from probability theory and of the certainty factor model are described. In Section 2, the original propagation functions of the certainty factor model are given, together with the proofs of correctness or counterexamples. A study of the logical operators in terms of the model comprises the contents of Section 3. Arguments for the abandoning of the certainty factor as uncertainty measurement are given in Section 4, and an uncertainty model is obtained that is correct with respect to its probabilistic foundation. Section 5 introduces a new uncertainty model which is based on results of the preceding sections. It is shown that enhancements can be made in uncertainty management by using the information stored in an underlying reason maintenance model (as was foreseen in Chapter 3). In Section 6, the embedding of the constructed uncertainty model in the framework of Chapter 3 is given. Finally, in Section 7 a brief discussion can be found on different aspects of the developed model.

4.1 Some basic definitions

The basic definitions of the certainty factor model were defined in terms of probability theory, therefore some basic knowledge concerning probability theory is necessary. In this section we concentrate on two important definitions with respect to uncertainty management: mutual independence and conditional independence of events (or pieces of evidence). These notions are of importance to the model as is made clear in the remainder of this chapter.

Definition 4.1
Let Ω be a sample space and P a probability function on Ω. Then two pieces of evidence $e_1,e_2 \in \Omega$ are *mutually independent* when $P(e_1 \cap e_2) = P(e_1)P(e_2)$.

Definition 4.2
Let Ω be a sample space and P a probability function on Ω. Then two pieces of evidence $e_1,e_2 \in \Omega$ are considered to be *conditionally independent* given a piece of evidence $e \in \Omega$ when $P(e_1 \cap e_2 | e) = P(e_1 | e)P(e_2 | e)$.

It should be noted that the notion of conditional independence is a separate definition and in general cannot be derived from the definitions of mutual independence and conditional probability. These two definitions lead to the following theorem.

Theorem 4.1

Let Ω be a sample space and P a probability function on Ω. When two pieces of evidence $e_1,e_2 \in \Omega$ are *mutually independent* and *conditionally independent* given a piece of evidence $e \in \Omega$ then

(a) $P(e_1 \cup e_2) = P(e_1)+P(e_2)-P(e_1)P(e_2)$.

(b) $P(e_1 \cup e_2|e) = P(e_1|e)+P(e_2|e)-P(e_1|e)P(e_2|e)$.

Proof

(a) $P(e_1 \cup e_2) = P(e_1)+P(e_2)-P(e_1 \cap e_2) = P(e_1)+P(e_2)-P(e_1)P(e_2)$.

(b) $P(e_1 \cup e_2|e) = P(e_1|e)+P(e_2|e)-P(e_1 \cap e_2|e) = P(e_1|e)+P(e_2|e)-P(e_1|e)P(e_2|e)$.

We continue with the definitions of the measurements of increased belief (MB) and increased disbelief (MD), see [Shortliffe, 1984, p. 248].

Definition 4.3

Let Ω be a sample space, P a probability function on Ω and $e,h \in \Omega$ with $e \neq \emptyset$.
The *measurement of increased belief* MB is a function such that

$$MB(h,e) \begin{array}{ll} = 1 & \text{when } P(h)=1, \\ = \max\left\{0, \dfrac{P(h|e)-P(h)}{1-P(h)}\right\} & \text{otherwise.} \end{array}$$

Definition 4.4

Let Ω be a sample space, P a probability function on Ω and $e,h \in \Omega$ with $e \neq \emptyset$.
The *measurement of increased disbelief* MD is a function such that

$$MD(h,e) \begin{array}{ll} = 1 & \text{when } P(h)=0, \\ = \max\left\{0, \dfrac{P(h)-P(h|e)}{P(h)}\right\} & \text{otherwise.} \end{array}$$

From these definitions the following properties can be derived straightforwardly, proofs can be found in [van der Gaag, 1990, p. 39].

Theorem 4.2

Let Ω be a sample space, P a probability function on Ω and $e,h \in \Omega$. Then

(a) $0 < P(h) = P(h|e) < 1 \Leftrightarrow MB(h,e) = MD(h,e) = 0$.

(b) $MB(h,e) > 0 \Rightarrow MD(h,e) = 0$.

(c) $MD(h,e) > 0 \Rightarrow MB(h,e) = 0$.

According to the original definitions of the certainty factor model, the certainty factor can now be defined.

Definition 4.5

Let Ω be a sample space, P a probability function on Ω and $e,h \in \Omega$.
The *certainty factor* CF is a function such that $CF(h,e) = MB(h,e)-MD(h,e)$.

4.2 The propagation functions of the original model

In [Shortliffe, 1984, p. 255] we find a set of propagation functions to update the measurements of belief and disbelief for the operators: disjunction, conjunction, (deductive) implication and the combination of pieces of evidence. We briefly discuss these functions here.

The following theorem describes the propagation function for the implication operator, the proof is given in [van der Gaag, 1990, pp. 60-63].

> Theorem 4.3
> Let Ω be a sample space, P a probability function on Ω and $e,e_1,h \in \Omega$.
> If $h \subseteq e_1 \subseteq e$, then the propagation functions for the *(deductive) implication* $e_1 \Rightarrow h$, given e, are given by
> $MB(h,e) = MB(e_1,e)MB(h,e_1)$,
> $MD(h,e) = MB(e_1,e)MD(h,e_1)$,
> $CF(h,e) = \max\{0,CF(e_1,e)\}CF(h,e_1)$.

The impact of the condition $h \subseteq e_1 \subseteq e$ on the reasoning process is that this process is narrowing, i.e. whenever *h* is valid, e_1 must be valid and so must be *e*.

A correct set of propagation functions for the combination operator is given in Theorem 4.4, for a proof is again referred to [van der Gaag, 1990, pp. 50-57].

> Theorem 4.4
> Let Ω be a sample space, P a probability function on Ω and $e_1,e_2,h \in \Omega$.
> If e_1 and e_2 are mutually independent and conditionally independent given h
> then the propagation functions for the *combination* of h, given e_1 and e_2, are given by
> $MB(h,e_1||e_2) = MB(h,e_1)+MB(h,e_2)-MB(h,e_1)MB(h,e_2)$,
> $MD(h,e_1||e_2) = MD(h,e_1)+MD(h,e_2)-MD(h,e_1)MD(h,e_2)$,
> $CF(h,e_1||e_2) =$
> $CF(h,e_1)+CF(h,e_2)-CF(h,e_1)CF(h,e_2)$ when $CF(h,e_1),CF(h,e_2) \geq 0$,
> $CF(h,e_1)+CF(h,e_2)/(1-\min\{|CF(h,e_1)|,|CF(h,e_2)|\})$ when $-1 < CF(h,e_1)CF(h,e_2) \leq 0$,
> $CF(h,e_1)+CF(h,e_2)+CF(h,e_1)CF(h,e_2)$ when $CF(h,e_1),CF(h,e_2) \leq 0$,
> undefined when $CF(h,e_1)CF(h,e_2)=-1$.

According to Definition 4.5, Theorem 4.4 implies that the first and third part of the function for $CF(h,e_1||e_2)$ are also correct under these conditions. The second part of this function however cannot be proved to hold under the given conditions. More important, no set of conditions or assumptions can be found that imply the given combination function for positive and negative evidence.

These independency conditions have their impact on the applicability of the model, since they have to be checked for all possible occurrences of possible combinations of hypotheses. This can be quite a burden when developing the domain model in some application area.

In summary, the propagation functions for both implication and combination, as defined by the original model, have been given and, under some conditions, have been proved to respect the probabilistic definitions of MB and MD, given in Definitions 4.3 and 4.4. However, when we take a look at the original definitions of the propagation functions for the disjunction and conjunction operators, we are able to construct examples that illustrate the incorrectness of these functions. First, the original functions are given.

The propagation functions for the *disjunction* $e_1 \vee e_2$ are given by
$MB(e_1 \vee e_2, e) = \max\{MB(e_1,e), MB(e_2,e)\}$,
$MD(e_1 \vee e_2, e) = \min\{MD(e_1,e), MD(e_2,e)\}$,
$CF(e_1 \vee e_2, e) = \max\{CF(e_1,e), CF(e_2,e)\}$.

The propagation functions for the *conjunction* $e_1 \wedge e_2$ are given by
$MB(e_1 \wedge e_2, e) = \min\{MB(e_1,e), MB(e_2,e)\}$,
$MD(e_1 \wedge e_2, e) = \max\{MD(e_1,e), MD(e_2,e)\}$,
$CF(e_1 \wedge e_2, e) = \min\{CF(e_1,e), CF(e_2,e)\}$.

In [van der Gaag, 1990, pp. 63-65], some very simple counterexamples are generated for these propagation functions. Even under extreme conditions, like $e_1 \subseteq e_2$, it cannot be shown that the probabilistic definitions of MB and MD are respected when the above propagation functions are used.

4.3 New propagation functions for the logical operators

Shortliffe and Buchanan [1984] themselves pointed out that the propagation functions for the disjunction as well as for the conjunction were pragmatic choices. Therefore, it is not a new criticism that the measurements of belief and disbelief of expressions, consisting of disjunctions and conjunctions, are not correct with respect to the probabilistic definition of the model. In this section, we analyse a set of intuitively appealing interpretations for the set of logical operators $\{\neg, \wedge, \vee\}$[1].

4.3.1 The negation operator

Neither [Shortliffe, 1984] nor [van der Gaag, 1990] contains an explicit propagation function for the negation operator. On p. 250, Shortliffe and Buchanan state:

> "Clearly, this result occurs because (for any h and any e) $MB(h,e) = MD(\neg h,e)$. This conclusion is intuitively appealing since it states that evidence that supports a hypothesis disfavours the negation of the hypothesis to an equal extent."

This remark concerning the propagation function for the negation of MB and MD can easily be derived from the definitions when the complement $\bar{h}$ is used for the interpretation of the $\neg h$. This is described by the following theorem, the proof is straightforward and therefore left to the reader.

[1] For reasons of clarity, these operators are defined on sets and they are therefore unequal to the logical operators defined on expressions as defined in Chapter 3. The mapping of the model onto the general framework is explained in Section 6.

Theorem 4.5

Let Ω be a sample space, P a probability function on Ω and $h \in \Omega$, let MB and MD be defined according to Definitions 4.3 and 4.4, and let negation $\neg h$ be interpreted by the complement $\bar{h}$.

Then the propagation functions for the *negation* are given by

(a) $MB(\neg h,e) = MD(h,e)$.

(b) $MD(\neg h,e) = MB(h,e)$.

4.3.2 The disjunction and conjunction operators

In [van der Gaag, 1990], the original propagation functions for the disjunction and conjunction operators have been shown to be incorrect. Further, it is emphasised that conjunction and disjunction operators are not defined in probability theory. Therefore, interpretations are necessary from the logical operators to set operations on the sample space Ω. The interpretation mapping that is used for the logical operators in [van der Gaag, 1990, pp. 33-34] is intuitively correct, the union is used to interpret the disjunction and the intersection is used for the interpretation of the conjunction. In the following, we take a closer look at this interpretation problem and present some theorems for the propagation functions of the logical operators.

The union seems to be an intuitively appealing interpretation for the disjunction. We use this interpretation in the following theorem.

Theorem 4.6

Let Ω be a sample space, P a probability function on Ω and $e_1,e_2,e \in \Omega$, let MB be defined according to Definition 4.3 and let disjunction $e_1 \vee e_2$ be interpreted by the union $e_1 \cup e_2$, resulting in

$$MB(e_1 \vee e_2,e) \begin{array}{ll} = 1 & \text{when } P(e_1 \cup e_2)=1, \\ = \max\left\{0, \dfrac{P(e_1 \cup e_2|e)-P(e_1 \cup e_2)}{1-P(e_1 \cup e_2)}\right\} & \text{otherwise.} \end{array}$$

If e_1 and e_2 are mutually independent and conditionally independent given e then the propagation function for the MB of the *disjunction* is equal to

$$MB(e_1 \vee e_2,e) = MB(e_1,e)+MB(e_2,e)-MB(e_1,e)MB(e_2,e).$$

Proof

We distinguish two cases.

(i) $P(e_1 \cup e_2)=1 \Rightarrow P(e_1)=1$ or $P(e_2)=1 \Rightarrow MB(e_1,e)=1$ or $MB(e_2,e)=1 \Rightarrow$
$MB(e_1 \vee e_2,e) = MB(e_1,e)+MB(e_2,e)-MB(e_1,e)MB(e_2,e) = 1.$

(ii) $P(e_1 \cup e_2)<1 \Rightarrow$

$$\frac{P(e_1 \cup e_2|e)-P(e_1 \cup e_2)}{1-P(e_1 \cup e_2)} =$$

$$= \frac{P(e_1|e)+P(e_2|e)-P(e_1|e)P(e_2|e)-P(e_1)-P(e_2)+P(e_1)P(e_2)}{1-P(e_1)-P(e_2)+P(e_1)P(e_2)} =$$

$$=\frac{P(e_1|e)-P(e_1)+P(e_2|e)-P(e_2)+P(e_1)P(e_2)-P(e_1|e)P(e_2|e)}{(1-P(e_1))(1-P(e_2))}=$$

$$=\frac{MB(e_1,e)}{(1-P(e_2))}+\frac{MB(e_2,e)}{(1-P(e_1))}+\frac{P(e_1)P(e_2)-P(e_1|e)P(e_2|e)}{(1-P(e_1))(1-P(e_2))}\quad [1].$$

Taking the last term apart:

$$\frac{P(e_1)P(e_2)-P(e_1|e)P(e_2|e)}{(1-P(e_1))(1-P(e_2))}=$$

$$=\frac{P(e_1)P(e_2)-(MB(e_1,e)(1-P(e_1))+P(e_1))(MB(e_2,e)(1-P(e_2))+P(e_2))}{(1-P(e_1))(1-P(e_2))}=$$

$$=\frac{P(e_1)P(e_2)-(MB(e_1,e)-MB(e_1,e)P(e_1)+P(e_1))(MB(e_2,e)-MB(e_2,e)P(e_2)+P(e_2))}{(1-P(e_1))(1-P(e_2))}=$$

$$=\frac{P(e_1)P(e_2)-MB(e_1,e)MB(e_2,e)+MB(e_1,e)MB(e_2,e)P(e_2)-MB(e_1,e)P(e_2)}{(1-P(e_1))(1-P(e_2))}+$$

$$+\frac{MB(e_1,e)MB(e_2,e)P(e_1)-MB(e_1,e)MB(e_2,e)P(e_1)P(e_2)+MB(e_1,e)P(e_1)P(e_2)}{(1-P(e_1))(1-P(e_2))}+$$

$$+\frac{MB(e_2,e)P(e_1)P(e_2)-MB(e_1,e)P(e_2)-P(e_1)P(e_2)}{(1-P(e_1))(1-P(e_2))}=$$

$$=-\frac{MB(e_1,e)MB(e_2,e)(1-P(e_1))(1-P(e_2))}{(1-P(e_1))(1-P(e_2))}$$

$$-\frac{MB(e_1,e)P(e_2)(1-P(e_1))}{(1-P(e_1))(1-P(e_2))}-\frac{MB(e_2,e)P(e_1)(1-P(e_2))}{(1-P(e_1))(1-P(e_2))}=$$

$$=-MB(e_1,e)MB(e_2,e)-\frac{MB(e_1,e)P(e_2)}{(1-P(e_2))}-\frac{MB(e_2,e)P(e_1)}{(1-P(e_1))}.$$

Using this result in [1] gives

$$=\frac{MB(e_1,e)}{(1-P(e_2))}+\frac{MB(e_2,e)}{(1-P(e_1))}-MB(e_1,e)MB(e_2,e)-\frac{MB(e_1,e)P(e_2)}{(1-P(e_2))}-\frac{MB(e_2,e)P(e_1)}{(1-P(e_1))}=$$

$$=MB(e_1,e)+MB(e_2,e)-MB(e_1,e)MB(e_2,e).$$

Using the same interpretation for the MD of the disjunction leads to the following theorem.

Theorem 4.7

Let Ω be a sample space, P a probability function on Ω and $e_1,e_2,e \in \Omega$, let MD be defined according to Definition 4.4 and let disjunction $e_1 \vee e_2$ be interpreted by the union $e_1 \cup e_2$, resulting in

$$MD(e_1 \vee e_2,e)\begin{array}{ll} = 1 & \text{when } P(e_1 \cup e_2)=0, \\ = \max\left\{0,\dfrac{P(e_1 \cup e_2)-P(e_1 \cup e_2|e)}{P(e_1 \cup e_2)}\right\} & \text{otherwise.} \end{array}$$

If e_1 and e_2 are mutually independent and conditionally independent given e then the propagation function for the MD of the *disjunction* is equal to $MD(e_1 \vee e_2,e) = MD(e_1,e)MD(e_2,e)+R_1+R_2$ where[2]

$$R_1 = \frac{P(e_1|e)(P(e_2)-P(e_2|e))(1-P(e_1))}{P(e_1)(P(e_1)+P(e_2)-P(e_1)P(e_2))} \text{ and } R_2 = \frac{P(e_2|e)(P(e_1)-P(e_1|e))(1-P(e_2))}{P(e_2)(P(e_1)+P(e_2)-P(e_1)P(e_2))}.$$

Proof

Again, we distinguish two cases.

(i) $P(e_1 \cup e_2)=0 \Rightarrow P(e_1)=0$ and $P(e_2)=0 \Rightarrow MD(e_1,e)=1$ and $MD(e_2,e)=1$ and $R_1=R_2=0 \Rightarrow MD(e_1 \vee e_2,e) = MD(e_1,e)MD(e_2,e)+R_1+R_2 = 1$.

(ii) $P(e_1 \cup e_2)>0 \Rightarrow$

$$\frac{P(e_1 \cup e_2)-P(e_1 \cup e_2|e)}{P(e_1 \cup e_2)} =$$

$$= \frac{P(e_1)+P(e_2)-P(e_1)P(e_2)-P(e_1|e)-P(e_2|e)+P(e_1|e)P(e_2|e)}{P(e_1)+P(e_2)-P(e_1)P(e_2)} =$$

$$= \frac{P(e_1)^2P(e_2)+P(e_1)P(e_2)^2-P(e_1)^2P(e_2)^2}{P(e_1)P(e_2)(P(e_1)+P(e_2)-P(e_1)P(e_2))} +$$

$$- \frac{P(e_1)P(e_2)P(e_1|e)+P(e_1)P(e_2)P(e_2|e)+P(e_1)P(e_2)P(e_1|e)P(e_2|e)}{P(e_1)P(e_2)(P(e_1)+P(e_2)-P(e_1)P(e_2))} \quad [1].$$

$$\text{Since } MD(e_1,e)MD(e_2,e) = \frac{(P(e_1)-P(e_1|e))(P(e_2)-P(e_2|e))}{P(e_1)P(e_2)} =$$

$$= \frac{(P(e_1)-P(e_1|e))(P(e_2)-P(e_2|e))(P(e_1)+P(e_2)-P(e_1)P(e_2))}{P(e_1)P(e_2)(P(e_1)+P(e_2)-P(e_1)P(e_2))} \quad [2].$$

Now [1] - [2] gives

$$\frac{P(e_2)P(e_1|e)P(e_2|e)-P(e_2)^2P(e_1|e)+P(e_1)P(e_2)^2P(e_1|e)}{P(e_1)P(e_2)(P(e_1)+P(e_2)-P(e_1)P(e_2))}$$

$$+ \frac{P(e_1)P(e_1|e)P(e_2|e)-P(e_1)^2P(e_2|e)+P(e_1)^2P(e_2)P(e_2|e)}{P(e_1)P(e_2)(P(e_1)+P(e_2)-P(e_1)P(e_2))}$$

$$- \frac{2\,P(e_1)P(e_2)P(e_1|e)P(e_2|e)}{P(e_1)P(e_2)(P(e_1)+P(e_2)-P(e_1)P(e_2))} =$$

$$= \frac{P(e_1|e)(P(e_2)-P(e_2|e))(1-P(e_1))}{P(e_1)(P(e_1)+P(e_2)-P(e_1)P(e_2))} + \frac{P(e_2|e)(P(e_1)-P(e_1|e))(1-P(e_2))}{P(e_2)(P(e_1)+P(e_2)-P(e_1)P(e_2))} = R_1+R_2.$$

Theorem 4.7 introduces two rest terms that disturb the concept of modelling the propagation functions by using MB and MD terms only.

The intersection seems to be an intuitively appealing interpretation for the conjunction.

[2] It should be noted that $0 \leq R_1 \leq 1$, $0 \leq R_2 \leq 1$ and $0 \leq R_1+R_2 \leq 1$.

Theorem 4.8

Let Ω be a sample space, P a probability function on Ω and $e_1,e_2,e \in \Omega$, let MB be defined according to Definition 4.3 and let conjunction $e_1 \wedge e_2$ be interpreted by the intersection $e_1 \cap e_2$, resulting in

$$MB(e_1 \wedge e_2,e) \begin{array}{ll} = 1 & \text{when } P(e_1 \cap e_2)=1, \\ = \max\left\{0, \dfrac{P(e_1 \cap e_2|e)-P(e_1 \cap e_2)}{1-P(e_1 \cap e_2)}\right\} & \text{otherwise.} \end{array}$$

If e_1 and e_2 are mutually independent and conditionally independent given e then the propagation function for the MB of the *conjunction* is equal to

$MB(e_1 \wedge e_2,e) = MB(e_1,e)MB(e_2,e)+R_1+R_2$ where[3]

$$R_1 = \frac{P(e_1)(P(e_2|e)-P(e_2))(1-P(e_1|e))}{(1-P(e_1))(1-P(e_1)P(e_2))} \text{ and } R_2 = \frac{P(e_2)(P(e_1|e)-P(e_1))(1-P(e_2|e))}{(1-P(e_2))(1-P(e_1)P(e_2))}.$$

Proof

We distinguish two cases.

(i) $P(e_1 \cap e_2)=1 \Rightarrow P(e_1)=1$ and $P(e_2)=1 \Rightarrow MB(e_1,e)=1$ and $MB(e_2,e)=1$ and $R_1=R_2=0 \Rightarrow$ $MB(e_1 \wedge e_2,e) = MB(e_1,e)MB(e_2,e)+R_1+R_2 = 1$.

(ii) $P(e_1 \cap e_2)<1 \Rightarrow$

$$\frac{P(e_1 \cap e_2|e)-P(e_1 \cap e_2)}{1-P(e_1 \cap e_2)} = \frac{P(e_1|e)P(e_2|e)-P(e_1)P(e_2)}{1-P(e_1)P(e_2)} =$$

$$= \frac{(P(e_1|e)P(e_2|e)-P(e_1)P(e_2))(1-P(e_1))(1-P(e_2))}{(1-P(e_1)P(e_2))(1-P(e_1))(1-P(e_2))} =$$

$$= \frac{P(e_1|e)P(e_2|e)+P(e_1)^2P(e_2)+P(e_1)P(e_2)^2+P(e_1)P(e_2)P(e_1|e)P(e_2|e)}{(1-P(e_1)P(e_2))(1-P(e_1))(1-P(e_2))} +$$

$$- \frac{P(e_1)P(e_2)+P(e_1)P(e_1|e)P(e_2|e)+P(e_2)P(e_1|e)P(e_2|e)+P(e_1)^2P(e_2)^2}{(1-P(e_1)P(e_2))(1-P(e_1))(1-P(e_2))} \quad [1].$$

$$\text{Since } MB(e_1,e)MB(e_2,e) = \frac{(P(e_1|e)-P(e_1))(P(e_2|e)-P(e_2))}{(1-P(e_1))(1-P(e_2))} =$$

$$= \frac{P(e_1|e)P(e_2|e)+P(e_1)^2P(e_2)P(e_2|e)+P(e_1)P(e_2)^2P(e_1|e)+P(e_1)P(e_2)}{(1-P(e_1)P(e_2))(1-P(e_1))(1-P(e_2))} +$$

$$- \frac{P(e_1)P(e_2)P(e_1|e)P(e_2|e)+P(e_1)P(e_2|e)+P(e_2)P(e_1|e)+P(e_1)^2P(e_2)^2}{(1-P(e_1)P(e_2))(1-P(e_1))(1-P(e_2))} \quad [2].$$

Now [1] - [2] gives

$$= \frac{P(e_1)(1-P(e_2))(P(e_2|e)-P(e_2))(1-P(e_1|e))}{(1-P(e_1)P(e_2))(1-P(e_1))(1-P(e_2))} + \frac{P(e_2)(1-P(e_1))(P(e_1|e)-P(e_1))(1-P(e_2|e))}{(1-P(e_1)P(e_2))(1-P(e_1))(1-P(e_2))} =$$

[3] Again, it should be noted that $0 \le R_1 \le 1$, $0 \le R_2 \le 1$ and $0 \le R_1+R_2 \le 1$.

$$= \frac{P(e_1)(P(e_2|e)-P(e_2))(1-P(e_1|e))}{(1-P(e_1)P(e_2))(1-P(e_1))} + \frac{P(e_2)(P(e_1|e)-P(e_1))(1-P(e_2|e))}{(1-P(e_1)P(e_2))(1-P(e_2))} = R_1+R_2.$$

Again, two rest terms are introduced by this interpretation. Using the same interpretation for the MD of the *conjunction* leads to the following theorem.

Theorem 4.9

Let Ω be a sample space, P a probability function on Ω and $e_1,e_2,e \in \Omega$, let MD be defined according to Definition 4.4 and let conjunction $e_1 \wedge e_2$ be interpreted by the intersection $e_1 \cap e_2$, resulting in

$$MD(e_1 \wedge e_2,e) \begin{array}{ll} = 1 & \text{when } P(e_1 \cap e_2)=0, \\ = \max\left\{0, \dfrac{P(e_1 \cap e_2)-P(e_1 \cap e_2|e)}{P(e_1 \cap e_2)}\right\} & \text{otherwise.} \end{array}$$

If e_1 and e_2 are mutually independent and conditionally independent given e then the propagation function for the MD of the *conjunction* is equal to

$$MD(e_1 \wedge e_2,e) = MD(e_1,e)+MD(e_2,e)-MD(e_1,e)MD(e_2,e).$$

Proof

We distinguish two cases.

(i) $P(e_1 \cap e_2)=0 \Rightarrow P(e_1)=0$ or $P(e_2)=0 \Rightarrow MD(e_1,e)=1$ or $MD(e_2,e)=1 \Rightarrow$
$MD(e_1 \wedge e_2,e) = MD(e_1,e)+MD(e_2,e)-MD(e_1,e)MD(e_2,e) = 1.$

(ii) $P(e_1 \cap e_2)>0 \Rightarrow$

$$\frac{P(e_1 \cap e_2)-P(e_1 \cap e_2|e)}{P(e_1 \cap e_2)} = \frac{P(e_1)P(e_2)-P(e_1|e)P(e_2|e)}{P(e_1)P(e_2)} =$$

$$= \frac{P(e_1)P(e_2)-P(e_1|e)P(e_2)+P(e_1|e)P(e_2)-P(e_1|e)P(e_2|e)}{P(e_1)P(e_2)} +$$

$$+ \frac{P(e_1)P(e_2)-P(e_2|e)P(e_1)+P(e_2|e)P(e_1)-P(e_1)P(e_2)}{P(e_1)P(e_2)} =$$

$$= \frac{P(e_1)-P(e_1|e)}{P(e_1)} + \frac{P(e_2)-P(e_2|e)}{P(e_2)} + \frac{P(e_1)P(e_2|e)+P(e_1|e)P(e_2)-P(e_1)P(e_2)-P(e_1|e)P(e_2|e)}{P(e_1)P(e_2)} =$$

$$= MD(e_1,e)+MD(e_2,e)-MD(e_1,e)MD(e_2,e).$$

Thus, under some conditions and with the commonly accepted interpretations, two propagation functions for the logical operators can be derived in a straightforward manner and two other propagation functions introduce rest terms, which are not available during reasoning. So even under the severe dependency conditions, it is impossible to provide an attractive set of propagation functions. This observation leaves us with two possibilities: change the used interpretations or reject Definitions 4.3 and 4.4 for the basic measurements. The former requires a complex probabilistic analysis, without any perspective of losing the severe dependency conditions, therefore, in the remainder of this chapter we concentrate on the latter possibility.

4.4 A final discussion on certainty factors

The suggested probabilistic foundation of the certainty factor model has been the subject of serious discussion over the years, see for instance [Shortliffe, 1975, 1984; Adams, 1976, 1984; Heckerman, 1986; Horvitz, 1986; van der Gaag, 1990; Qiu, 1992; Heckerman, 1992]. The emphasis of this discussion has been on the definition of the certainty factor and the (probabilistic) assumptions made with respect to the arguments of the propagation functions. In this section, we summarise our analysis in terms of these comments.

In the previous sections we adapted the certainty factor model such that it satisfies the axioms of probability theory. Although we can be satisfied with this result from a theoretical viewpoint, two important problems remain to be solved. Firstly, when constructing domain models, the probabilistic conditions of the propagation functions have to be taken into account. Secondly, when these conditions cannot be met, how can the propagation functions be adjusted without violating the probabilistic axioms?

Checking the probabilistic conditions of the propagation functions for all possible combinations of pieces of evidence in a given domain model is, in theory, possible. However, checking these conditions in practice would drastically increase the time needed for the development of complex domain models. It is even questionable whether these domain models can be constructed under these conditions at all.

When the independence conditions (see Theorems 4.6 through 4.9) cannot be guaranteed, the propagation functions should be able to cope with situations in which dependent pieces of evidence are provided. Observing the proofs of these theorems, we can conclude that when other propagation functions can be specified they will certainly contain prior and conditional probabilities[4] of the pieces of evidence involved. This results in the loss of one of the most valuable characteristics of the model: the fact that only the measurements of belief and disbelief are used in the propagation functions.

Another criticism concerns the semantics of the arguments of the certainty factor. Consider the propagation function for the combination of evidence, given in Theorem 4.4.

> Now, when $MB(h,e_1) > 0$ and $MD(h,e_2) > 0$,
> it appears that both $MB(h,e_1||e_2) = MB(h,e_1) > 0$ and $MD(h,e_1||e_2) = MD(h,e_2) > 0$,
> which is in contradiction with (b) and (c) of Theorem 4.2.

The reason for this behaviour is that no history of the individual pieces of evidence is kept, and therefore it cannot be seen that $MB(h,e_1||e_2)$ and $MD(h,e_1||e_2)$ are not both supported by the combination of e_1 and e_2, but that each of them is supported by one of the other. This favours the introduction of a derivation argument in the definitions of the measurements of belief and disbelief (see also [van der Gaag, 1990, p. 36]). We agree on the observation, but we argue that the introduction of an extra argument is not necessary when one changes the semantics of the

[4] Remember the rest terms in Theorems 4.7 and 4.8.

second argument of the measurements of belief and disbelief. We change the interpretation of this argument from the environment that is responsible for increase or decrease of belief to a *structured* environment. This interpretation of an environment not only contains the complete set of propositions used in the evidence, but also the way in which the evidence is structured, i.e. which (logical) operations have been used to combine the propositions in order to obtain the evidence. Therefore, this interpretation of the environment implicitly represents the notion of a derivation argument as proposed by van der Gaag.

A final criticism can be made of the method of using one separate measurement (the certainty factor) to represent two different notions (belief and disbelief in a proposition). As we feel it, the interpretation of both measurements can best be done by the user of the results, since he or she has the insight in the semantics of each individual atomic proposition and expression, and can therefore make the better interpretation.

Solving the mentioned problems in the given probabilistic setting of the model does not seem to be a promising objective. We therefore decide to abandon the probabilistic foundation of the model developed so far. Although this seems to be quite a thing to do, we can make the following remarks concerning this decision. The probabilistic foundation was not the starting point of the development of the original model. It was developed afterwards, when trying to provide some theoretical basis for the model and a well-defined semantics for the measurements of belief and disbelief. Not only the foundation into probability theory has been subject of discussion (see for instance [Adams, 1976, 1984; Heckerman, 1986; van der Gaag, 1990]), but also the semantics have more than once been the subject of discussion (see for example [Horvitz, 1986; Heckerman, 1992]), since it is very tempting to interpret the belief measurements as conditional probabilities which they are not. In fact, it is easy to construct situations in which a proposition has a greater conditional probability than another proposition, while it has a measurement of belief that is smaller. Maintaining the probabilistic foundation for reasons of semantics is therefore not an obvious choice.

In conclusion can be stated that the advantages of formulating a probabilistic foundation for the model are outweighed by the disadvantages. Therefore, we decide to abandon the probabilistic foundation of the model of belief and disbelief (i.e. we no longer take Definitions 4.3 and 4.4 as the definitions of our measurements). In the next section, we introduce a possibilistic foundation, and develop the model from this perspective.

4.5 The model of belief and disbelief

As a result of the remarks made in the preceding section, we redefine the measurements of belief and disbelief as possibilistic measurements. This introduces the problem of defining a semantic for the measurements. For instance, the linguistic term set semantic [Bonissone, 1986], or fuzzy number approach [Zadeh, 1978] can be used for this purpose. In the model, we use just numbers from [0..1], leaving out the choice for a specific semantic. A set of propagation functions is developed that can be used on the measurements. Contrary to the set of propagation functions defined in the probabilistic model, no restrictions are introduced by these propagation functions.

Although the switch from a probabilistic to a possibilistic model is fundamental, it does not mean that we have to completely reject the results of the preceding sections. In the following subsections, we carefully examine which parts of the model can still be used in this new setting. For instance, the definition of the uncertainty label (MB,MD) as the uncertainty label T_{ul} can still be used. This definition of the uncertainty model shows one of the main characteristics of the model of belief and disbelief, evidence supporting as well as evidence refuting a hypothesis are represented in separate measurements. Contrary to the probabilistic approaches, these measurements are not subject to restrictions like $P(A)+P(\overline{A})=1$.

The term derivation is explained and formally defined, after which it is used in the possibilistic definition of the measurements of belief and disbelief. This is followed by an extension of the theory of triangular norms and conorms to enable the handling uncertainty labels of type (MB,MD). Next, a definition is given of independence, when used in the possibilistic context. Then, the set of previously developed propagation functions is examined, and checked for validity with respect to triangular norm theory and the new notion of independence. Finally, we develop a set of possibilistic propagation functions, and show that these functions can be used to solve the first benchmark problem, as described in Chapter 1.

4.5.1 Defining the derivation

In our analysis of the certainty factor model, it was acknowledged that, apart from the proposition it is defined upon, a second argument can be helpful in the definition of an uncertainty measurement. This second argument should reflect the environment in which the uncertainty for the proposition has been determined. In the context of Chapter 3, such an environment for propositions is not defined, we therefore introduce the notion of a derivation here. A derivation is a subset of the set of atomic propositions A and contains all propositions that have been used when deriving the labelling of a proposition or an expression.

For an atomic proposition (see Definition 3.1), there are three situations in which the derivation has to be determined. The proposition is part of the initial labelling, it is in the conclusion of a deductive inference step or a combination of these situations. When the proposition is labelled by the initial labelling, its derivation is defined to be empty, since no other proposition has contributed to its labelling. When the proposition is labelled by a primitive inference step $(e \Rightarrow a)$, the derivation is equal to the derivation of the premise expression (e). Obviously, the inference step must have been labelled $\mathcal{T}$ by the inference function $I\mathcal{F}$. When the proposition is labelled by a combination of two or more contributions, the derivation is equal to the union of the derivations of the contributors.

For an expression (see Definition 3.2), the derivation is based on the derivations of the atomic propositions in the subexpressions. The derivation of such a proposition is only enclosed when it has contributed to the evaluation of the expression (i.e. its label is *true* (t) or *false*(f)), this means that propositions labelled *unknown* (u) do not influence the derivation of the expression they are used in.

Definition 4.6
A *derivation* d is a subset of the set of atomic propositions A. Derivations can be specified for both atomic propositions and expressions.

A *derivation of an atomic proposition* d_a is defined as[5]
(initial) $d_a = \emptyset$ when a is labelled by the initial labelling (a is a premise),
(primitive) $d_a = d_e$ when a is labelled by an inference step and $I\mathcal{J}(e \Rightarrow a)=\mathcal{T}$,
(combination) $d_a = d_{a_1} \cup d_{a_2}$ when there are two contributions for a.

A *derivation of an expression* d_e is defined as
(atomic) $d_e = d_a \cup \{a\}$ when $L(a) \in \{t,f\}$, $d_e = \emptyset$ when $L(a) \in \{u\}$,
(negation) $d_e = d_{e_1}$ when $e = \neg e_1$,
(conjunction) $d_e = d_{e_1} \cup d_{e_2}$ when $e = e_1 \wedge e_2$,
(disjunction) $d_e = d_{e_1} \cup d_{e_2}$ when $e = e_1 \vee e_2$.

The set of derivations is called D.

As a result of this definition, we can use the terms $MB(a,d_a)$ and $MD(a,d_a)$ to denote the measurements of belief and disbelief in terms of the general framework.

4.5.2 The measurements of belief and disbelief

As a result of the arguments given in the beginning of this section, we decided to abandon the probabilistic foundation of the model of belief and disbelief (i.e. we no longer use Definitions 4.3 and 4.4). Instead, we redefine the measurements of belief and disbelief in a possibilistic context. In Definition 4.7, the measurements of belief $MB(e,d_e)$ and disbelief $MD(e,d_e)$ are specified, again based on two arguments. The first argument, reflecting the hypothesis, is represented by an expression[6], the second argument, reflecting the environment, is represented by a derivation. Now, the measurements of belief and disbelief are defined on the interval [0..1], yielding the following definition.

Definition 4.7
The *measurement of belief* $MB(e,d_e)$ and the *measurement of disbelief* $MD(e,d_e)$ are defined as the assignment of a label from [0..1] to each tuple, consisting of an expression and its derivation.
In functional notation, $MB,MD\colon E,D \rightarrow [0..1]$.

[5] It should be noted that in general $a \notin d_a$.

[6] Remember that the set of atomic propositions A is subsumed by the set of expressions E.

Apart from being possibilistic measurements, there is a lack of a semantics for both $MB(a,d_a)$ and $MD(a,d_a)$. The reason for this is twofold, we do not want to restrict ourselves to a specific semantic and we would like as few demands as possible when developing propagation functions. This leaves the choice of a semantic to the user of the framework, which at the same time introduces one of the following two problems. Firstly, an existing semantic can be chosen (for instance by using entropy functions), resulting in the need for checking whether the propagation functions specified by the framework remain valid. Secondly, a new semantic can be developed that must coincide with the semantic used in the development of our propagation functions.

4.5.3 Triangular norms and conorms for the model of belief and disbelief

Now that we have a possibilistic definition of the measurements, we can proceed with the development of the propagation functions. Recalling Chapter 2, families of triangular norms and conorms constitute the most general propagation functions for the logical operators in possibilistic reasoning. Conditions are formulated for triangular norms (T-norms), triangular conorms (S-norms) and a norm on the negation operator (N-norm). In Chapter 3, the definitions according to [Bonissone, 1986] were given for one-dimensional norms.

In our model, we use a two-dimensional uncertainty label (MB,MD), where triangular norms are based on a single uncertainty value. Here, we adjust the definitions, given in Chapter 3, so that they can be used on uncertainty labels. We take (0,1) and (1,0) to represent 0 (*false*) and 1 (*true*) respectively. In fact, we have made a preliminary choice for the embedding of the model (see Section 6).

Definition 4.8

A *triangular norm* T is a function that satisfies[7]

(boundary) $T((0,1),(0,1)) = (0,1)$,
(boundary) $T(\mathbf{a},(1,0)) = T((1,0),\mathbf{a}) = \mathbf{a}$,
(commutativity) $T(\mathbf{a},\mathbf{b}) = T(\mathbf{b},\mathbf{a})$,
(associativity) $T(\mathbf{a},T(\mathbf{b},\mathbf{c})) = T(T(\mathbf{a},\mathbf{b}),\mathbf{c})$,
(monotonicity) $T(\mathbf{a},\mathbf{b}) \leq_M T(\mathbf{c},\mathbf{d})$ if $\mathbf{a} \leq_M \mathbf{c}$ and $\mathbf{b} \leq_M \mathbf{d}$.[8]
In functional notation, T: ([0..1],[0..1]),([0..1],[0..1]) → ([0..1],[0..1]).

Definition 4.9

A *triangular conorm* S is a function that satisfies

(boundary) $S((1,0),(1,0)) = (1,0)$,
(boundary) $S(\mathbf{a},(0,1)) = S((0,1),\mathbf{a}) = \mathbf{a}$,
(commutativity) $S(\mathbf{a},\mathbf{b}) = S(\mathbf{b},\mathbf{a})$,
(associativity) $S(\mathbf{a},S(\mathbf{b},\mathbf{c})) = S(S(\mathbf{a},\mathbf{b}),\mathbf{c})$,
(monotonicity) $S(\mathbf{a},\mathbf{b}) \leq_M S(\mathbf{c},\mathbf{d})$ if $\mathbf{a} \leq_M \mathbf{c}$ and $\mathbf{b} \leq_M \mathbf{d}$.
In functional notation, S: ([0..1],[0..1]),([0..1],[0..1]) → ([0..1],[0..1]).

[7] In this, $\mathbf{a}=(MB_a,MD_a)$, $\mathbf{b}=(MB_b,MD_b)$, $\mathbf{c}=(MB_c,MD_c)$ and $\mathbf{d}=(MB_d,MD_d)$.

[8] The $\leq_M$ relation is defined as $(MB_a,MD_a) \leq_M (MB_c,MD_c)$ when $MB_a \leq MB_c$ and $MD_a \leq MD_c$.

Definition 4.10

A *negation norm* N is a function that satisfies

(boundary) $N((0,1)) = (1,0)$,

(boundary) $N((1,0)) = (0,1)$,

(involution) $N(N(\mathbf{a})) = \mathbf{a}$,

(monotonicity) $N(\mathbf{a}) \leq_M N(\mathbf{c})$ if $\mathbf{a} \leq_M \mathbf{c}$.

In functional notation, N: $([0..1],[0..1]) \rightarrow ([0..1],[0..1])$.

Given the orderings relation $\leq_M$, the definitions of the T-norm and S-norm are straight extensions of the one-dimensional definitions. Concerning the N-norm, one important change has been made, the *monotonicity* condition has changed from strictly decreasing to non-decreasing. The reason for this change is that belief supporting and refuting a piece of evidence is no longer represented by just one number. Therefore, an N-norm like $N(a)=1-a$, as specified in most of the one-dimensional models, is no longer the most suitable candidate in our model of belief and disbelief. When we have specified the N-norm for the model of belief and disbelief, we will show that De Morgan's law is still valid for pairs of T- and S-norms.

4.5.4 Derivation and independence

One of the characteristics of the set of propagation functions for the probabilistic model is the presence of the check on the dependence of propositions to be combined. In the possibilistic approach, we do not want to ignore this check, instead we search for an alternative in the context of the model of belief and disbelief. It is a plausible thought that the resulting belief of a combination of two independent propositions differs from the combination of two dependent propositions. In the probabilistic model, this notion of independence was captured by the conditions of mutually and conditionally independence of pieces of evidence. Leaving the probabilistic foundation therefore results in the necessity of a new definition of independence. We provide the following definition.

Definition 4.11

Two derivations $d_1,d_2 \in D$ are considered to be *causal independent* when $d_1 \cap d_2 = \emptyset$.

The meaning of Definition 4.11 is clear, two derivations are causal independent when they can be constructed without using a single piece of evidence in both derivations. We use this notion of causal independence in the definition of the propagation functions of the model of belief and disbelief, given in the next subsection.

4.5.5 The propagation functions

According to the observations made in the previous subsection, we formulate the propagation functions of our model of belief and disbelief in terms of propositions, expressions and derivations. We use the propagation functions of the probabilistic model as the starting point and explain the choice for their possibilistic counterparts in the model of belief and disbelief.

The propagation function for the negation, as given in Theorem 4.5, is not restricted by any probabilistic condition. Further, it is a plausible function for the negation. Therefore, we also use this function for the negation in the model of belief and disbelief.

Propagation function 4.1
Let $e \in E$, $d_b, d_d \in D$ and $MB(e,d_b), MD(e,d_d)$ be given.
Then the propagation functions for the *negation* are defined by
(a) $MB(\neg e,d_d) = MD(e,d_d)$.
(b) $MD(\neg e,d_b) = MB(e,d_b)$.

This set of propagation functions for the negation fulfils the demands on an N-norm, as given in Definition 4.10. Further, with the definition of this propagation function for the negation, De Morgan's Law on T- and S-norms, $T(\mathbf{a},\mathbf{b}) = N(S(N(\mathbf{a}),N(\mathbf{b})))$, holds for all pairs of T-norms and S-norms as defined in Definitions 4.8 and 4.9. Again, pairs of T- and S-norms can be defined for the propagation in conjunction and disjunction operators.

In Section 3, probabilistic conditions were needed for the definition of propagation functions for the conjunction and disjunction operators in the probabilistic model. By using Definition 4.11, we can reformulate the impact of independence in terms of derivations, and use them to formulate the propagation functions in the possibilistic model. An observation that can be made here is that the propagation functions delivered by the probabilistic model have a form that is well known in the theory on triangular norms and conorms. The functions working on the separate measurements (MB and MD) are the T_2 and S_2 norms[9]. This results in the following propagation functions for the conjunction and disjunction.

Propagation function 4.2
Let $e_1,e_2 \in E$, $d_{b_1},d_{b_2},d_{d_1},d_{d_2} \in D$ and $MB(e_1,d_{b_1}),MD(e_1,d_{d_1}),MB(e_2,d_{b_2}),MD(e_2,d_{d_2})$ be given.
Further, let (d_{b_1},d_{b_2}) as well as (d_{d_1},d_{d_2}) be causal independent.
Then the propagation functions for the *conjunction* are defined by
(a) $MB(e_1 \wedge e_2,d_{b_1} \cup d_{b_2}) = T_2(MB(e_1,d_{b_1}),MB(e_2,d_{b_2}))$.
(b) $MD(e_1 \wedge e_2,d_{d_1} \cup d_{d_2}) = S_2(MD(e_1,d_{d_1}),MD(e_2,d_{d_2}))$.

Propagation function 4.3
Let $e_1,e_2 \in E$, $d_{b_1},d_{b_2},d_{d_1},d_{d_2} \in D$ and $MB(e_1,d_{b_1}),MD(e_1,d_{d_1}),MB(e_2,d_{b_2}),MD(e_2,d_{d_2})$ be given.
Further, let (d_{b_1},d_{b_2}) as well as (d_{d_1},d_{d_2}) be causal independent.
Then the propagation functions for the *disjunction* are defined by
(a) $MB(e_1 \vee e_2,d_{b_1} \cup d_{b_2}) = S_2(MB(e_1,d_{b_1}),MB(e_2,d_{b_2}))$.
(b) $MD(e_1 \vee e_2,d_{d_1} \cup d_{d_2}) = T_2(MD(e_1,d_{d_1}),MD(e_2,d_{d_2}))$.

Since the one-dimensional norms are used on the separate measurements, it can easily be

[9] The T_2, T_3, S_2 and S_3 norms are given in Example 3.7.

checked that these definitions of the propagation functions satisfy the demands of the definitions of the two-dimensional T-norm and S-norm, and De Morgan's Law.

Recalling Chapter 3, in fuzzy set theory the attention has been focused on the development of propagation functions for the implication operator. We discussed that numerous propagation functions have been developed for the implication and gave a few examples. In Theorem 4.3, we specified the propagation functions for the (deductive) implication operator in the probabilistic model. The given probabilistic conditions do not demand equivalents in terms of the derivation. The propagation function used is also well-defined in possibility theory (see Example 3.8). Therefore, we can formulate the propagation functions for the deductive inference step without further conditions on the derivation. According to the definition of the derivation of a proposition, the derivation of the conclusion is equal to that of the premise.

Propagation function 4.4
Let $a \in A$, $e \in E$, $d_b, d_d \in D$ and let $MB(e,d_b), MD(e,d_d)$ be given as well as the inference step $e \Rightarrow a$ with strength $(MB(a,e), MD(a,e))$.
Further, let $MB(e,d_b) > 0$.

Then the propagation functions for *deductive inference* are defined by[10]
(a) $MB(a,d_b) = MB(a,e)MB(e,d_b)$.
(b) $MD(a,d_b) = MD(a,e)MB(e,d_b)$.

In Section 2, probabilistic conditions are specified for the propagation functions of the combination operator. By using Definition 4.11, we reformulated the impact of independence in terms of derivations. This results in the following propagation functions for the combination.

Propagation function 4.5
Let $a_1, a_2 \in A$, $d_{b_1}, d_{b_2}, d_{d_1}, d_{d_2} \in D$ and $MB(a_1,d_{b_1}), MD(a_1,d_{d_1}), MB(a_2,d_{b_2}), MD(a_2,d_{d_2})$ be given.
Further, let (d_{b_1}, d_{b_2}) as well as (d_{d_1}, d_{d_2}) be causal independent.
Then the propagation functions for the *combination* are defined by
(a) $MB(a_1 \| a_2, d_{b_1} \cup d_{b_2}) = S_2(MB(a_1,d_{b_1}), MB(a_2,d_{b_2}))$.
(b) $MD(a_1 \| a_2, d_{d_1} \cup d_{d_2}) = S_2(MD(a_1,d_{d_1}), MD(a_2,d_{d_2}))$.

This concludes the first part on the development of a set of propagation functions for our model of belief and disbelief. However, due to the independence conditions, the propagation functions given in Propagation functions 4.2, 4.3 and 4.5 can still only be applied in specific situations. Therefore, we have to generalise them to make them applicable in each situation.

4.5.6 Enhancements for the propagation functions

The propagation functions of the calculus are still based on the uncertainty measurements of their direct predecessors. In Section 3, it was argued that this principle of locality in the

[10] Note that in both functions the resulting derivation is equal to d_b, derivation d_d is ignored.

propagation process should be reconsidered. The use of information concerning the derivation of a piece of evidence was expected to be of help in creating a better uncertainty propagation.

Observing Propagation functions 4.1 through 4.5, we can see that the functions for both negation (4.1) and implication (4.4) can be applied in any situation. In the case of the other three propagation functions, the demand of causal independence of the derivations must be fulfilled. In Theorem 4.11, we describe how this check can actually be carried out by describing the relation between a derivation and a set of justifications. A question that remains is what we should do in case the demand cannot be fulfilled (i.e. when the derivations are based on common pieces of evidence). We propose a strategy here that is capable of handling this situation and subsumes the propagation functions as given in the previous subsection. Since we treat the three operators equally, we use the * symbol to denote one of the operators $\{\wedge,\vee,\|\}$ and the term d_* to denote the obtained derivation.

First, we observe that in the case of independence of evidence, two general forms for these propagation functions are used, the norms T_2 and S_2. The propagation functions using the T_2 norm are used for the measurement of disbelief of the disjunction and the measurement of belief of the conjunction. Propagation functions using the S_2 norm are used for the measurement of belief of the disjunction, the measurement of disbelief of the conjunction and both measurements of the combination. We introduce a propagation function for both types of function, based on observations concerning their semantics.

In this, we need the notion of a subderivation. From Definition 4.6 we can learn that a derivation of an atomic proposition or expression can be constructed out of several derivations, provided by several implications, or belong to subexpressions. When a part of a derivation can function as a derivation for the proposition or expression on its own, it is called a subderivation. A formal definition is given below.

Definition 4.12

A derivation $d_s \in D$ is called a *subderivation* of a derivation $d_e \in D$ of expression $e \in E$ when $d_s \subseteq d_e$ and d_s is a derivation for e.[11]

Consider the situations in which the measurement of belief or disbelief M[12] of two pieces of evidence $\{x_1,x_2\}$ with derivations $\{d_1,d_2\}$ have to be combined by the *S_2 norm*. From the logical point of view, the presence of just one of the two pieces of evidence is sufficient. This observation leads to the idea that in general we can at least use the maximum of both beliefs as the lower bound of our propagation result. The triangular norm S_3 can be used to represent this propagation function: $S_3(M(x_1,d_1),M(x_2,d_2))$. The resulting derivation d_* is chosen accordingly, yielding either d_1 or d_2. However, we can still use the S_2 norm when we can find derivations d_x and

[11] Note that each derivation is a subderivation of itself.

[12] Since there is no difference in applying the norms on either MB or MD, we use M for short.

d_y that are subderivations of the derivations d_1 and d_2 and that are causal independent, i.e. $d_x \subseteq d_1$, $d_y \subseteq d_2$ and $d_x \cap d_y = \emptyset$. In other words, we are searching for a pair of subderivations that are causal independent and produce a propagation result, by using $S_2(M(x_1,d_x),M(x_2,d_y))$, that is greater than both original measurements. In this case, the resulting derivation d* is equal to $d_x \cup d_y$. We can capture this in the following description of the propagation function.

Propagation function 4.6

Let $x_1,x_2 \in A$, $d_1,d_2,d_x,d_y \in D$ and $M(x_1,d_1),M(x_2,d_2)$ be given.

Then the general propagation function for measurement M under operation * from $\{\wedge,\vee,||\}$ is given by

$$M(x_1{*}x_2,d_*) = \max\{S_3(M(x_1,d_1),M(x_2,d_2)),\max_{d_x,d_y}\{S_2(M(x_1,d_x),M(x_2,d_y))\}\},$$

where the maximum operator is used over all pairs $d_x \subseteq d_1$, $d_y \subseteq d_2$ with $d_x \cap d_y = \emptyset$.

A similar argument can be used in situations in which the T_2 *norm* is used. In these situations, the presence of both pieces of evidence is necessary when the logical counterpart is examined. If the derivations of both pieces of evidence are causal independent, the original propagation function, the T_2 norm, can still be used: $T_2(M(x_1,d_1),M(x_2,d_2))$. Again, the resulting derivation d_* is chosen accordingly, yielding either d_1 or d_2. However, when a (sub)derivation can be found that supports both pieces of evidence, it is our interpretation that the propagation result will be *greater* than the result of the original function. Therefore, the original propagation function provides the lower bound of the propagation function. To achieve the upper bound of this result, we have to search for a subderivation $d_\cap$ of $d_1 \cap d_2$ that supports both pieces of evidence, this is called a common subderivation. The belief in this case can be assumed equal to the minimum of belief in $x_1{*}x_2$ given $d_\cap$, represented by the T_3 norm: $T_3(M(x_1,d_\cap),M(x_2,d_\cap))$. The upper bound can be found by maximising over all common subderivations $d_\cap$ of $d_1 \cap d_2$ and the resulting derivation d_* is of course equal to $d_\cap$.

Propagation function 4.7

Let $x_1,x_2 \in A$, $d_1,d_2,d_\cap \in D$ and $M(x_1,d_1),M(x_2,d_2)$ be given.

Then the general propagation function for measurement M under operation * from $\{\wedge,\vee,||\}$ is given by

$$M(x_1{*}x_2,d_*) = \max\{T_2(M(x_1,d_1),M(x_2,d_2)),\max_{d_\cap}\{T_3(M(x_1,d_\cap),M(x_2,d_\cap))\}\},$$

where the maximum operator is used over all common subderivations $d_\cap \subseteq d_1 \cap d_2$.

By using the results of Propagation functions 4.6 and 4.7 in propagation functions for disjunction, conjunction and combination, we have developed a set of propagation functions for the model of belief and disbelief that are not restricted by dependence conditions. This set of propagation functions not only subsumes the set of propagation functions given in Propagation functions 4.2, 4.3 and 4.5, but also subsumes the set of propagation functions given in Buchanan and Shortliffe's original model (see Section 2). In fact, both Shortliffe and Buchanan's and our initial sets are the extremes of this generalised set.

4.5.7 Further enhancements for the propagation functions

The introduction of the propagation functions defined in Propagation functions 4.6 and 4.7 has solved the problem of restrictive independence conditions. Unfortunately, these functions have also introduced a problem, since the associativity conditions in the T- and S-norm definitions are violated by these functions. This is an unwanted effect of these propagation functions, since the associativity condition is an intuitively appealing condition. In this subsection, we generalise the propagation functions in such a way that the associativity condition is satisfied again.

First, we define generalisations of the T- and S-norms, so that they are able to cope with more than two arguments at the time. We call these generalised norms T^n and S^n, definitions are provided in recursive form.

Definition 4.13

The *generalised triangular norm* T^n is modelled by repeated application of a triangular norm T, thus $T^n\{m_1,...,m_n\} = T(T^{n-1}\{m_1,...,m_{n-1}\},m_n)$ and $T^2\{m_1,m_2\} = T(m_1,m_2)$.

The *generalised triangular conorm* S^n is modelled by repeated application of a triangular conorm S, thus $S^n\{m_1,...,m_n\} = S(S^{n-1}\{m_1,...,m_{n-1}\},m_n)$ and $S^2\{m_1,m_2\} = S(m_1,m_2)$.

In functional notation, T^n,S^n: $\{[0..1]\}_n \rightarrow [0..1]$.

Examples of generalised norms and conorms are given in Example 4.1, these are the generalised norms of the T_2, T_3, S_2 and S_3 norms.

Example 4.1

$T^n{}_2\{m_1,...,m_n\} = \Pi\ (m_i)$, $\quad S^n{}_2\{m_1,...,m_n\} = 1 - \Pi\ (1\text{-}m_i)$,

$T^n{}_3\{m_1,...,m_n\} = \min\ \{m_i\}$, $\quad S^n{}_3\{m_1,...,m_n\} = \max\ \{m_i\}$.

The T^n and S^n norms are used in situations where two or more pieces of evidence $\{x_1,...,x_n\}$ have to be combined by an operator from $\{\wedge,\vee,||\}$, i.e. $x_1\wedge x_2\wedge x_3$ or $x_1||x_2||x_3||x_4$. Each piece of evidence is accompanied by its derivation, $\{d_1,...,d_n\}$. Before describing the generalised propagation functions, based on the T^n and S^n norms, we first introduce the definition of the (in)dependent partition set of a set of derivations $\{d_1,...,d_n\}$.

When constructing Propagation functions 4.6 and 4.7, we searched for a pair of causal independent subderivations (in Propagation function 4.6) and a common dependent subderivation (in Propagation function 4.7). Here, we define similar notions for a *set* of derivations. An independent subset of the set of derivations $\{d_1,...,d_n\}$ is a subset in which each pair of derivations has a pair of subderivations that are causal independent. A dependent subset of a set of derivations $\{d_1,...,d_n\}$ is a subset in which a common subderivation can be found for each derivation.

Definition 4.14
A *dependent subset* J of $\{d_1,...,d_n\}$ is a subset of $\{d_1,...,d_n\}$ in which a common subderivation can be found for each derivation in J.

An *independent subset* J of $\{d_1,...,d_n\}$ is a subset of $\{d_1,...,d_n\}$ in which for each pair of derivations in J a pair of subderivations can be found that are causal independent.

(In)dependent subsets are used to split up the original derivation set $\{d_1,...,d_n\}$. A disjunct set of (in)dependent subsets that completely partitions $\{d_1,...,d_n\}$ is called an (in)dependent partition of $\{d_1,...,d_n\}$.

Definition 4.15
An *(in)dependent partition* I of $\{d_1,...,d_n\}$ is a set of (in)dependent subsets $\{J_1,...,J_k\}$ of $\{d_1,...,d_n\}$ satisfying
(i) $J_1 \cup ... \cup J_k = \{d_1,...,d_n\}$,
(ii) $J_1 \cap ... \cap J_k = \emptyset$.

It is clear that for a given set of derivations $\{d_1,...,d_n\}$, it is possible to construct more than one (in)dependent partition. The set of all possible (in)dependent partitions of $\{d_1,...,d_n\}$ is called the (in)dependent partition set of $\{d_1,...,d_n\}$.

Definition 4.16
The *(in)dependent partition set* P of $\{d_1,...,d_n\}$ is the set of all possible (in)dependent partitions $\{I_1,...,I_p\}$ of $\{d_1,...,d_n\}$.

Having defined the notion of an (in)dependent partition set, we can proceed with the definition of general propagation functions that satisfy the associativity condition. The strategy used in this is to find an (in)dependent partition of $\{d_1,...,d_n\}$ such that the resulting belief for $\{x_1,...,x_n\}$ is maximal.

We start with the generalisation of Propagation function 4.6. The measurements of belief and disbelief of a set $\{x_1,...,x_n\}$ with derivations $\{d_1,...,d_n\}$ are calculated for the situation in which the presence of just one x_i is sufficient to satisfy the situation. Similar to the strategy used in Propagation function 4.6, the maximal belief is found when ***independent*** sources are combined, resulting in the use of the independent partition set P. Thus, we have to maximise the belief over all independent partitions in the independent partition set ($I \subseteq P$).

For each independent partition I, the measurements of belief and disbelief are calculated by using the S^n_3 norm over all independent subsets of that partition ($J \subseteq I$). This can be done while the presence of just one independent subset is sufficient to satisfy this situation.

Similar to the use of the S_2 norm in Propagation 4.6, the $S^n{}_2$ norm is used to calculate the measurement of belief or disbelief for each independent subset J, where $J = \{d_i\}$.

The resulting derivation d_* is equal to the union of the derivations in J, i.e.d_* is $\cup d_i$. This results in the following general propagation function based on S-norms.

Propagation function 4.8
Let $x_1,...,x_n \in A$, $d_1,...,d_n \in D$ and $M(x_1,d_1),...,M(x_n,d_n)$ be given.[13]
Then the general S-norm propagation function for measurement M under operation * from $\{\wedge,\vee,||\}$ is given by

$$M(x_1*...*x_n,d_*) = \max_{I \subseteq P} \left\{ \underset{J \subseteq I}{S^n{}_3} \left\{ \underset{d_i}{S^n{}_2} \left\{ M(x_i,d_i) \right\} \right\} \right\},$$

where P is the independent partition set of $\{d_1,...,d_n\}$.

Recalling Propagation function 4.6, we can see that this is indeed a special case of Propagation function 4.8, when $\{x_1,x_2\}$ is the evidence set with the set of derivations $\{d_1,d_2\}$. Two cases can be distinguished. Firstly, if we can find a partition of independent subsets then $P=\{\{d_1,d_2\}\}$ and the S_2 norm is used in Propagation function 4.6 where the $S^n{}_2$ norm is used in the Propagation function 4.8. Since n=2, these functions are equal. Secondly, if we cannot find a partition of independent subsets then $P=\{\{d_1\},\{d_2\}\}$ and the S_3 norm is used in Propagation function 4.6 where the $S^n{}_3$ norm is used in Propagation function 4.8. In case n=2, these two operations are equal.

A similar strategy can be used with the generalisation of Propagation function 4.7. Here, the measurements of belief and disbelief of set $\{x_1,...,x_n\}$ with derivations $\{d_1,...,d_n\}$ are calculated for the situation in which the presence of each x_i is necessary to satisfy the situation. Similar to the strategy used in Propagation function 4.7, the maximal belief is found when *dependent* sources are combined, resulting in the use of the dependent partition set P. Thus, we have to maximise the belief over all dependent partitions in the dependent partition set ($I \subseteq P$).

For each dependent partition I, the measurements of belief and disbelief are calculated by using the $T^n{}_2$ norm over all dependent subsets of that partition ($J \subseteq I$). This must be done while the presence of each dependent subset is necessary to satisfy this situation.

Similar to the use of the T_3 norm in Propagation 4.7, the $T^n{}_3$ norm is used to calculate the measurement of belief or disbelief for one dependent subset J, where $J = \{d_i\}$.

Again, the obtained derivation d_* is equal to the union of the derivations in J, i.e. $\cup d_i$ yielding the following general propagation function based on T-norms.

[13] Again, we use M for both MB and MD.

Propagation function 4.9

Let $x_1,\ldots,x_n \in A$, $d_1,\ldots,d_n \in D$ and $M(x_1,d_1),\ldots,M(x_n,d_n)$ be given.

Then the general T-norm propagation function for measurement M under operation * from $\{\wedge,\vee,\|\}$ is given by

$$M(x_1*\ldots*x_n,d_*) = \max_{I \subseteq P} \Big\{ \; \underset{J \subseteq I}{T^n_2} \Big\{ \; \underset{d_i}{T^n_3} \Big\{ \; M(x_i,d_i) \; \Big\} \Big\} \Big\},$$

where P is the dependent partition set of $\{d_1,\ldots,d_n\}$.

Recalling Propagation function 4.7, we can see that this is a special case of Propagation function 4.9, when $\{x_1,x_2\}$ is the evidence set with the set of derivations $\{d_1,d_2\}$. Again, two cases can be distinguished. Firstly, if we can find a partition of dependent subsets then $P=\{\{d_1,d_2\}\}$ and the T_3 norm is used in Propagation function 4.7 where the T^n_3 norm is used in Propagation function 4.9. In case n=2, these functions are equal. Secondly, if we cannot find a partition of independent subsets then $P=\{\{d_1\},\{d_2\}\}$and the T_2 norm is used in Propagation function 4.7 where the T^n_2 norm is used in Propagation function 4.9. Again, in case n=2, these two operations are equal.

4.5.8 Solving the first benchmark problem

We return to one of the fundamental problems concerning uncertainty models, the *news report example* [Henrion, 1987], that was described in Chapter 1. We claim that we have found a solution to the problem when our generalised propagation functions are used.

We first observe that actually two problems arise in the news report example, the problem of updating one's belief set when new evidence becomes available and the problem of combining the belief of some proposition based on dependent evidence. The former should be solved by the inference model, the latter problem is solved by using the propagation functions as defined in Propagation functions 4.8 and 4.9.

The news report example illustrates the effect of applying the propagation function for the combination of evidence without checking the condition of causal independence. This results in a belief update for the fact that several thousand people are killed. This update is based on applying the combination function twice, using the three sources of information: the radio station, the tv station and the newspaper. When the generalised propagation functions are used, the derivations of the three sources contain a common cause: the unconfirmed telephone interview. Therefore, the maximum operator is used on the measurements of belief of the three derivations, and the result is a smaller belief in the hypothesis than when calculated with the original propagation functions. In this case, the selection of the maximal derivation path is simple, since there are no other combinations of independent subderivations. The situation can be different when, for instance, the report from the tv station is also based on another source of information.

We demonstrate the enhanced propagation functions with a numerical example. The symbol to denote a deductive inference step (justification) was introduced in Chapter 3. Firstly, we denote the terms we use in the example. The unconfirmed telephone call is denoted by *phone call*, while the receiver of the information is denoted by *receiver*. Three news reports are distinguished: *newspaper*, *radio station* and *tv station*.

Now, consider the first steps in the example.[14] The phone call is believed with absolute certainty (MB=1.0), and, for reasons of clarity, it is assumed that all news agencies have an equal belief in this call (MB=0.8).[15] However, the receiver assigns different beliefs to the different sources of information, newspaper reports are strongly believed (MB=0.9), the tv reports a little less (MB=0.8) and the radio reports are weakly believed (MB=0.7). Weighting the belief in the separate information sources, the receiver has three supporting pieces of evidence: from the newspaper (MB=0.72), from the tv station (MB=0.64) and from the radio station (MB=0.56). Assuming independence of these evidence sources, the resulting belief for the receiver would increase to MB=0.96. When the enhanced propagation functions are used, the resulting belief is only MB=0.72 (the maximum belief of the subderivations, in this case *phone call* → *newspaper* → *receiver*). Figure 4.1 depicts this situation.

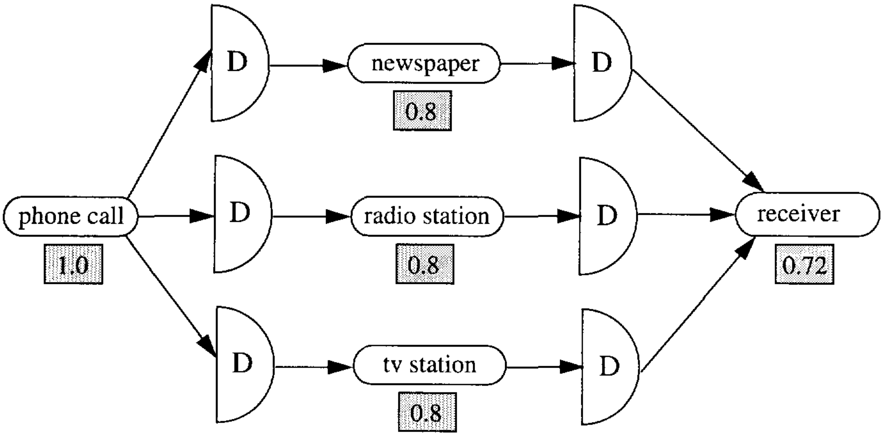

Figure 4.1 First conclusion of the receiver.

Now, suppose the situation is changed by learning that the tv station has received extra information concerning the accident from a local reporter. Following the approach of the first situation, an equal amount of belief is given to this additional report (MB=0.8). However, the belief of the tv station itself increases to MB=0.96 (=0.8+0.8-0.8*0.8) by independently combining the supporting evidence from both the phone call and the local reporter. This results in a change of belief from tv station to receiver into MB=0.77 (0.96*0.8) and the receiver's belief can also

[14] For reasons of clarity, we use the measurement of belief (MB) only in this example.

[15] Skipping the part of first explaining the news reports by the independent sources. We refer to [D'Ambrosio, 1990] for an illustration of such a process.

be updated to MB=0.77 since this is greater than the current MB=0.72. However, there is also a possible combination that leads to an MB=0.90 (0.72+0.64-0.72*0.64). The two independent subderivations that are used here are *phone call → newspaper → receiver* (MB=0.72) and *local report → tv station → receiver* (MB=0.64). Other combinations of subderivations all give results that are less than 0.90. Using Propagation function 4.5 here would result in an MB close to 1.0 (0.99<MB<1.0) for the receiver. This situation is depicted in Figure 4.2.

The news report example is a typical case in which a common cause disturbs an intuitively correct propagation of uncertainty. Our generalised set of propagation functions is designed to cope with all cases of common causes when updating uncertainty of disjunction, conjunction and combination operators.

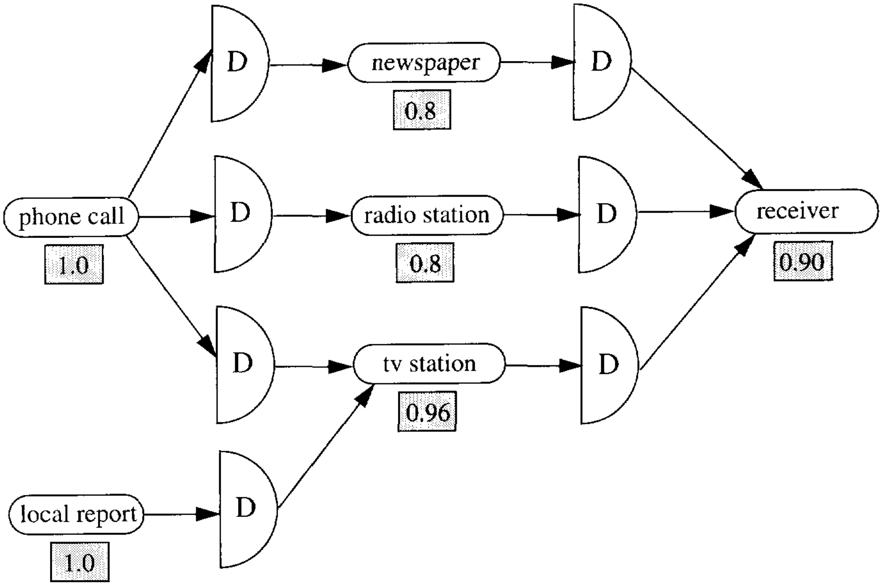

Figure 4.2 Second conclusion of the receiver.

4.5.9 Computational aspects of the enhanced propagation functions

To here, we have not paid much attention to the computational aspects of the propagation functions. Recalling the beginning of the chapter, the computational simplicity of the Shortliffe and Buchanan model is one of the strong arguments in favour of its use. Since the new propagation functions are more general, their computational complexity has to be reanalysed.

Analysing Propagation functions 4.8 and 4.9 (and even 4.6 and 4.7), we must admit that the propagation functions have indeed raised the complexity from originally being linear to being exponential. The reason for this is twofold, it is caused by the search for the (in)dependent partitions and in the inspection of the subderivations. The complexity of Propagation functions 4.6

and 4.7 is O(2^D) and that of Propagation functions 4.8 and 4.9 is O(2^{ND}), where N is the number of pieces of evidence to be combined and D is the average number of subderivations that can be distinguished for the pieces of evidence. Leaving this as it is, we lose one of the advantages of the compositional approach.

However, the complexity can be decreased by not considering the subderivations of propositions (D=1). This decreases the complexity of Propagation functions 4.6 and 4.7 to 2. As a trade-off, the quality of the uncertainty propagation decreases from optimal to suboptimal. For instance, in Propagation function 4.6, only partitions $\{\{d_1,d_2\}\}$ and $\{\{d_1\}\{d_2\}\}$ are considered, and a choice is made between the application of the S_2 and S_3 norm based on the causal independence relation of the derivations d_1 and d_2.

The same procedure can be used to decrease the complexity of the complexity of Propagation functions 4.8 and 4.9. However, the resulting complexity is still exponential in the number of pieces of evidence (N) to be combined, O(2^N). Two additional comments can be made here. By using a smart strategy when generating the partition set, parts of this set can be pruned and the total number of partitions is decreased. However, the construction of the partition set remains an operation of exponential character. Further, it should be noted that N is not the total number of propositions in some application. In fact, N is the number of evidence sources to be combined by just one operation, i.e. the number of propositions used in the premise of an inference step or the number of supporting evidence sources of some proposition. Therefore, it is reasonable to assume that N will be small in real-world applications, and there will be just a small increase in computational complexity of the enhanced propagation functions.

4.6 Embedding the model of belief and disbelief in the general framework

In the foregoing, we defined an uncertainty model based on the measurements of belief and disbelief. In this section, we describe the embedding of this model in the general framework as developed in Chapter 3.

4.6.1 The embedding function

In the general framework, a conflict is defined as the combination of both *true* and *false* labels assigned to an atomic proposition. When introducing uncertainty into the framework, we have to redefine a conflict in terms of uncertainty. We define two situations that are considered to represent a conflict in the uncertainty model that is based on the measurements of belief and disbelief. Firstly, the situation in which positive (MB) and negative (MD) evidence is combined, where at least one of the two measurements is equal to 1. Secondly, the situation in which positive (MB) and negative (MD) evidence that are both using the same derivation (Recall Theorem 4.2) is combined. We need this definition of an uncertainty conflict in constructing a correct embedding of the three-valued model.

Definition 4.17

An *uncertainty conflict* † is defined to be the combination of $MB(a,d_1)$ and $MD(a,d_2)$ for a proposition $a \in A$, with $d_1,d_2 \in D$, when at least one of the following situations occurs

(i) $MB(a,d_1),MD(a,d_2) > 0$ and $MB(a,d_1) = 1$ or $MD(a,d_2) = 1$,

(ii) $MB(a,d_1),MD(a,d_2) > 0$ and $d_1 = d_2$.

Using the notions of a derivation and an uncertainty conflict, in Theorem 4.10 we present a correct embedding of the three-valued labelling L into the uncertainty label (MB,MD), as defined in the context of the general framework.

Theorem 4.10

Consider the *uncertainty labelling* UL based on $T_{ul} = (MB(a,d_b),MD(a,d_d))$, where MB and MD are defined in Definition 4.7.

Further, let the set of *propagation functions* $(f_{\neg},f_{\wedge},f_{\vee},f_{\Rightarrow},f_{||})$ be defined according to Propagation functions 4.1 ($f_{\neg}$), 4.2 ($f_{\wedge}$), 4.3 ($f_{\vee}$), 4.4 ($f_{\Rightarrow}$) and 4.5 ($f_{||}$), and supported by the generalised Propagation functions 4.8 and 4.9.

Then, a *correct embedding* B of the three-valued model into the uncertainty model based on UL and $(f_{\neg},f_{\wedge},f_{\vee},f_{\Rightarrow},f_{||})$ can be defined through

$B(t)=(1,0)$, $B(u)=(0,0)$ and $B(f)=(0,1)$.

Proof

According to the truth tables of $f_{\neg}$, $f_{\wedge}$, $f_{\vee}$ and $f_{||}$ for (MB,MD),[16] see Tables 4.1 through 4.4, and the definition of a correct embedding, as given in Definition 3.31, it can be concluded that the given embedding B is correct.

The following remark must be made here. When *a* is a premise, its derivation d_a is equal to ∅ and the measurements belief and disbelief are given by MB(a,∅) and MD(a,∅). We demand that the values of these measurements are known at the start of the reasoning process and do not have to be calculated by one of the propagation functions. This is exactly the definition of the initial (uncertainty) labelling. In [van der Gaag, 1990] this problem is solved by introducing an extra proposition (u) to the set of atomic propositions A. This method however demands for each uncertainty labelling to obey $MB(u,d_u)=1$, $MD(u,d_u)=0$ and still assumes that there is some environment d_u for which this holds. By explicitly stating that the environment of the initial uncertainty labelling is empty, we gave the initial labelling its place in the embedding.

With the definition of a correct embedding, the boundary constraints of the general framework are satisfied. The last step to come from the general reasoning framework to an uncertainty reasoning framework is the definition of a (deductive) inference function. This means that the inference function, given in Definition 3.18, has to be redefined in terms of the model of belief and disbelief. Examining this definition closely, we can see that this inference function uses the

[16] It should be noted that uncertainty label (MB,MD)=(1,1) that is used in the combination function (see Table 4.5) is an uncertainty conflict (†), according to Definition 4.7.

Table 4.1 Negation

¬	
(1,0)	(0,1)
(0,0)	(0,0)
(0,1)	(1,0)

Table 4.2 Conjunction

∧	(1,0)	(0,0)	(0,1)
(1,0)	(1,0)	(0,0)	(0,1)
(0,0)	(0,0)	(0,0)	(0,1)
(0,1)	(0,1)	(0,1)	(0,1)

Table 4.3 Disjunction

∨	(1,0)	(0,0)	(0,1)
(1,0)	(1,0)	(1,0)	(1,0)
(0,0)	(1,0)	(0,0)	(0,0)
(0,1)	(1,0)	(0,0)	(0,1)

Table 4.4 Combination

||	(1,0)	(0,0)	(0,1)
(1,0)	(1,0)	(1,0)	(1,1)
(0,0)	(1,0)	(0,0)	(0,1)
(0,1)	(1,1)	(0,1)	(0,1)

evaluation function $W_{||}$ and the strength of the inference step $L_{\Rightarrow}$ in the deductive reasoning strategy for the inference step.

We start with the evaluation function W for the premise of a primitive inference step. In the three-valued framework, *true* is demanded for a successful evaluation. In Propagation function 4.4, $MB(e,d_e)$ is used to reflect the belief in the premise expression. In the model of belief and disbelief it is sufficient to demand for this $MB(e,d_e)$ to be greater than 0 instead of using some arbitrary value like 0.2.

The second demand for a primitive inference step to be successful is the fact that no conflict may be derived through it. Therefore, the combination function, $W_{||}(L(a),W_{\Rightarrow}(L(e),l))$ as used in Definition 3.9 is replaced by $f_{||}(UL(a),f_{\Rightarrow}(UL(e),(MB(a,e),MD(a,e))))$. In this, UL(a) is equal to the uncertainty label $(MB(a,d_a),MD(a,d_a))$ of proposition *a* and UL(e) to the uncertainty label $(MB(e,d_e),MD(e,d_e))$ of expression *e*.

Definition 4.18

A *deductive inference function* $I\mathcal{F}$ for the model of belief and disbelief is defined by

$I\mathcal{F}(s) = \mathcal{T}$ when $MB(e,d_e) \geq 0.0$ and $f_{||}(UL(a),f_{\Rightarrow}(UL(e),(MB(a,e),MD(a,e)))) \neq \dagger$,[17]

$I\mathcal{F}(s) = \mathcal{F}$ otherwise.

In functional notation, $I\mathcal{F}$: IS → $\{\mathcal{T},\mathcal{F}\}$.

This concludes the embedding of the three-valued model in the model of belief and disbelief.

[17] Recall that $UL(a)=(MB(a,d_a),MD(a,d_a))$ and $UL(e)=(MB(e,d_e),MD(e,d_e))$.

4.6.2 The relation between derivations and reason maintenance

The notion of a derivation, as defined in Section 5, was defined especially for enabling a correct embedding of the uncertainty model. Here, we clarify the position of a derivation in the general framework. It is not surprising that this position is described in relation to the reason maintenance model.

In the following theorem, it is shown that the derivation d_c of some atomic proposition *c*, can always be determined by *inspecting the subset of justifications that has this atomic proposition in its consequent*. This not only means that the reason maintenance model can be used to determine the contents of derivations but also their *structure*, i.e. the combinations in which, in terms of logical operators, the propositions in the derivation were used.

> <u>Theorem 4.11</u>
> Let A be a set of atomic propositions, D a set of derivations and J a set of justifications. Suppose $c \in A$ and let $d_c \in D$ be the derivation of c.
> Then $d_c = \{ \forall_{j \in J}\ \mathrm{Cons}(j) = \langle c,l_c \rangle \mid \cup_{a \in \mathrm{Ante}(j)} (d_a \cup \{a\}) \}$.
>
> <u>Proof</u>
> The proof consists of two parts.
> (i) Consider some $x \in A$ and assume $x \in d_c \Rightarrow$ {according to Definition 4.6}
> $\exists_{e \Rightarrow x}\ I\mathcal{F}(e \Rightarrow x)=\mathcal{T}$ and $x \in d_e \Rightarrow$
> $\exists_{j \in J}\ j=\{\langle a_i,l_i \rangle\} \rightarrow \langle x,l_x \rangle \Rightarrow$
> $x \in \{a_i\}$ or $\exists_{a_i}\ x \in d_{a_i} \Rightarrow$
> $x \in \{ \forall_{j \in J}\ \mathrm{Cons}(j) = \langle c,l_c \rangle \mid \cup_{a \in \mathrm{Ante}(j)} (d_a \cup \{a\}) \}$.
> (ii) Consider an $x \in A$ and assume $x \in \{ \forall_{j \in J}\ \mathrm{Cons}(j) = \langle c,l_c \rangle \mid \cup_{a \in \mathrm{Ante}(j)} (d_a \cup \{a\}) \} \Rightarrow$
> $\exists_{j \in J}\ j=\{\langle a_i,l_i \rangle\} \rightarrow \langle c,l_c \rangle$ where $x \in \{a_i\}$ or $\exists_{a_i}\ x \in d_{a_i} \Rightarrow$
> $I\mathcal{F}(e \Rightarrow x)=\mathcal{T}$ and $x \in d_e \Rightarrow$
> $x \in d_c$.

In Section 4, it was stated that the second argument of the measurements of belief and disbelief should reflect the total *structured* environment. With Theorem 4.11, it has been shown that the derivation, as given in Definition 4.6, can be used as this structured environment when it is supported by a reason maintenance model as defined in Chapter 3. We used this important result when the structure of derivations was used in the development of propagation functions that are able to react to the situation they are used in. Theorem 4.11 has cleared the semantics of a derivation in terms of reason maintenance, and has thereby made a formal link between the uncertainty model and the reason maintenance model.

4.6.3 The relabelling process

The problem of relabelling was characterised in Chapter 3 and a process model to solve the problem was described. In the setting of this chapter, relabelling is necessary when the uncertainty labelling ($\mathrm{MB}(a,d_a),\mathrm{MD}(a,d_a)$) changes for some proposition *a*. Now the labelling of each expression *e* and each proposition *c* should be changed when proposition *a* is part of the derivation of *e* or *c* respectively ($a \in d_e$ or $a \in d_c$). This situation can be detected by inspecting the derivation of each expression and proposition or, in terms of the reason maintenance model, by inspecting the antecedents of each justification. In Theorem 4.11, the equivalence between derivation and reason maintenance model has been specified. As a result of this theorem, the relabelling process given in Chapter 3 can be applied directly here.

4.7 Discussing the uncertainty model

In this chapter, the general framework for compositional reasoning given in Chapter 3 has been extended toward uncertain reasoning. To start with, an existing model for uncertain reasoning, the certainty factor model of Shortliffe and Buchanan, was analysed on its probabilistic foundation. It turned out that, under severe conditions, a part of the original model of Shortliffe and Buchanan, extended with some new propagation functions, can serve as a probabilistic uncertainty model, let us call this *the revised model*. However, it was argued that the practical impact of the probabilistic conditions in both the original and the revised model prohibits these models to be used in most real-world applications.

Making use of these results, we developed an uncertainty model based on the measurements of belief and disbelief, but without their probabilistic interpretation. New possibilistic definitions for these measurements were introduced, enabling the loss of the severe probabilistic conditions on the propagation functions. A set of propagation functions was developed to be used in this *possibilistic model of belief and disbelief*. These functions are based on both the theory of triangular norms and conorms (negation, conjunction and disjunction operators) and fuzzy set theory (implication operator).

We reformulated the condition on the independence of evidence, as used in the probabilistic models, in terms of our possibilistic model and developed a set of general propagation functions to be used in any situation. This means that, although an alternative dependency condition was formulated, the set of propagation functions of the possibilistic model can be also be applied in situations in which this condition is violated. The resulting model subsumes both the set of propagation functions given by Shortliffe and Buchanan and the set provided by our revised model. Finally, a correct embedding was defined for the possibilistic model of belief and disbelief with respect to the demands specified in Chapter 3. The embedding was completed by the definition of a deductive inference function.

In this last section, we compare the model of belief and disbelief with two existing, comparable approaches, described in [Hájek, 1992] and [Bonissone, 1987].

4.7.1 Compositional uncertain reasoning according to Hájek

As stated in Chapter 1, the term compositional reasoning was due [Hájek, 1992], where another general approach is presented on the management of uncertainty. Differences and equivalencies of both frameworks are briefly analysed here.

Hájek's general framework is also based on the three-valued labelling of *true*, *false* and *unknown*. The reason for this choice is similar to ours, instead of just true and false, there must be the possibility of supplying some neutral label to a proposition. Therefore, the use of at least a three-valued logic is needed. The logical foundation of Hájek's framework is developed upon the algebraic theory of ordered Abelian groups, which gives a set of conditions for the uncertainty propagation functions in a way similar to ours.

Considering the reasoning in Hájek's framework, there is no distinction between expression and inference level, i.e. the inference primitive is part of the expression level[18] and no constructions are available to represent the control of the reasoning process. Furthermore, reason maintenance is not supported by Hájek's framework.

With respect to the representation of uncertainty, Hájek's approach is concerned with models based on probability theory only. However, when the conditions of the algebraic theory are inspected, close relationship with the conditions of the theory of triangular norms and conorms can be shown. This is a strong argument for the idea that possibilistic models can be supported equally well by Hájek's approach.

Regarding the management of uncertainty, Hájek considers existing models only. The certainty factor model and the subjective Bayesian method are proved to be natural extensions of the framework, and criticised from a probabilistic viewpoint. The Dempster-Shafer theory is given more attention, since it is considered a fundamentally better method for a probabilistic uncertainty model. However, the main difference between Hájek's and our approach is the lack of new developments in the propagation of uncertainty. Where our approach aims at providing better uncertainty propagation, which yields a solution for a problem of existing uncertainty models, Hájek's approach has been developed as a general framework for a number of existing uncertainty models only.

4.7.2 Reasoning with uncertainty model of Bonissone

In [Bonissone, 1987], the reasoning under uncertainty model (RUM) is introduced. Similar to our approach, RUM distinguishes an inference and control layer, comparable to our expression, inference and control level. However, there is no logical foundation in RUM that is comparable to the three-valued logic in Hájek's and our approaches. The boundary conditions of uncertainty propagation functions are defined by the definitions of triangular norms and conorms.

The uncertainty model developed in [Bonissone, 1987] is based on an interval measurement, [L(A),U(A)], where L(A) is the lower bound of belief in A and U(A) is the upper bound. Similar to our model, a distinction is made between evidence confirming as well as refuting a proposition, the latter is represented by $[L(\overline{A}),U(\overline{A})]$. However, these measurements are related by

[18] Only the negation and conjunction operators are supported, the disjunction however is omitted.

$L(\overline{A})=1-U(A)$ and $U(\overline{A})=1-L(A)$, as in the Dempster-Shafer theory, and therefore reflect just one measurement. In our model, the measurements of belief and disbelief are defined completely independently, the only condition on them is defined in the uncertainty conflict.

Further, it is argued that the inference layer should have several uncertainty calculi at its disposal. The choice of a specific calculus, however, has to be specified in the control layer, where only global knowledge of the calculi and user-preferences concerning the semantics of the norms and conorms can be used. However, the choice of an uncertainty calculus can only be made once [Bonissone, 1987, p. 252]:

> "Once a calculus has been selected, the combining rules for every value combination are uniquely determined."

This results in the use of a fixed set of propagation functions during reasoning. Contrary to this, our approach is based on a set of generalised propagation functions (using Propagation functions 4.8 and 4.9) that are implemented at the inference level. The determination of the appropriate propagation function is not only based on the characteristics of the calculi but also on the actual situation of the reasoning process in which it is used. This is not just a more flexible approach, it also enables the decision concerning the propagation function to be taken at the right moment: when the propositions and their derivations, as used in the propagation function, are known.

Chapter 5

Explanatory Reasoning

To here, we have only used a deductive inference strategy for the primitive inference step in the general framework. Because of the nature of deductive reasoning (its truth-preserving character) its interpretation is straightforward. As explained in Chapter 2, abductive reasoning is more complex, in the sense that explanations are assumed that can be violated later on, without violating the reasoning process as such. This characteristic is even more interesting when uncertainty is introduced. A set of explanations is assumed with an initial uncertainty distribution and, according to the results of additional reasoning steps, this initial distribution is updated. In this chapter is investigated whether such a reasoning principle can be realised in the compositional architecture as developed in the preceding chapters.

According to the introduction in Chapter 2, abduction is concerned with an interpretation of a causal relation from consequent to cause. Such a definition of abduction is, among others, given by Pople [1973]: *the generation of hypotheses which, if true, would explain some collection of observed facts.* More recent approaches include the notion of selecting the best or most plausible explanation for the observations, see for instance [Peng, 1990]: *inferring the most plausible explanation(s) for a given set of observations.* Today, the latter description is used to describe the general abduction problem.

In this chapter, we introduce an inference strategy for the primitive inference step of the framework that can be used in abductive problem solving in a compositional setting. To avoid confusion, we use the term *explanatory reasoning* to describe the reasoning strategy developed in this chapter. Our, informal, definition of explanatory reasoning is given by *the compositional construction of explanations for a collection of observed facts.* When compared to the given description of abductive reasoning, explanatory reasoning primarily distinguishes itself by explicitly mentioning the compositional architecture. The capability of explanatory reasoning for abductive problem solving is discussed in the closing section. A part of the theory developed in this chapter has also been described in [Goedhart, 1994].

In Section 1, we start with the description of the types of explanations that are supported by the framework and the formal description of the explanatory inference of primitive inference steps. In Section 2, some semantic considerations are outlined with respect to explanatory reasoning. A set of propagation functions is developed for explanatory reasoning in the model of belief and disbelief given in Section 3. Restrictions and limitations of explanatory reasoning are analysed in Section 4. Finally, the capability of explanatory reasoning for abductive problem solving and its relation with other formalisms are discussed in Section 5.

5.1 Defining explanatory reasoning

As discussed in Chapter 2, we use a form of abduction in which relations between explanations and observations are known beforehand. These relations are represented by the primitive inference step of the framework, as described in Chapter 3. In this section, we first describe two types of explanations that are considered meaningful in the context of explanatory reasoning. The definition of an explanatory reasoning strategy for the primitive inference step reflects these explanation types. The impact of this strategy on the theory developed for the general framework is analysed.

5.1.1 Types of explanations

First, we take a closer look at the objective of introducing an explanatory reasoning strategy for the primitive inference step. We consider explanatory reasoning as a form of compositional reasoning in which a set of propositions, which we call the *explanation*, can be assumed to have caused some actual fact.[1] However, we need to be more specific to the relation *between* the propositions within an explanation. In the following, we distinguish two types of explanations: *converging* and *diverging* explanations.

In a *converging explanation*, the assumed propositions in the explanation must all be valid *at the same time* to explain the given proposition. The fact that a combination of cat allergy and the presence of a cat is the explanation of the observation that someone is sneezing is an example of a converging explanation.

In a *diverging explanation*, the validness of *at least one* proposition in the explanation suffices to explain the given proposition. The fact that either a dead battery or an empty fuel tank is considered the explanation of the observation of an engine unwilling to start is an example of a diverging explanation.

Combinations of converging and diverging type explanations can be constructed by the nesting of explanations, i.e. a proposition used in a converging explanation can itself be explained by a diverging explanation and so on. With these two types of explanations, we have described the semantics of explanatory reasoning on the level of primitive inference steps.

5.1.2 Explanatory inference

Recalling Chapter 3, the primitive inference step is a causal relation ($\Rightarrow$), with an expression (e) as its premise and a proposition (a) as its conclusion. The deductive strategy for this inference step is given in the definition of the deductive inference function in Chapter 3. This strategy reasons from premise to conclusion, i.e. in the direction of the causal relation. The explanatory strategy of this primitive inference step is concerned with reasoning in the opposite

[1] The term *observation* is omitted here, since propositions within an explanation themselves can give rise to other explanatory reasoning steps. Following the literature, the term observation is used for the initial input of a diagnostic problem, i.e. comparable to the manifestations from [Peng, 1990].

direction. Given the labelling of some proposition used in the conclusion, the premise can be used to explain the labelling of this proposition. In other words, the propositions used in the premise are assumed to have caused the labelling of the proposition in the conclusion. As a result of this reasoning strategy, executing a primitive inference step adds a set of labels for these propositions to the current labelling.

When we consider the types of explanations, converging and diverging, it is clear how these types are represented by the premise expression. A converging explanation is represented by a conjunction of propositions, where a diverging explanation is modelled by a disjunction. This is in line with the semantics of the evaluation functions, $W_\wedge$ and $W_\vee$, for these operators, that have been defined in Definition 3.5. The $W_\neg$ function is used to model the assignment of a negative label to the propositions in the explanation. Obviously, the evaluation function for the combination, $W_{||}$, as given in Definition 3.8, can be used unchanged.

The explanatory inference strategy can be formalised by the definition of an explanatory inference function. This inference function is based on the type of explanation (converging or diverging) represented by the premise of an inference step ($e \Rightarrow a$). Further, an initial labelling ($L^i_\Leftarrow$) of the propositions in the explanation must be provided by the inference step.[2] Again, we demand that the execution of an inference step may not arise a conflict in reasoning (†). Similar to the deductive strategy, this can be done by checking the outcome of the combination function $W_{||}$. The current labels of the propositions in the explanation, $L(a_i)$, have to be combined with their new labels, which are given by $L^i_\Leftarrow$. The type of explanation, converging or diverging, is taken into account by using the universal and existential operators.

As explained in Chapter 3, the result of an inference function $\mathcal{IF}$ is taken from the set $\{\mathcal{T},\mathcal{F}\}$, where $\mathcal{T}$ should be interpreted as *true* and $\mathcal{F}$ as *false*.

Definition 5.1
An *explanatory inference function* $\mathcal{IF}$ is defined as the assignment of a label from $\{\mathcal{T},\mathcal{F}\}$ to each element $s \in IS$, where
$\mathcal{IF}(s) = \mathcal{T}$ when $\exists_i\, W_{||}(L(a_i),L^i_\Leftarrow) \neq †$, otherwise $\mathcal{IF}(s) = \mathcal{F}$(diverging),
$\mathcal{IF}(s) = \mathcal{T}$ when $\forall_i\, W_{||}(L(a_i),L^i_\Leftarrow) \neq †$, otherwise $\mathcal{IF}(s) = \mathcal{F}$(converging).
In functional notation, $\mathcal{IF}\colon IS \rightarrow \{\mathcal{T},\mathcal{F}\}$.

The following example illustrates the working of this explanatory inference function when applied to the example of a converging explanation.

Example 5.1
Consider the primitive inference step for explaining the observation of someone sneezing by a combination of cat allergy and the presence of a cat to be represented by

[2] Contrary to $L_\Rightarrow$, that is used for the labelling in deductive inference, $L^i_\Leftarrow$ is used for the initial explanatory labelling. This does not mean that the direction of causal relation has changed, it is just interpreted in the opposite direction.

cat_allergy ∧ cat_present ⇒ sneezing.

When a current labelling L is given by {<cat_allergy,*u*>,<cat_present,*u*>,<sneezing,*t*>}, the result of the explanatory inference strategy for the inference step transforms labelling L into {<cat_allergy,*t*>,<cat_present,*t*>,<sneezing,*t*>}.

However, when L was given by {<cat_allergy,*u*>,<cat_present,*f*>,<sneezing,*t*>}, the result of the explanatory strategy for the inference step would transform labelling L into {<cat_allergy,*t*>,<cat_present,†>,<sneezing,*t*>}, which is not allowed since the type of the explanation is converging. Consequently, another explanation has to be found by the inference model.

The definitions concerning the domain model of the framework, as discussed in Section 1 of Chapter 3, are not changed. The only thing that has changed is the reasoning strategy for the primitive inference step. Similarly, the description of the inference model of the framework can be used without changes. The only difference between the deductive and explanatory strategy for the primitive inference step is the fact that in the latter case the current labelling is updated with *a set of bindings* (the explanation), where in the case of deduction, the current labelling is updated with *one single binding* (the conclusion) only.

5.1.3 Adapting the reason maintenance model

Since the representation structures that can be used by the reason maintenance model depend on the reasoning type that is used for the primitive inference step, an additional structure is needed to model the explanatory strategy. This new component is called the *assumption*, it is used to record successfully made explanatory inference steps. In the assumption, the antecedent records the proposition to be explained and the consequent the assumed explanation.

Definition 5.2
An *assumption* is a structure of the form $<a,l> \rightarrow \{<a_i,l_i>\}$, where $a,a_i \in A$, $l,l_i \in T_l$.
The tuple <a,l> is called the antecedent (Ante) of the assumption, the set $\{<a_i,l_i>\}$ is called the consequent (Cons).

The contradiction can still be used to model a conflict in reasoning. Similar to the deductive dependency network, we can define an explanatory dependency network.

Definition 5.3
An *explanatory dependency network* consists of the set of atomic propositions A and a set of assumptions U.

In Chapter 3, some additional definitions were given concerning the validity of justifications and the construction of monotonic proofs. These definitions were primarily used in a theorem that proved the correctness of the inference model when deductive strategy is used for the primitive inference step. A similar approach can be used for the explanatory strategy but, because of this strong similarity and lack of new ideas, it is omitted here. The reason maintenance model is used when uncertainty propagation and relabelling are discussed.

5.1.4 The explanatory uncertainty model

In Chapter 3, we extended the three valued labelling L into an uncertainty labelling based on some uncertainty type T_{ul}. Enabling the use of uncertainty throughout the framework, we introduced uncertainty bindings and uncertainty evaluation functions, and concluded with the definition of a general uncertainty model. We have to reconsider this definition in the context of the explanatory reasoning strategy.

In the preceding, we argued that when using the explanatory strategy, the evaluation functions for the logical operators, $W_{\neg}$, $W_{\wedge}$ and $W_{\vee}$, are equally valid. The same remark was made concerning the combination function $W_{||}$. Therefore, the presence of uncertainty equivalents for these evaluation functions ($f_{\neg}$,$f_{\wedge}$,$f_{\vee}$,$f_{||}$) in the definition of an (explanatory) uncertainty model is obvious. Recalling the deductive strategy, the uncertainty model was extended with a fifth propagation function ($f_{\Rightarrow}$), reflecting the impact of uncertainty of the premise onto the uncertainty of the conclusion. A similar propagation function can be used to represent the influence of the uncertainty of an observation (or any other proposition to be explained) onto the assumed explanation.

From the preceding, it is clear that we can still use the definition of an uncertainty model as given in Definition 3.29. When actually constructing an uncertainty model, the specification of these uncertainty propagation functions makes the difference between both deductive and explanatory strategies. Further, the definitions of an embedding and a correct embedding (see Definitions 3.30 and 3.31) can also be used without any changes. The difference in reasoning strategy is represented in the uncertainty model by the specific choice of the propagation functions.

To make the difference in strategies more explicit, we can change the names of three of the propagation functions that actually represent this difference. Therefore, $f_{\wedge}$ is changed into the propagation function for converging explanations f_{conv}, $f_{\vee}$ into the propagation function for a diverging explanation f_{div} and $f_{\Rightarrow}$ into the propagation function for the influence of the uncertainty of the observation $f_{\Leftarrow}$. This results in the following definition for an (explanatory) uncertainty model.

Definition 5.4

An *(explanatory) uncertainty model* UM for the set of atomic propositions is defined by an uncertainty labelling UL, together with a set of five uncertainty propagation functions ($f_{\neg}$,f_{conv},f_{div},$f_{\Leftarrow}$,$f_{||}$) on UL, where $f_{\neg}$: T_{ul}: $\rightarrow T_{ul}$, f_{conv},f_{div},$f_{\Leftarrow}$,$f_{||}$: T_{ul},$T_{ul} \rightarrow T_{ul}$.

Also when using the explanatory strategy, a relabelling process must be activated when the uncertainty label of some proposition changes. The relabelling process acts in a way similar to the approach used in Chapter 3. In Appendix C, the relabelling algorithm is described, again it consists of two steps: a marking step and a relabelling step. The computational complexity of this relabelling process is of O(|A||U|), in which |A| is the number of atomic propositions and

|U| is the number of assumptions used in the dependency network. The relabelling process is demonstrated on an example explanatory dependency network.

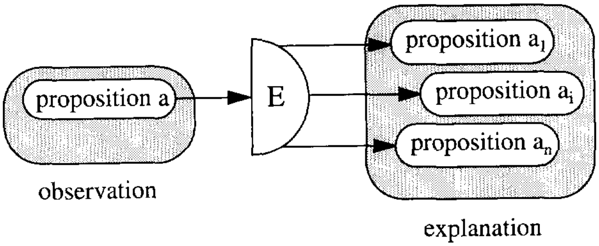

Figure 5.1 Symbol for explanatory inference.

First, we define a symbol for the explanatory reasoning. This symbol is equal to the symbol for a deductive strategy of an inference step, except for the following. Capital D (deductive) has been replaced by capital E (explanatory), with a subscript denoting the number of the explanation when necessary. Obviously, when using the explanatory strategy, the input consists of one proposition (the given fact) and the output of at least one proposition (the explanation). In Figure 5.1, the symbol for the explanatory reasoning strategy is depicted.

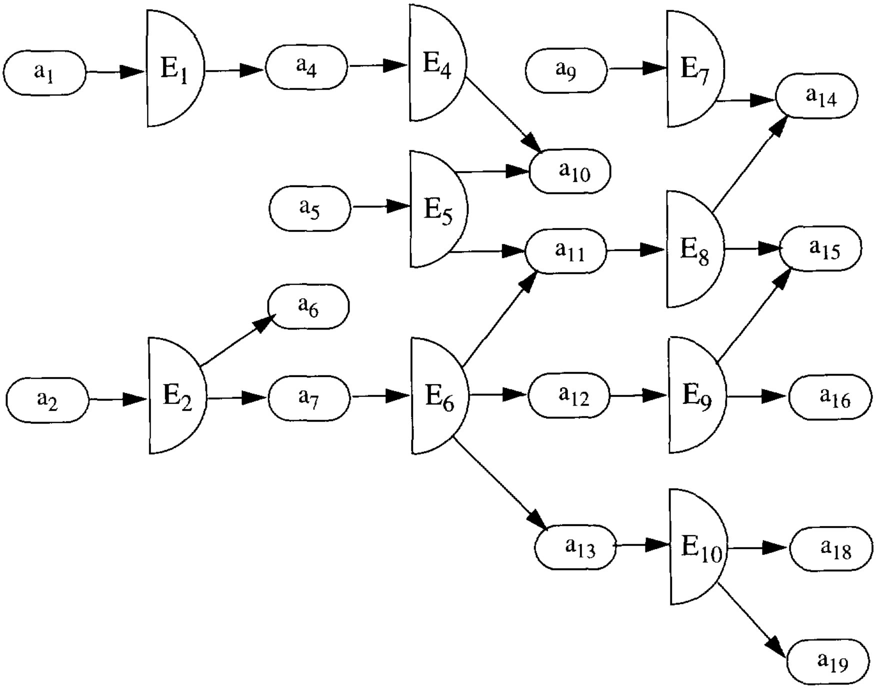

Figure 5.2 An explanatory dependency network.

In Figure 5.2, an example explanatory dependency network is given. A total of 9 inference steps have been executed, in which 16 different propositions are used. The explanation types of the depicted inference steps is not important, i.e. E_5 can be the result of executing $a_{10} \vee a_{11} \Rightarrow a_5$ and E_9 the depiction of $a_{15} \wedge a_{16} \Rightarrow a_{12}$. The impact of executing inference step $a_7 \vee a_8 \Rightarrow a_3$ and adding it to the network in the form of E_3 is demonstrated.

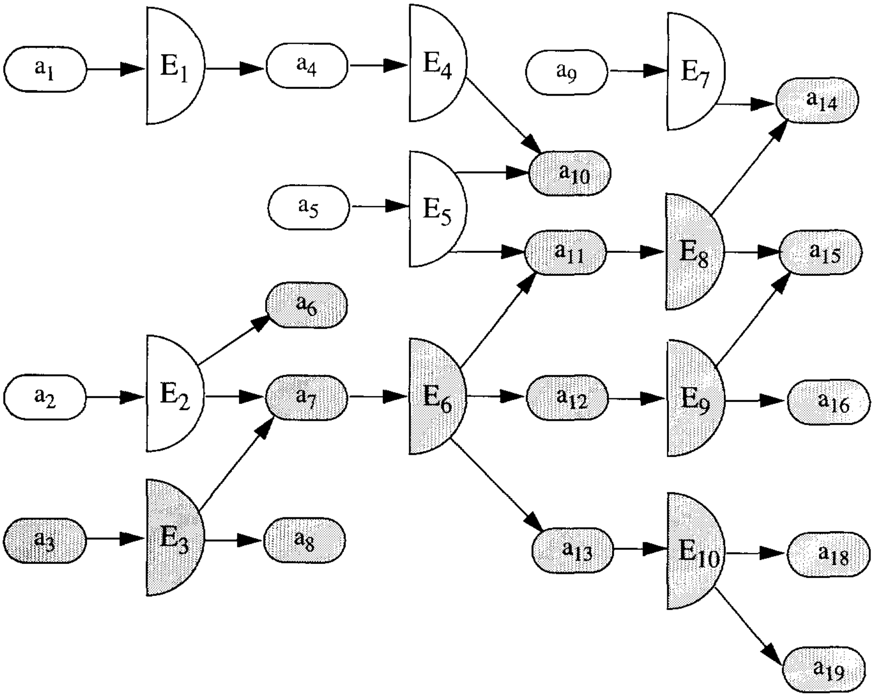

Figure 5.3 Relabelling in an explanatory dependency network.

A relabelling process has to be started as soon as inference step $a_7 \vee a_8 \Rightarrow a_3$ is executed (resulting in E_3), since proposition a_7 is relabelled. The direct consequence of relabelling some proposition is that propositions that are used to explain the proposition also have to be relabelled. Further, the propositions that share an explanation with a relabelled proposition also have to be relabelled, according to the type of the common explanation. In our example, proposition a_7 is relabelled according to the fact that it is used in both E_2 and E_3. As a result of this, propositions a_6 and a_8 are relabelled since they share an explanation with a_7. Propositions a_{11}, a_{12} and a_{13} are relabelled since they are used to explain a_7 through E_6. Since a_{10} is used in the same explanation as a_{11}, it also has to be relabelled, and so on. In Figure 5.3, the propositions and inference steps that are visited by the relabelling process are marked.

5.1.5 Explanatory problem solving

We conclude this section with some remarks concerning the nature of the general framework when the explanatory strategy is used. The definitions of an (uncertainty) program and an (uncertainty) extension, given in Chapter 3, can be used unchanged from a syntactic point of view. Obviously, the semantics of these notions are influenced by the use of the explanatory reasoning strategy for the primitive inference step.

The constraint set of a program, C_0, consisting of primitive inference and control steps, still represents the specific knowledge base used in the application area. Using the explanatory inference strategy, the initial labelling, L_0, represents the collection of observations for which an explanation has to be found. The query Q is used to represent this request in terms of C_0. An extension is now characterised by a labelling L_Q that is the result of an explanatory reasoning process on Q that has been executed by the inference model with positive result: $\mathcal{C}\mathcal{J}(Q)=\mathcal{T}$.

It should be noted that the extension L_Q is a labelling that satisfies the explanatory strategy for the inference steps that have been made to derive this labelling. Comparable to the deductive strategy, more than one extension can be found by the inference model. Again, it is not possible to automatically rank these extensions, since the framework does not include a ranking or ordering mechanism. Using explanatory reasoning for abductive problem solving is discussed in the closing section of the chapter.

5.2 Some semantic considerations on explanatory reasoning

In the preceding section, a formal definition was given of the explanatory reasoning strategy for primitive inference steps. Apart from this formal aspect, a discussion on the semantics of this type of reasoning throws light on the attraction of this strategy. The ability to reconsider previously assumed explanations plays a pivotal role in this discussion.

In the given compositional architecture, two forms of reconsidering previous inference steps can be distinguished. The explanation provided by the inference step is invalidated when one or more propositions in the explanation of the inference step receive additional support. Inference steps that are invalidated (recall Example 5.1) cause the inference model to search for another reasoning path, since the current one leads to inconsistency. This approach is similar to the strategy used on the deductive strategy, the control steps are used to find an alternative reasoning path. The discovery of additional support for some proposition (i.e. the proposition is used in at least two explanations) is a far more interesting situation.

The strength of explanatory reasoning is in the presence of the relation that can be specified *between* the propositions in an explanation. The type of this relation prescribes the changes that have to be made in the labelling of these propositions when additional support is presented. When additional support is found for a proposition, the labelling of the propositions that share a common explanation with this proposition should be reconsidered. The actual changes that are made to the labelling of other propositions in the explanation depend on the type of the explanation, which represents two semantically different relations. When uncertainty or belief measures are used in the labelling, some interesting reasoning characteristics can be supported by explanatory reasoning.

Consider a primitive inference step with a diverging explanation, assuming that the propositions in the explanation can explain the given proposition by themselves. When an increase in belief is observed for one of these propositions, it seems plausible that the belief in other propositions decreases, since it has become more plausible that the proposition that has received additional support explains the given proposition on its own. This behaviour concerning diverging explanations was called *explain away* by Pearl [1988]. Applied to the example of a diverging explanation given in Section 1, this can be seen as decreasing one's belief in the *dead battery* being the explanation of the *engine unwilling to start*, due to an additional observation (for instance by checking the meter) that supports the explanation of the *empty fuel tank.*

Similarly, consider a primitive inference step with a converging explanation, assuming a simultaneous appearance of the propositions in the explanation explains the given proposition. When an increase in belief is observed for one of these propositions, it seems plausible that the belief in the other propositions increases as well. Since, in this case, it has become more plausible that the converging explanation as a whole explains the observation. Being the opposite of explain away, this behaviour is called *joint explain.* Applied to the example of a converging explanation given in Section 1, this is illustrated by increasing the belief one has in the *presence of a cat*, as soon as additional support is given to the explanation of *cat allergy.*

Although the concepts of diverging and converging explanations seem equally important, their role in explanatory reasoning is different. A major characteristic of abductive reasoning is the selection between alternative solutions. This characteristic is represented by a diverging explanation, a converging explanation is used to represent an alternative when it consists of more than one proposition. Therefore, converging explanations are only of importance when they are used in a diverging environment. In the example of a converging explanation, this is illustrated by the existence of some alternative for sneezing, like *having a cold.* This comment is supported by the observation that diverging explanations (explain away) have already been investigated in the literature (for instance [Pearl, 1988]), where converging explanations (and joint explain) have not.

To conclude, the result of explanatory reasoning is the satisfaction of the individual inference steps, according to the (uncertainty) propagation functions that have been defined in the uncertainty model. The framework generates all extensions (labellings) that are internally consistent with respect to the current problem situation (represented by the set of observations) and these explanatory reasoning principles (represented by the explanation types and accompanying uncertainty propagation functions).

5.3 Explanatory reasoning with the model of belief and disbelief

In the preceding sections, we introduced a reasoning strategy for the general framework that enables reasoning towards explanations. The importance of using uncertainty in this reasoning strategy has already been acknowledged and can best be demonstrated by developing an uncertainty model for this strategy. Therefore, in this section we introduce a set of propagation functions for explanatory reasoning in the context of the model of belief and disbelief.

Since the model of belief and disbelief was developed in a deductive setting (see Chapter 4), we have to reanalyse its basic representation measurements (MB and MD). Further, the set of propagation functions is changed from a deductive to an explanatory strategy. From Definition 5.4 we learn that at least three additional propagation functions ($f_{\Leftarrow}$, f_{conv} and f_{div}) must be developed for the model. The result of these changes is demonstrated on the second benchmark example.

5.3.1 The uncertainty measurements

Obviously, we would like to use the same definition for the measurements of belief and disbelief that was introduced in Chapter 4. However, this definition is based on the derivation of an atomic proposition. The notion of a derivation is directly related to the reasoning strategy used (just like the structures in the reason maintenance model). Therefore, a derivation has to be redefined for the explanatory strategy.

When using the explanatory strategy, there are three situations to determine the derivation of an atomic proposition: the proposition is part of the initial labelling, it is part of an explanation or a combination of these situations. When the proposition is labelled by the initial labelling (it is part of the observation set) its derivation has to be empty. When the proposition is part of the explanation (e) of some other proposition (b), through a primitive inference step (e ⇒ b), the derivation is equal to the derivation of that proposition (d_b) extended with the proposition itself. Obviously, in this case the inference step must have been successfully executed by the inference model: $\mathcal{IF}(e \Rightarrow b)=\mathcal{T}$. When the proposition is labelled by a combination of two or more contributions, the derivation is equal to the union of the derivations of the contributors. This yields the following definition of a derivation.

Definition 5.5

A *derivation* d is a subset of the set of atomic propositions A. Derivations are specified for atomic propositions.

A *derivation of an atomic proposition* d_a is defined as

(initial) $d_a = \varnothing$ when a is labelled by the initial labelling (a is an observation),

(explanation) $d_a = d_b \cup \{b\}$ when a is part of the explanation e for b and $\mathcal{IF}(e \Rightarrow b)=\mathcal{T}$,

(combination) $d_a = d_{a_1} \cup d_{a_2}$ when there are two contributions for a.

The set of derivations is called D.

With this changed definition of a derivation, we can use the terms MB(a,d_a) and MD(a,d_a) to denote the measurements of belief and disbelief. Since in explanatory reasoning it is no longer necessary to assign these measurements to expressions, we can slightly change Definition 4.7 of the measurements.

Definition 5.6
The *measurement of belief* MB(a,d_a) and the *measurement of disbelief* MD(a,d_a) are defined as the assignment of a label from [0..1] to each tuple, consisting of an atomic proposition and its derivation.
In functional notation, MB,MD: A,D → [0..1].

5.3.2 Propagation functions for explanatory reasoning

From the definition of an (explanatory) uncertainty model, it can be concluded that at least three additional propagation functions ($f_{\Leftarrow}$, f_{conv} and f_{div}) must be developed for the model of belief and disbelief. We first analyse whether the remaining propagation functions ($f_{\neg}$,$f_{||}$), developed in Chapter 4, can be used in explanatory reasoning.

The $f_{\neg}$ function is used to model the assignment of a negated label to the propositions in the explanation. Due to the unary nature of the operator there is no need for a redistribution of belief in the case of additional evidence, as is necessary for the conjunction and disjunction operators. Further, the semantics of Propagation function 4.1 are also plausible when using the explanatory strategy.

Propagation function 5.1
Let a ∈ A, d_b,d_d ∈ D and MB(a,d_b),MD(a,d_d) be given.
Then the propagation functions for a *negated proposition in an explanation* are defined by
(a) MB(¬a,d_d) = MD(a,d_d).
(b) MD(¬a,d_b) = MB(a,d_b).

In Chapter 4, a set of propagation functions for the combination ($f_{||}$) was given (Propagation function 4.5) and generalised later on (Propagation function 4.8). The semantics underlying these (enhanced) propagation functions for the combination of belief are still attractive in the explanatory strategy. For the moment, we use the equivalent of Propagation function 4.5 in which the separate contributions for an explanation are combined through the S_2 norm. The restriction on the derivations that is introduced by using this propagation function is discussed in Section 4. Increasing generality, the propagation function is defined for an arbitrary number (N) of contributors.

Propagation function 5.2
Let a ∈ A, d_{b_1},...,d_{b_N},d_{d_1},...,d_{d_N} ∈ D and
MB(a,d_{b_1}),...,MB(a,d_{b_N}), MD(a,d_{d_1}),...,MD(a,d_{d_N}) be given.
Let the derivations in the sets {d_{b_i}} and {d_{d_i}} be causal independent.

Then the propagation functions for the *explanatory combination* are defined by
(a) $MB_{||}(a,d_{b_1} \cup ... \cup d_{b_N}) = S^N{}_2(MB(a,d_{b_1}),...,MB(a,d_{b_N}))$.
(b) $MD_{||}(a,d_{d_1} \cup ... \cup d_{d_N}) = S^N{}_2(MD(a,d_{d_1}),...,MD(a,d_{d_N}))$.

Propagation function $f_{\Leftarrow}$ is used to propagate the influence of the uncertainty of an observation (or any other proposition to be explained) onto the propositions in the explanation. Given the primitive inference step ($e \Rightarrow a$), the measurement of belief in the given proposition $MB(a,d_b)$ is used to weight the initial belief of the propositions in the explanation.[3] This initial belief consists of either a positive ($MB(a_i,a)>0$) or a negative ($MD(a_i,a)>0$) belief, for each proposition a_i in the explanation.

Propagation function 5.3
Let $a,a_i \in A$, $d_b,d_d \in D$ and let $MB(a,d_b),MD(a,d_d)$ be given.
Further, let an explanatory strategy for inference step $e \Rightarrow a$ provide initial belief $(MB(a_i,a),MD(a_i,a))$ for each proposition a_i in explanation e.

Then the propagation functions for the propositions in the *explanation* are defined by[4]
(a) $MB(a_i,d_b \cup \{a\}) = MB(a_i,a)MB(a,d_b)$.
(b) $MD(a_i,d_b \cup \{a\}) = MD(a_i,a)MB(a,d_b)$.

The following example illustrates this propagation function when applied to the example of a diverging explanation from Section 1.

Example 5.2
Consider the primitive inference step for explaining the observation of a car that won't start by either a dead battery, not enough fuel in the tank or a motor defect, given by

dead_battery $\vee$ $\neg$fuel_enough $\vee$ motor_defect $\Rightarrow$ car_wont_start.

In this example, suppose the initial labelling of the explanation is given by
MB(dead_battery,car_wont_start) = 0.6,
MB($\neg$fuel_enough,car_wont_start) = 0.5,
MB(motor_defect,car_wont_start) = 0.2.
Consider a labelling, in which dead_battery, fuel_enough and motor_defect are labelled *unknown*:[5]
MB(dead_battery) = MD(dead_battery) = 0,
MB(fuel_enough) = MD(fuel_enough) = 0,
MB(motor_defect) = MD(motor_defect) = 0.
Furthermore, suppose in this labelling MB(car_wont_start) = 0.9.

Then, the labelling is updated with MB(dead_battery) = 0.54, MD(fuel_enough) = 0.9*MB($\neg$fuel_enough,car_wont_start) = 0.45 and MB(motor_defect) = 0.18, according to Propagation function 5.3 (a).

[3] This is similar to the weighting of the premise onto the conclusion in the deductive strategy.
[4] Note that in both functions the resulting derivation is equal to $d_b \cup \{a\}$, derivation d_d is ignored.
[5] For reasons of clarity, we have left out the derivation argument.

As explained earlier, the strength of explanatory reasoning is in the fact that the uncertainty of propositions in an explanation is changed as soon as the uncertainty of one of the explanations changes. The functions f_{conv} and f_{div} must propagate these changes in the uncertain labelling according to their semantics, as described in Section 2. We introduce a set of simple propagation functions that possesses this characteristic. Similar to the propagation functions for the combination, these propagation functions are developed for explanations consisting of an arbitrary number (N) of propositions.

Diverging explanations

In a diverging explanation, the propositions are assumed to be able to explain the given fact (or observation) all by themselves. According to the semantics of explain away, increasing belief (either MB or MD) in one proposition in the explanation should lead to decreasing the belief in the remaining propositions.

In the general case, suppose we have a primitive inference step with N propositions in the explanation, from which K propositions receive additional belief. The resulting belief in these propositions is combined through Propagation function 5.2. The remaining N-K propositions in the explanation have to be relabelled through the f_{div} function.

According to Propagation function 5.2, the propagation functions used for the combination are increasing[6] (i.e. $M_{||}(a_i,d_i) \geq M(a_i,d_i)$ for propositions a_i receiving additional belief). Thus, in the case of a diverging explanation, we are interested in a function f_{div} that decreases the belief in the remaining propositions. We first determine the interval for the belief $M_{div}(a_j,d_j)$ of a remaining proposition a_j. Clearly, the upper bound is given by the original measurement $M(a_j,d_j)$. The lower bound is reached when the increase in belief of the propositions that have additional support is maximal, i.e. $M_{||}(a_i,d_i)=1.0$. We argue that in this case, the belief in a remaining proposition a_j cannot be ignored completely (yielding $M_{div}(a_j,d_j)=0.0$), since there is still a possibility that the proposition has occurred. Therefore, for the lower bound of the interval we choose the initial belief in the simultaneous occurrence of the propositions. We recall that the propositions within a diverging explanation are independent. Using this independency of the propositions, a generalised T_2 norm can be used to calculate the initial belief in the simultaneous occurrence of propositions.

For reasons of simplicity, a linear scale is used to model the ratio between the increasing belief of the propositions with additional support and the decreasing belief of the remaining propositions. The preceding considerations lead to the following uncertainty propagation function f_{div}.

[6] Remember that positive influence is represented by an increasing MB and negative influence by an increasing MD.

Propagation function 5.4

Let $a_1,...,a_K,...,a_N \in A$, $d_1,...,d_N \in D$ and $M(a_1,d_1),...,M(a_N,d_N)$ be given.
Now, suppose $a_1,...,a_K$ receive additional belief, resulting in $M_{||}(a_1,d_1),...,M_{||}(a_K,d_K)$.
Then the relabelling propagation function for a *diverging explanation* is defined by

$$M_{div}(a_j,d_j) = M(a_j,d_j) \prod_{i=1}^{K} \frac{M(a_i,d_i)}{M_{||}(a_i,d_i)} \quad \text{for all } K < j \leq N.$$

The extremes of this, linear, function can be shown to satisfy the semantics we used when developing the function. The upper bound is found when $M_{||}(a_i,d_i) = M(a_i,d_i)$ for each updated proposition a_i, and clearly $M_{div}(a_j,d_j)$ equals $M(a_j,d_j)$. The lower bound is found when $M_{||}(a_i,d_i) = 1.0$ for each of the updated propositions a_i and $M_{div}(a_j,d_j)$ is equal to $M(a_j,d_j)\ \Pi\ (M(a_i,d_i)) = M(a_j,d_j)\ T_2{}^K(M(a_i,d_i)) = T_2{}^{K+1}(M(a_i,d_i))$.

An illustration of this propagation function is given in the following example. Consider the diverging explanation of a car unwilling to start from Example 5.2. For reasons of clarity, we have replaced ¬fuel_enough by proposition empty_fuel_tank.

Example 5.3

The primitive inference step is represented by

dead_battery ∨ empty_fuel_tank ∨ motor_defect ⇒ car_wont_start.

Consider an initial labelling of the explanation, provided by this inference step, given by
MB(dead_battery,car_wont_start) = 0.6,
MB(empty_fuel_tank,car_wont_start) = 0.5,
MB(motor_defect,car_wont_start) = 0.2.
Proposition car_wont_start is an observation and thus MB(car_wont_start)[7] = 1.0[8]. Suppose both dead_battery and empty_fuel_tank receive additional belief from other observations. The combined belief for these propositions is calculated by Propagation function 5.2, suppose $MB_{||}$(dead_battery) = 0.8 and $MB_{||}$(empty_fuel_tank) = 0.6.

MB_{div}(motor_defect) can vary between 0.06 (=0.6*0.5*0.2, the initial belief in the simultaneous occurrence of the three causes) and 0.2 (the initial belief). The belief in the remaining proposition, motor_defect, is calculated through Propagation function 5.4:

$$MB_{div}(\text{motor\_defect}) = 0.2 * \frac{0.6*0.5}{0.8*0.6} = 0.125.$$

Converging explanations

In a converging explanation, the propositions are assumed to explain the given fact by simultaneous occurrence. When using the semantics of joint explain, increasing the belief in one explanation must also increase the belief in the remaining propositions of the explanation.

[7] Since the derivation does not play a role in this example, we leave it out.

[8] Any value between 0.0 and 1.0 for MB(car_wont_start) is handled by Propagation function 5.3.

In the general case, suppose we have a primitive inference step with N propositions in the explanation, where K propositions receive additional belief. The resulting belief of these propositions is again combined through Propagation function 5.2. The remaining N-K propositions in the explanation have to be relabelled through the f_{conv} function.

The propagation functions used for the combination are increasing. In the case of a converging explanation, we are interested in a function f_{conv} that also increases the belief in the remaining propositions. We first determine the interval for the belief $M_{conv}(a_j,d_j)$ of a remaining proposition a_j. This time, the lower bound is given by the original measurement $M(a_j,d_j)$. The upper bound is reached here when the increase in belief of the propositions that have additional support is maximal, i.e. $M_{||}(a_i,d_i)=1.0$. This value of 1.0 can also be used for the maximum of $M_{conv}(a_j,d_j)$.

Again, a linear scale is used to model the ratio between the increasing belief in the propositions with additional support and the increasing belief in the remaining propositions. The preceding considerations lead to the following uncertainty propagation function f_{conv}.

<u>Propagation function 5.5</u>

Let $a_1,\ldots,a_K,\ldots,a_N \in A$, $d_1,\ldots,d_N \in D$ and $M(a_1,d_1),\ldots,M(a_N,d_N)$ be given.
Now, suppose $a_1,\ldots,a_K$ receive additional belief, resulting in $M_{||}(a_1,d_1),\ldots,M_{||}(a_K,d_K)$.
Then the relabelling propagation function for a *converging explanation* is defined by

$$M_{conv}(a_j,d_j) = M(a_j,d_j) + (1-M(a_j,d_j))\frac{\left(\prod_{i=1}^{K} M_{||}(a_i,d_i) - \prod_{i=1}^{K} M(a_i,d_i)\right)}{\left(1-\prod_{i=1}^{K} M(a_i,d_i)\right)} \quad \text{for all } K < j \leq N.$$

This propagation function is illustrated by applying it to the example of a converging explanation from Section 1. The primitive inference step that is used is also used in Example 5.1.

<u>Example 5.4</u>

The primitive inference step is again represented by

cat_allergy $\wedge$ cat_present $\Rightarrow$ sneezing.

Consider an initial labelling of the explanation given by
MB(cat_allergy,sneezing) = 0.9,
MB(cat_present,sneezing) = 0.8,
Again, sneezing is provided by an observation and therefore MB(sneezing) = 1.0.

Suppose proposition cat_allergy receives additional belief because of other observations. The combined belief for this proposition is again calculated by Propagation function 5.2, suppose $MB_{||}$(cat_allergy) = 0.95.

Then, belief in the remaining proposition cat_present is calculated through Propagation function 5.5:

$$MB_{conv}(cat\_present) = 0.8+(1-0.8) * \frac{0.95-0.9}{1-0.9} = 0.9.$$

As argued in Section 2, the use of converging explanations is mainly considered in a diverging environment. In the following example, we create such a diverging environment by extending the explanation with an alternative solution to sneezing by having a cold. Clearly, the belief in this alternative solution decreases when the converging solution of cat allergy and the presence of a cat increases. The increase in belief in the converging solution cat allergy and the presence of a cat is calculated by the relevant part of Propagation function 5.5: $MB_{||}(cat\_allergy \wedge cat\_present)-MB(cat\_allergy \wedge cat\_present)$. The propagation function for the diverging explanation is then used to adapt the belief in having a cold.

Example 5.5

The primitive inference step is again represented by

$(cat\_allergy \wedge cat\_present) \vee has\_cold \Rightarrow sneezing$.

Consider an initial labelling of the explanation given by
MB(cat_allergy,sneezing) = 0.6,
MB(cat_present,sneezing) = 0.5,
MB(has_cold,sneezing) = 0.8,
Again, sneezing is provided by an observation and therefore MB(sneezing) = 1.0.

Suppose proposition cat_allergy receives additional belief because of other observations. The combined belief for this proposition is again calculated by Propagation function 5.2, suppose $MB_{||}(cat\_allergy) = 0.8$.

Then, belief in the remaining proposition (cat_present) in the converging part of the explanation is calculated through Propagation function 5.5:

$$MB_{conv}(cat\_present) = 0.5+(1-0.5) * \frac{0.8-0.6}{1-0.6} = 0.75.$$

The initial and updated belief in the converging explanation $(cat\_allergy \wedge cat\_present)$ are calculated by Propagation function 5.5:

$MB(cat\_allergy \wedge cat\_present) = 0.6*0.5 = 0.3$ and

$MB_{||}(cat\_allergy \wedge cat\_present) = 0.8*0.75 = 0.6$.

Now $MB_{div}(has\_cold)$ is used to calculate the resulting belief in proposition has_cold, through Propagation function 5.4. In this the preceding results are used for the change in belief in the updated solution $(cat\_allergy \wedge cat\_present)$:

$$MB_{div}(has\_cold) = 0.8 * \frac{0.3}{0.6} = 0.4.$$

5.3.3 The second benchmark problem

We now consider the second of the benchmark problems concerning uncertainty propagation, as described in Chapter 1. The propagation functions developed in this section are applied to the *alarm call example* by Kim and Pearl [1983].

As with the first benchmark problem (see Chapter 4), it should be noted that actually two problems arise in the alarm call example. The first problem is again caused by the reasoning process being non-monotonic. As soon as Mr. Holmes receives the telephone call from his neighbour, he assumes that a burglary is than taking place. After hearing the radio newscast, he revises this explanation by stating that *either* a burglary *or* an earthquake can have triggered the alarm. This problem is solved by the inference model of the general framework, which is capable of handling non-monotonic reasoning.

As soon as Mr. Holmes receives the telephone call from his neighbour concerning the burglar alarm sound, he makes an *explanatory* inference step. In this step, a *diverging* explanation (consisting of both burglary and earthquake) is used, assigning strong belief to the burglary and less strong belief to an earthquake. Now, hearing the newscast report concerning the earthquake, additional support is supplied for this to be the explanation for the alarm sound. According to the semantics of the f_{div} function, this decreases the belief in other propositions in a diverging explanation. As a result of this, the framework decreases the belief in a burglary. We demonstrate this behaviour with a numerical example.

Now, consider the first steps in the example.[9] After receiving the phone call from his neighbour, which is believed with absolute certainty: MB(phone_call)=1.0. Mr. Holmes explains this phone call by the alarm sound really going off. Suppose the initial labelling[10] of this explanation is MB(alarm_sound,phone_call)=0.8. Next, Mr. Holmes explains the alarm going off by a diverging explanation consisting of a burglary and an earthquake. The initial labelling of this explanation is given by MB(burglary,alarm_sound)=0.9 and MB(earthquake,alarm_sound)= 0.2. With Propagation function 5.3 (a), the belief in the alarm sound is weighted on these explanations. This results in MB(burglary)=0.72 (=0.8*0.9) and MB(earthquake)=0.16 (=0.8*0.2). Figure 5.4 depicts this situation.

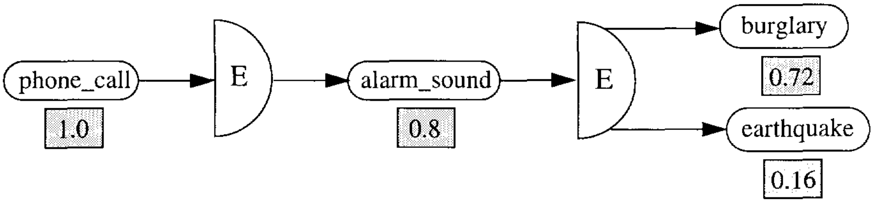

Figure 5.4 First explanation of the phone call.

[9] For reasons of clarity, we use the measurement of belief (MB) only in this example.

[10] Skipping the non-monotonic part of first explaining the alarm sound by the burglary only.

The situation is changed as soon as Mr. Holmes hears the newscast report concerning the earthquake. Since there is no ground for questioning the authenticity of the report itself, this is represented by MB(newscast_report)=1.0. However, knowing the reputation of the broadcast station in question, Mr. Holmes' faith in the contents of the report (earthquake has actually taken place) is given by MB(earthquake,newscast_report)=0.5. The two contributions for the belief in the earthquake are combined by Propagation function 5.2 (a): MB(earthquake)=0.58 (=0.5+ 0.16-0.5*0.16). The belief in a burglary taking place at the same time is decreased to M_{div}(burglary)=0.20 (=0.72*0.16/0.58). This situation is given in Figure 5.5.

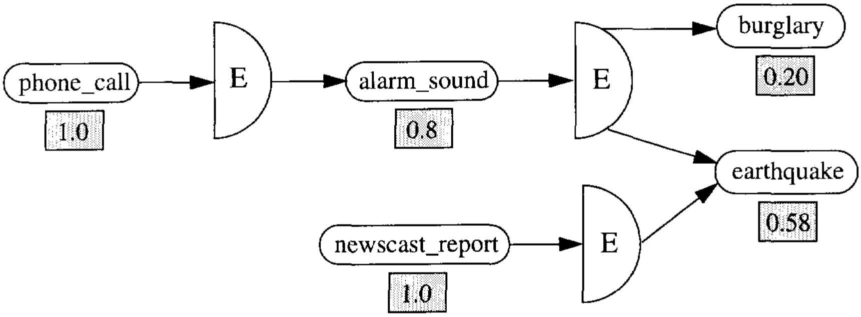

Figure 5.5 Second explanation of the phone call.

The alarm call example is a typical example that cannot be solved when traditional (deductive) compositional reasoning is used. The relation *between* hypotheses, that can only be represented in an additional structure, is the central problem. By introducing the explanatory strategy for the primitive inference step and accompanying semantics, the framework has the possibility to handle these kinds of situations.

5.4 Discussing restrictions and limitations

During the development of the explanatory reasoning strategy, a number of restrictions was imposed. In this section, we analyse these restrictions, separating them into two categories. Some simple restrictions are discussed that are of importance when developing applications with the explanatory reasoning framework. These restrictions are simple in the sense that they do not question the basic concept of explanatory reasoning. Contrary to this, two fundamental restrictions can be distinguished that limit the expressiveness of explanatory reasoning. These limitations are discussed more extensively, and are used in the final discussion on the appropriateness of explanatory reasoning for abductive problem solving.

5.4.1 Some restrictions on explanatory reasoning

During the development of the explanatory reasoning strategy, a number of simple restrictions were encountered that need some additional explication. These restrictions are related to different aspects of the explanatory strategy: the representation of explanations, the model of belief and disbelief employed, and the intended application area.

Allowing multiple explanations for a proposition

There is no limit to the number of explanatory reasoning steps that can be made to explain one proposition. In Figure 5.6, a situation is depicted in which two primitive inference steps have been executed to explain one proposition.

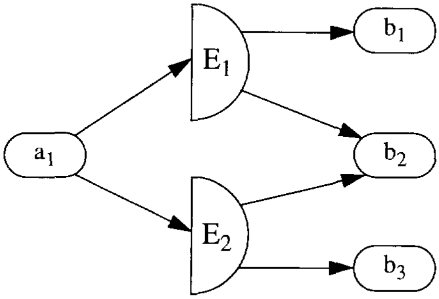

Figure 5.6 Allowing two explanations for one proposition.

In this example, two explanatory inference steps (E_1 and E_2) are used for proposition a_1. These inference steps share a common proposition b_2 in their explanation. Now consider what will happen when explanatory reasoning is used.

Explanatory reasoning uses E_1 to label propositions b_1 and b_2. When E_2 is derived, propositions b_1 and b_3 are relabelled according to the explanation types of E_1 and E_2 respectively and proposition b_2 is relabelled, using the influence of both E_1 and E_2. In this, propagation function 5.2 is used to combine the belief in b_2. This propagation function does not take into account that the same proposition (a_1) has twice activated the explanation in which b_2 is used. Two approaches can be used to solve this problem.

The first is simply to forbid the use of two explanatory reasoning steps for one proposition. This is not too restrictive if one recalls that the types of an explanation can be diverging as well as converging. The notion of a diverging explanation enables the construction of an explanation in which just one proposition suffices to explain the given proposition. This is semantically equivalent to using two separate inference steps to explain the given proposition.

The more general approach is to enhance the propagation function for the combination as proposed in Chapter 4. In Propagation functions 4.6 through 4.9, the derivation argument is used to select the appropriate uncertainty propagation function. A similar approach can be realised here, since the derivation has already been adapted to explanatory reasoning (see Definition 5.5). Such an approach does not restrict the reasoning process and still excludes the unwanted update in belief in this situation.

Independence of propositions in an explanation

In the propagation functions for the converging and diverging explanations, developed for the model of belief and disbelief, it was assumed that the propositions used are independent. As a result of this, the generalised T_2 norm can be used to calculate the belief in the simultaneous occurrence of the propositions. When this assumption cannot be fulfilled, the propagation functions for the converging and diverging explanations have to changed.[11] The result of such a change in the formulation of propagation functions does not interfere with the concept of explanatory reasoning.

Complexity of propagation functions

The computational complexity of the complete uncertainty propagation process is related to both the complexity of the propagation functions and that of the relabelling strategy. In Appendix C, it is shown that the computational complexity of the relabelling strategy is linear in the number of propositions and assumptions in the dependency network, i.e. $O(|A||U|)$. This yields that the total complexity of the uncertainty propagation process is dependent on the complexity of the uncertainty propagation functions.

In analysing the propagation functions in the preceding sections, it is concluded that the only propagation function that can cause exponential computational complexity is the propagation function for the combination. In Propagation function 5.2, a linear propagation function is given for the combination. In Chapter 4, a more advanced version of this propagation function that provides a satisfactorily performance in practical situations was described. Therefore, the computational complexity of the explanatory reasoning strategy is equal to that of the deductive strategy.

The initial uncertainty distributions

When using the model in some application area, the choice of the uncertainty model to be used depends on the ability to acquire the numbers that are used by the initial uncertainty distribution in the explanatory reasoning strategy. For instance, when inference step $a_1 \vee a_2 \vee a_3 \Rightarrow g$ is part of the knowledge base, an initial labelling of a_1, a_2 and a_3, based on the presence of g, must be possible. When a probabilistic uncertainty measurement is used, the acquisition of the initial labelling is in terms of $P(a_1|g)$, $P(a_2|g)$ and $P(a_3|g)$. In practice, these numbers are more difficult to acquire than $P(g|a_1)$, $P(g|a_2)$ and $P(g|a_3)$. When working with possibilistic uncertainty models, as we did with the model of belief and disbelief, the acquisition of these initial distributions for the primitive inference steps must be done by using (experts') estimations.

[11] Recall the enhanced propagation functions in Chapter 4.

5.4.2 Limitations of explanatory reasoning

Two types of restrictions are fundamental to the concept of explanatory reasoning and are proved to be real limitations. We describe these limitations using simple examples, and point out their consequences for explanatory reasoning and compositional reasoning in general.

Prohibiting belief propagation between propositions with additional support

A fundamental restriction is put forward by the prohibition on the mutual influence of propositions within an explanation that receives additional support. Clearly, the propagation functions for converging and diverging explanations are defined for the propositions in the explanation that *do not* receive additional support. A simple example is used to illustrate the fundamental problem that is introduced by this restriction.

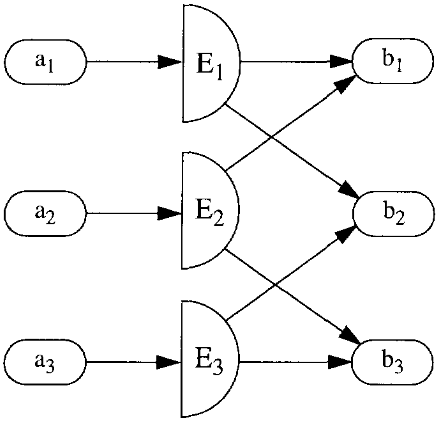

Figure 5.7 Allowing mutual influencing of explanations.

In this example, propositions b_1, b_2 and b_3 are in the explanations of E_1, E_2 and E_3 that are used to explain a_1, a_2 and a_3 respectively. Now, when mutual belief influence is allowed, b_1 and b_2 influence each other through E_1, b_1 and b_3 through E_2 and b_2 and b_3 through E_3. When the explanatory inference steps are interpreted independently, this introduces cycles in belief propagation. For instance, belief in b_1 influences the belief in b_2 through E_1, b_2 influences b_3 through E_3 and b_3 influences b_1 through E_2.

A compositional reasoning environment is just too limited to include the dependencies between propositions that are needed here, i.e. the relation between b_2 and b_3 (given in E_3) is important for the calculation of the belief in b_1, that has no connection with E_3. In fact, in these kinds of situations one would like to use a more global reasoning step in which b_1, b_2 and b_3 are used to explain the propositions a_1, a_2 and a_3. This cannot be accomplished by separately inspecting the individual primitive inference steps.

In the model of belief and disbelief, a restriction is made by prohibiting mutual belief influence between propositions that have additional support. As a result of this, a clear separation is made between the propositions that do and that do not have such additional support. We make this separation explicit by using the terms *supported* and *free* propositions in an explanation. These types of propositions have their own semantics. Supported propositions are labelled according to the supporting explanatory inference steps, regardless of other propositions in the explanation. Free propositions are labelled according to the presence of supported propositions. When observing the final labelling of an explanation, these semantics have to be kept in mind.

The impact of bidirectional influences

In Section 3, it seems that we have found a solution for the second benchmark problem, within explanatory reasoning. However, a critical remark can be made here that results in a limitation of the applicability of explanatory reasoning.

Recalling the numerical example, we introduce an extra proposition (the practical joke) that was not part of the original problem description. This proposition will help us to visualise the limited capability of explanatory reasoning. According to the situation in Figure 5.4, the explanatory dependency network is adapted to this situation, MB(practical_ joke, phone_call) =0.1 is assumed for the initial labelling.

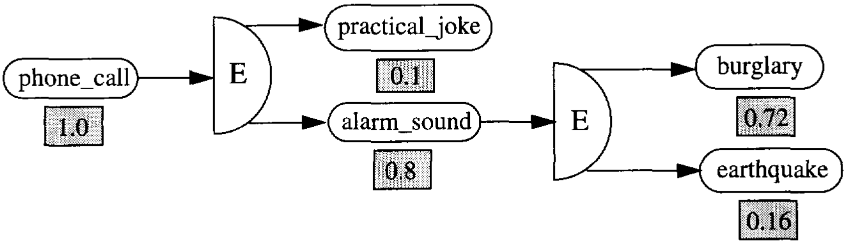

Figure 5.8 Adapted explanatory dependency network.

When we turn to the explanation of the phone call as given in Figure 5.8, we can see that there is no reason to change the initial belief distribution of the phone call. When we think of this a little longer, it can be considered counterintuitive: The alarm going off seems to be more plausible now, and the explanation of the neighbour's joke seems to have lost some of its belief because of the additional information provided by the newscast report. However, the measurements of belief in the explanation of the phone call have not changed (see Figure 5.9).

Viewing this problem in general, it is acknowledged that additional belief in (a part of) the explanation should not only influence the remaining propositions of the explanation, but also the fact that is explained by it. On first sight this can be solved by extending the propagation functions for diverging and converging explanations. However, this leads to belief propagation in opposite directions (bidirectional reasoning), which is beyond the capability of compositional reasoning.

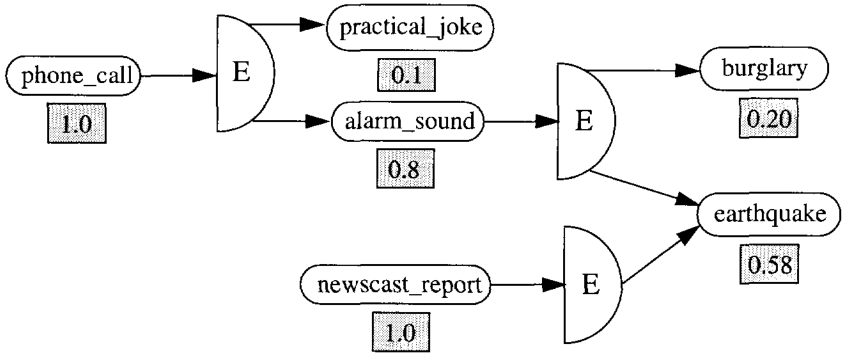

Figure 5.9 Second explanation of the phone call.

One could argue that a solution can be found within compositional reasoning by changing the primitive inference step into e ⇔ a and by merging the semantics of deductive and explanatory strategies. Although this can easily be realised, it is not the main part of the problem. The actual problem is in the fact that the compositional reasoning principle is violated by allowing a primitive reasoning step to be used in the interpretation of itself. In the example, the alarm call is used to generate the explanation consisting of burglary and earthquake. After receiving additional support for the earthquake, this is used to update the alarm call again. As a consequence, it is no longer possible to determine the belief in alarm call incrementally.

To conclude, if bidirectional influences are tolerated, the incremental basis of compositional reasoning must be sacrificed. It is no longer possible to determine the belief of propositions on local information only, since all propositions in a connected dependency network can influence each other through these bidirectional relations. This kind of reasoning process demands the use of a fundamentally different propagation formalism, see Section 5 for a further discussion.

5.5 Concluding remarks

At the end of Chapter 3 we discussed whether the design constraints, as specified in Chapter 2, were satisfied by the general framework. The discussion on the first design constraint (*1. The inference model should be able to support deductive as well as abductive reasoning*) was postponed until this chapter. Because of the introduction of the explanatory strategy for the primitive inference step one could argue that the first design constraint is now satisfied. However, it must be admitted that explanatory reasoning does not cover the complete notion of abductive reasoning. Therefore, a comparison is made first with other formalisms for handling abduction. Finally, the profits and limitations of explanatory reasoning are clearly outlined, especially with respect to abductive problem solving.

5.5.1 A comparison with other abductive formalisms

In Chapter 2, three formalisms were briefly described that enable abductive reasoning under uncertainty. It should be noted that all three formalisms use a probabilistic uncertainty measurement. However, the *type* of uncertainty is not the issue here. How these formalisms treat the problem of abduction when compared to explanatory reasoning is more important.

The parsimonious covering theory

The parsimonious covering theory of Peng and Reggia [1990] was developed from a diagnostic problem-solving standpoint. The model reasons from consequences (called manifestations) to causes (called disorders). The model supports an abductive reasoning process in which all possible disorder sets (called cover sets) that explain a set of manifestations can be generated. When these cover sets are ordered according to a probabilistic measurement based on the strengths of the causal links between disorders and manifestations, a best solution can be selected. A branch and bound type algorithm is used to optimise the search process. When doing this, only a subset of the total number of cover sets is generated. This approach clearly distinguishes between the generation of cover sets and their ordering.

Obviously, the generation of cover sets can be modelled in the general framework by an explanatory reasoning process. In fact, two approaches can be used. The first approach is a straight copy of Peng and Reggia's method, the second approach uses the expressiveness of explanatory reasoning to model the generation of cover sets. We illustrate both approaches, using the example of a simple diagnostic problem, given in [Peng, 1990, p. 106], which is given in Figure 5.10.

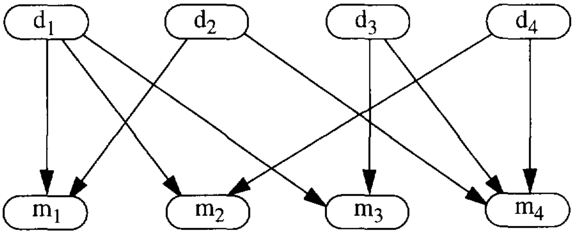

Figure 5.10 A simple diagnostic problem [Peng, 1990, p. 106].

A straight copy of the method of Peng and Reggia can be implemented by transforming the causal links between disorders and manifestations into primitive inference steps. For instance, causal link $<d_1,m_2>$ is transformed into primitive inference step $d_1 \Rightarrow m_2$. A program can be defined in terms of the general framework by grouping these inference steps (according to the manifestation) in a control step of type choice. For instance, this choice construct for manifestation m_2 is given by *choose*$\{d_1 \Rightarrow m_2, d_4 \Rightarrow m_2\}$. The generation of cover sets is now equal to the generation of extensions for a program, in which the set of control steps is given by four of these choice constructs and the set of premises is given by the set of manifestations. To be more precise, it can easily be checked that the set of extensions that is generated by the framework's inference model is equal to the set of *relevant* covers.

Using the second approach, one primitive inference step replaces a set of causal links by a common manifestation. For instance, the set of causal links $\{<d_2,m_4>,<d_3,m_4>,<d_4,m_4>\}$ is transformed into primitive inference step $d_2 \vee d_3 \vee d_4 \Rightarrow m_4$. When doing this, only four primitive inference steps are needed to model the simple diagnostic problem (see Figure 5.11). Again, the extension(s) of a program based on these four inference steps and the set of premises consisting of the manifestations reflect the generation of cover sets. However, the main profit of using this modelling approach is in the fact that facilities like explain away and joint explain can be used in the selection of explanations.

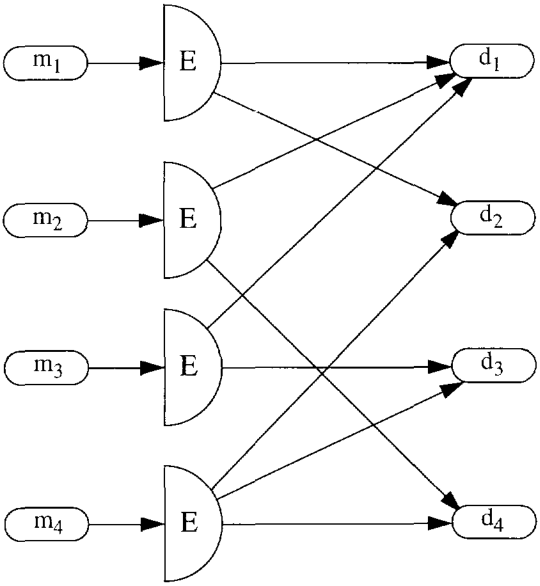

Figure 5.11 Using explanatory reasoning on the simple diagnostic problem.

The second component of the model of Peng and Reggia (the ordering of cover sets) cannot be simulated in the framework as it is. The problem is that no facility is provided by the framework to obtain an ordering of extensions. Defining such a function can be done additionally to the framework (see Chapter 7 for a brief discussion on this subject). Peng and Reggia use the strengths of the causal relations between disorders and manifestations to define such an ordering. A comparable method for ordering extensions can be developed when the first approach for cover set generation is used.

When the second approach is used to model Peng and Reggia's method, the ordering can take advantage of the results of the converging and diverging propagation functions. Belief in disorders that explain only one manifestation is increased or decreased, according to the semantics of explain away and joint explain. Further, disorders are grouped in the explanations, which eases the design of the ordering function.

Apart from the lack of an ordering function for explanations, the general framework has some clear advantages over the model of Peng and Reggia. Firstly, the model of Peng and Reggia does not support an equivalent of the converging explanation, and therefore there is the limitation of using disjoint disorder sets only. Secondly, facilities like explain away and joint explain, which enable plausible human-like reasoning patterns, cannot be expressed in the reasoning part of the model of Peng and Reggia. These patterns are dealt with in the ordering of cover sets, which is less clear. Finally, the model of Peng and Reggia cannot cope with additional evidence, the complete set of manifestations must be supplied beforehand. When additional manifestations are provided, the cover set generation and solution ordering must be restarted from the beginning.

Bayesian belief networks

In Pearl [1988], a distinction is made between intensional and extensional systems for the management of uncertainty.

Extensional systems basically stem from the theory of logic. They treat uncertainty as a generalised truth value that can be attached to atoms and formulas. Formulas are constructed incrementally from subformulas (and eventually atoms) by using operators. These operators have their own, unique, uncertainty propagation function based on the type of the operator and the uncertainty of the subformulas. The functions are used regardless of other information that is available in a specific problem instance.

Intensional systems are based on set theory, where uncertainty is assigned to sets of *possible worlds*. The representation structure of intensional systems is a network of relevant relations between variables. Propagation is not performed by static predefined functions on the operators used to combine sets; instead, a formalism is added that supports a global coherent distribution of uncertainty over the variables, taking all information on a specific problem instance into account.

To be more explicit, extensional systems support the modelling of reasoning primitives, where intensional systems primarily aim at the modelling of structure. Clearly, the compositional reasoning approach fits the description of an extensional system. Contrary to this, a Bayesian belief network is an example of an intensional system. Apart from this fundamental difference, some comparisons can be made.

The notions of explain away and joint explain are supported by both formalisms. The way in which this is realised is characteristic for the difference between intensional and extensional systems. In the compositional approach, the relation that enables explain away and joint explain must be specified *explicitly*. In the belief network approach these features are the *implicit* result of the belief propagation mechanism that has been added to the network structure.

Pearl [1988] places emphasis on the importance of bidirectional causal influences in modelling plausible reasoning. The propagation of uncertainty within Bayesian belief networks (or network models in general) illustrates that this aspect has been uniformly integrated in the concept of intentional systems. Recalling the limitations imposed by bidirectional reasoning in Section 4, we agree with Pearl that in this respect intensional systems clearly outperform their extensional counterparts.

Their simple computational complexity is a strong argument in favour of using extensional systems. Under the restrictions mentioned in Section 4, explanatory reasoning maintains this simple computational complexity of uncertainty propagation. Developing serious applications with Bayesian belief networks demands the use of multiple connected networks. However, the computational complexity of these kinds of networks is exponential by definition, and requires the development of special-purpose algorithms for the management of belief in specific problem instances, see for instance [van de Stadt, 1994].

With respect to abductive problem solving, it is clear that neither the general framework nor the theory on belief networks supports a method to choose between solutions that are provided. In [Neapolitan, 1990] it is acknowledged that such an *ordering of solutions* must be added separately to the representation and propagation aspects of the theory on belief networks. An example of such an ordering is given by Neapolitan [1990] based on the joint probability (conditional on the evidence), maximised over (a subset of) the set of not instantiated variables. Concerning the general framework, we have already acknowledged the lack of such an ordering facility and the way in which it can be added to the framework.

Abductive Horn clauses

A method of extending Horn clause logic programming with an abductive construction is given by Poole [1991; 1993]. The developed language is an extension of Prolog with the disjoint declaration. A disjoint declaration assigns an initial probability distribution to a set of (mutually exclusive) hypotheses, which can be either atomic or predicate symbols in the Horn clause logic. Deductive reasoning on the Horn clauses can be used to discover whether an observation is actually explained by one of the assumables. Contrary to the notion of explanatory reasoning, no attempt is made to interpret the Horn clauses in the way explanatory reasoning does.

In essence, the abductive problem-solving capability of the resulting formalism is comparable to the model of Peng and Reggia. Sets of assumables represent the possible disorders, and deductive Horn clauses are used (compare to the causal links) to discriminate between the assumables. The differences between Poole's method and the general framework are therefore comparable to the differences between the framework and the model of Peng and Reggia.

5.5.2 Profits and limitations of explanatory reasoning for abductive problem solving

The final analysis is given of the explanatory reasoning strategy. The main issue in this analysis is the capability of explanatory reasoning for abductive problem solving.

To start with, it is concluded that explanatory reasoning has attractive reasoning characteristics, like explain away and joint explain, that have not been realised in compositional reasoning thus far. Other advantages of using a compositional reasoning approach, like its well-defined inference pattern and ease of domain model construction, are still present. Another important characteristic is the fact the computational complexity of the propagation functions, when using the explanatory reasoning strategy, is simple. In short, these are the profits of using the explanatory strategy for the reasoning primitives.

Two limitations of the explanatory reasoning approach were described in Section 4. The first limitation could be omitted by slightly changing the semantics of the explanatory reasoning strategy, the second limitation proved to be fundamental. Bidirectional reasoning, necessary to model some aspect of plausible reasoning, was argued to violate the basics of compositional reasoning by disturbing the incremental form of belief updating. Since this is an indispensable characteristic of compositional reasoning, it yields the final conclusion that compositional reasoning is too limited to enable general abductive problem solving.

This does not mean that we have to completely abandon the idea of abductive reasoning in compositional systems. When, in a specific area of application, the restrictions and limitations formulated in Section 4 do not apply, a compositional architecture can be used in combination with the explanatory reasoning strategy. The success of deductive compositional systems has demonstrated the possibilities of the practical use of such restricted problem-solving methods. We briefly discuss two types of problems that can be solved.

In the previous first subsection, it is discussed how explanatory reasoning can be used in abductive problem solving, as described in [Peng, 1990]. The reason for this is the fact that there is only one single layer between disorders and manifestations. The limitation of not being able to model bidirectional influences cannot occur in such problem descriptions. Therefore, if a problem can be formulated in this restricted form, explanatory reasoning can be used in solving these problems.

These one layer problem descriptions are quite restricting to the range and complexity of the problems to be solved. However, when a multi-layer model is used, the bidirectional limitation hampers the explanatory formalism. Still, a special category of these multi-layer problems can be solved when one realises that apart from the bidirectional influence, a partially consistent explanatory dependency network is constructed during explanatory reasoning. Now, when one is able to model the problem in such a way that this partial consistent part suffices to provide a consistent labelling of the propositions of interest of a problem, the bidirectional influences can be ignored.

To illustrate this, we (for the last time) use the example of explaining the telephone call. In Figure 5.12, we have extended the explanation of the newscast report by an alternative explanation (some measurement failure of seismic activity). When the solution of the problem is now concentrated on just the belief Mr. Holmes has in either a burglary or an earthquake, a consistent labelling of this part of the network is available.

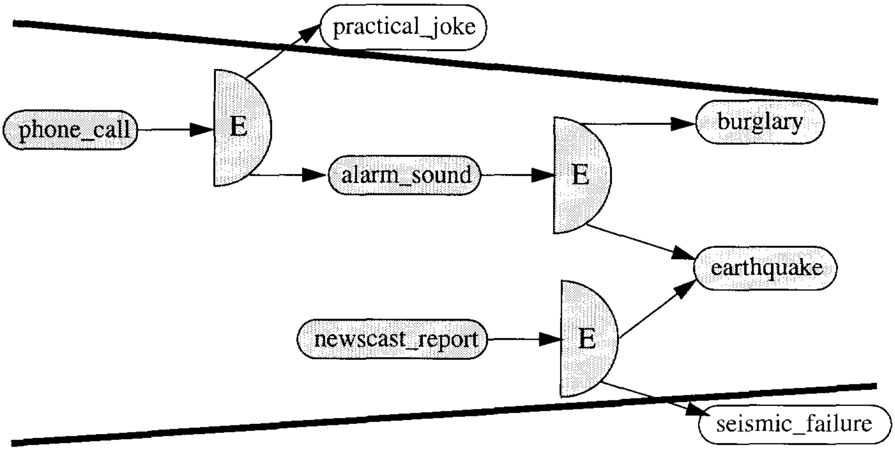

Figure 5.12 Partial explanation of the phone call.

The characteristics of this category of applications are that the propositions one is interested in are used in the explanation part of the primitive inference steps only, and that some sort of *focus of attention* can be used to separate important parts from the rest of the dependency network. This focus of attention includes the given observations and the propositions of interest to the problem solution. Such a set of propositions of interest must be specified by the user of the system. The definition of a program (see Definition 3.12) can be extended for this purpose by including this set. When the labelling of this focus of attention can be done consistently and without using bidirectional influences, meaningful solutions can be found. Obviously, domain model construction is restricted, and even made impossible, by these criteria. However, focusing on the important parts of a reasoning process is very human-like.

In Figure 5.12, such a focus of attention is drawn around the two observations (phone call and newscast report) and includes the two propositions of interest (burglary and earthquake). Both propositions of interest are labelled consistently (see Section 4), and can be used to determine a solution for the observations. When one is interested in the measurements of belief of burglary versus earthquake, meaningful judgements can be made according to this partial solution.

Chapter 6

The Knowledge Engineering Environment Delfi3

In the preceding chapters, research was concentrated on the theory of uncertainty management in compositional reasoning. However, in Chapter 1 we had already acknowledged the importance of using theoretical results in applicable environments, allowing the development of real-world AI applications. In this chapter, we illustrate that an existing environment for knowledge-based system development, Delfi3, can be considered a special case of our general framework. Further, the model of belief and disbelief, as developed in Chapter 4, has been incorporated into the knowledge representation language of Delfi3, which is called LORE.

In Section 1, the ideas behind the design of both Delfi3 and LORE are briefly described. An overview of the domain and inference model of the representation language LORE is given in Section 2. Next, the Delfi3 environment components are described in Section 3, while the applicability of the environment is briefly discussed in Section 4. The mapping of LORE onto the general framework is described in Section 5. Further, the integration of the uncertainty model and the relation with the reason maintenance model are outlined in Sections 6 and 7. The examples used in Section 2 are from a demonstration knowledge base, in Section 4, some examples are provided from real-world applications. The chapter is concluded with some critical remarks in Section 8.

6.1 Introducing Delfi3

In complex domains there is an increasing use of knowledge-based systems. The complexity of the tasks performed by these systems puts high demands on aspects such as their size, reliability and maintenance. The development of knowledge-based systems requires representation and reasoning techniques that are capable of meeting these demands. These techniques are only of practical use, however, when they are supported by efficient environments. The aim of the Delfi project is to study such representation and reasoning techniques and develop such environments [de Swaan Arons, 1991]. Main result of the project is the knowledge engineering environment Delfi3 [Goedhart, 1991].

The knowledge engineering environment Delfi3 is based on the knowledge representation language LORE [Jonker, 1990]. This language offers a uniform integration of frames and logic programming and provides language constructs enabling strong typing, multiple inheritance and modularisation. The Delfi3 environment consists of a compiler, a linker and an interpreter for the language LORE.

6.1.1 Design aspects of LORE

The Delfi3 environment aims to support the development of knowledge-based systems for complex application domains, such as medicine and engineering. Existing systems often use a combination of model-based and heuristic reasoning. Further, these systems tend to be large and are embedded in traditional software environments. During the development of LORE and Delfi3, these aspects where considered to be of great importance.

In LORE, objects and relations support the construction of *domain models*, where objects represent concepts in the application domain and relations represent relationships between these concepts. Objects are frame-like structures, they have the advantage of information grouping (which allows fast inferences). Further, objects can be placed in hierarchies which allow multiple inheritance and the modelling of exceptions. Relations can be defined between an arbitrary number of objects and have user-defined semantics. The combination of objects and relations offers the representation power of structural inheritance networks, as discussed in Chapter 2.

Developing reliable complex knowledge-based systems is not only a matter of testing but also a matter of offering the appropriate language constructs. In LORE, this aspect has received much attention, and led to the introduction of typing and modularisation. The benefit of strong type checking is that a lot of programming errors can be detected at compile-time. The key to large, knowledge-based systems is modularisation. Although found in conventional programming languages, modularisation is rarely used in knowledge representation languages.

The *inference model* of LORE is based on logic programming. The choice for this formalism was supported by its strong theoretical background and the possibilities it offers for the implementation of advanced inference techniques (recall Chapter 2). The main difference between LORE and Prolog expressions is the restriction on relation calls in LORE to contain incomplete data structures. Although this slightly limits the expressiveness of the language, it has the advantage of eliminating the *occur check* when evaluating relation calls, thereby producing a very fast unification and thus providing a high executive performance.

When developing knowledge-based systems the representation of uncertain knowledge is often incorporated. Therefore, the *uncertainty model*, as reported in Chapter 4 was incorporated into LORE. The propagation functions of the uncertainty model require the presence of a derivation argument. This argument is supported by a *reason maintenance model* that has been added to the Delfi3 interpreter.

6.2 The knowledge representation language LORE

The design aspects as described in the previous section led to the specification of the knowledge representation language LORE. We briefly outline the language constructs. For a detailed description of these constructs, we refer to [Jonker, 1990]. Users of the language are referred to [Goedhart, 1992].

6.2.1 Objects

In LORE, a distinction is made between the definition level and the individual level. At both levels objects and relations, which are the basic structures in LORE, can be defined. At the definition level objects represent concepts, while at the individual level objects represent actual items. Relations at the definition level represent possible relationships between concepts (represented by objects at definition level), while relations at the individual level represent actual relations between items (represented by objects at the individual level).

Objects at the definition level can be compared to frames (recall Chapter 2), in that they have attributes that characterise the concept or item they represent. Attributes can be *shared*, which means that they have the same value for each instantiation of the definition object, or *private*, which means that each instantiation can contain a specific value for that attribute. Attributes are further characterised by facets (type, value, default, if-needed and if-added). In the type facet the type of the attribute values is given, the value facet can be used to specify the value of a shared attribute (values of private attributes are specified at the individual level). The default, if-needed and if-added facets can be used when the value of an attribute has to be found at run-time. They contain a logical expression that can be evaluated by the inference model, when the value of an attribute is missing at run-time.

Example 6.1

```
DOBJ Mammal
SHARED
   milk_giving : <BOOL> VAL TRUE
PRIVATE
   has_hooves : <BOOL>
      IFN [ ASK("Does the animal have hooves? ",‘has_hooves) ]
|  chews_cud : <BOOL>
      IFN [ ASK("Does the animal chew cud? ",‘chews_cud) ]
|  ungulate : <BOOL>
      DEF FALSE
|  food : <TEXT # "meat","grass","insects","leaves","fish","others">
      IFN [ ASK("What kind of food does the animal eat? ",‘food) ]
|  color : <TEXT>
      IFN [ ASK("Give the color of the animal : ",‘color) ]
|  stripes : <TEXT # "black", "yellow", "others">
      IFN [ ASK("Describe the stripes of the animal : ",‘stripes) ]
|  identity : <TEXT>
EOBJ
```

In Example 6.1, a definition object *Mammal* is defined that can be used in a simple example of a knowledge base on animal recognition. Since all mammals give milk, the attribute *milk_giving* is shared and can be given value *TRUE*. Private attributes are for instance the *food* the mammal eats or the colour of its *stripes*. If-needed and default facets are given that can be used to find values of attributes (for instance by using the standard predicate *ASK*, that prompts a question to the user of the system). The aim of the knowledge base is to find the value of the attribute *identity*.

Objects at the individual level are linked to the objects at the definition level by means of instance-of connections. A list of values must be supplied to specify the values of private attributes, values of shared attributes can only be given at the definition level. The special symbol *?* is used to denote that the value is *unknown*.

Example 6.2

```
IOBJ
   unknown_mammal   : Mammal(?,?,?,?,?,?,?)
   unknown_ungulate : Mammal(TRUE,TRUE,TRUE,"grass",?,?,?)
EOBJ
```

In Example 6.2, two individual objects are specified for definition object *Mammal*. For the first individual object, *unknown_mammal*, no attribute values are specified. For the second individual object, *unknown_ungulate*, four attributes are given an initial value, because of the fact that the animal is known to be an ungulate from the start.

6.2.2 Relations

Definition relations denote a possible relationship between definition objects; these definition objects are indicated by the arguments of the definition relation. Apart from these arguments, definition relations have a *view*, which is a logical expression in terms of the arguments. The domain model represented through the combination of definition objects and definition relations is a structural inheritance network (see Chapter 2).

Example 6.3

```
DREL Ungulate
DOMAIN a : <Mammal>
[ a.food = "grass"
  OR a.has_hooves AND a.chews_cud
]
EREL

DREL Zebra
DOMAIN a : <Mammal>
[ a.color = "white" AND a.stripes = "black" ]
EREL
```

In Example 6.3, we have presented two definition relations from the animal recognition knowledge base, *Ungulate* and *Zebra*. Both are defined on one argument, of type *Mammal*. Evaluation of these definition relations can be done through evaluating the view, which is a logical expression containing the attributes of domain argument *a*. The actual individual object of type *Mammal* is specified when the definition relation is used. The "." symbol is used to select an attribute from an object.

Instances of definition relations can be specified at the individual level. The object-type arguments must be filled with individual objects. When a definition relation is used, the knowledge base is checked for the presence of individual relations first. If they can be applied in the situation, they are used. Otherwise the view of the definition relation is evaluated.

Example 6.4

```
IREL
   Ungulate(unknown_ungulate)
EREL
```

In Example 6.4, an individual relation is specified for definition relation *Ungulate*. Since we know that the individual object *unknown_ungulate* is an ungulate, this fact can be used directly when definition relation *Ungulate* is used on this individual object. The evaluation of the view of the definition relation is not necessary.

The interaction with other software programs is considered vital for the development of serious applications with knowledge-based systems. External relations are incorporated into LORE to support this interaction. External programs can be activated and a facility is added to exchange information between a knowledge-based system and an external program.

External relations can be compared to definition relations except for the fact that they have no instances at the individual level and the view is replaced by a call statement containing the name of an external program. When the inference model evaluates the external relation, this external program is executed.

Example 6.5

```
XREL MammalDatabase
DOMAIN food,color,stripes : <TEXT>
RANGE identity : <TEXT>
CALL MammalDB
EREL
```

In Example 6.5, an external relation is specified that can be used in the animal recognition knowledge base. Suppose we have a database of mammals that is able to uniquely determine the *identity* of the mammal based on its attributes *food*, *color* and *stripes*. We implement an external program *MammalDB* that supports this query to the database, and use the external relation *MammalDatabase* to run this program during the evaluation of the knowledge base.

Query relations are introduced into LORE, to activate the knowledge base. Query relations are definition relations that have no arguments and no individual relations. The use of query relations is explained in Section 4.

In Example 6.6, a query relation is specified that can be used in the animal recognition knowledge base. The query relation uses the standard predicate *KNOWN* to check the presence of a value for the attribute *identity* of individual object *unknown_mammal*. Since this attribute does not have a value, the if-needed facet is evaluated, which activates the *BRULES* predicate (see Example 6.8). They lead to the evaluation of production rule *IsZebra* (see Example 6.7) and definition relation *Zebra* (see Example 6.3) and so on.

Example 6.6

```
QREL FindIdentity
[  KNOWN(unknown_mammal.identity) AND
     WRITE("Aha, it's a ",unknown_mammal.identity) AND WRITELN(2)
     OR
     WRITE("Sorry, I can't find the identity") AND WRITELN(2)
]
EREL
```

6.2.3 Production rules

Production rules are the most well-known construct to represent heuristic knowledge and are therefore subsumed in the set of LORE language constructs. Production rules are defined at the definition level, just like definition objects and definition relations, and are activated through standard predicates.

Example 6.7

```
RULE IsUngulate
GIVEN a : <Mammal>
IF
[ Ungulate(a) ]
THEN
ABOUT a CONCLUDE ungulate = TRUE
ERUL

RULE IsZebra
GIVEN a : <Mammal>
IF
[ a.ungulate AND Zebra(a) ]
THEN
ABOUT a CONCLUDE identity = "zebra"
ERUL
```

The format of the production rules is trivial, the *given* arguments (that are comparable to the *domain* arguments of relations) specify the argument(s) the rule is defined upon. The premise is specified through a logical expression and the conclusion(s) by explicitly stating object and attribute.

In Example 6.7, two production rules are taken from the animal recognition knowledge base. The *given* argument of both rules is of type Mammal. In the premises of the production rules, definition relations *Ungulate* and *Zebra* are used, these have been defined in Example 6.3. The first rule can be used to conclude whether an animal is an ungulate, the second rule is concerned with the identity of an animal.

Backward and forward reasoning are both supported in LORE, by the standard predicates BRULES and FRULES. When using these special predicates, the values of the given arguments must be specified together with the attribute that is the starting point of the reasoning process. The conflict set, the set of production rules that can be fired in a specific situation, depends on these argument values in the predicate (that must be unified with the given arguments). All production rules in the conflict set created during backward or forward chaining are evaluated consecutively.

Example 6.8

```
DOBJ Mammal
SHARED
  ....
PRIVATE
  ....
| ungulate : <BOOL>
    DEF FALSE
    IFN [ BRULES(SELF[1],ungulate) ]
  ....
| identity : <TEXT>
    IFN [ BRULES(SELF[1],identity) ]
EOBJ
```

In Example 6.8, the standard predicate *BRULES* is used to extend the if-needed facets of attributes *ungulate* and *identity* of definition object *Mammal* from Example 6.1. When such an if-needed facet is evaluated, the *BRULES* predicate determines the conflict set of production rules that have the given attribute in their conclusion. The remaining arguments in the *BRULES* predicate are used to unify with the given arguments of the production rules from the conflict set. For instance, the value of standard predicate *SELF* is unified with the *given* argument *a* in rule *IsZebra*.

6.2.4 Inheritance

In LORE, object hierarchies can be built at the definition level by linking objects by means of super connections. With these super connections, attributes are inherited from objects higher in the object hierarchy. The changes construct in LORE allows overruling the default inheritance by changing facets of inherited attributes. LORE supports multiple inheritance, the only restriction imposed on the object hierarchy is that it should be a directed acyclic graph. The problem with multiple inheritance is how to distinguish attributes with the same name, which are inherited from different objects. Although in the literature (see also Chapter 2) various solutions to this problem have been suggested, these approaches have their drawbacks. Therefore, in LORE conflicts must be solved explicitly by using the inherit construct. This deals with conflicts at the

[1] The standard predicate SELF can be used to refer to the individual object for which the if-needed facet is evaluated.

appropriate level: that of the attributes. In large knowledge bases it is hard to detect conflicts (even after extensive run-time testing). Therefore, conflict detection in the case of multiple inheritance in LORE is carried out at compile-time. A prerequisite is that the object hierarchy does not change at run-time, which is true in the case of LORE. For an exhaustive description of the inheritance mechanism we refer to [Jonker, 1990].

6.2.5 Typing

Apart from the set of predefined types (INT, REAL, BOOL, TEXT, LIST), types in LORE are connected to the names of definition objects. The type set of a LORE knowledge base is a lattice $(T,\leq_t)$, constructed by the set T of all definition object names and a partial ordering $\leq_t$ on T. The partial ordering $\leq_t$ is derived from the transitive closure of the super connection. The ordering of types is crucial, the demand for the type of a value to be compatible with the type of the corresponding attribute results in more flexible type checking than that of conventional languages. The typing rules in LORE are part of the static semantics and can thus be checked at compile-time. However, type checking is also performed at run-time during unification. A variable with type t_V can only be bound to a value with type t_W when the ordering $t_W \leq_t t_V$ is satisfied. The typing mechanism is explained in detail in [Jonker, 1990].

6.2.6 Modularisation

LORE offers modules to support the structured design of large knowledge bases. In LORE, modules are more then just a structure to split up large knowledge bases into manageable parts, they offer data hiding (the export mechanism), explicit interfacing (the import mechanism), re-use of knowledge bases and incremental compilation. With the use of the import and export mechanisms, a hierarchy of modules is built up. The same restriction is placed on this module hierarchy as on object hierarchies, namely that it should be a directed acyclic graph. Another equivalence between object and module hierarchies is the possibility of name conflicts; therefore, no structures with the same name, imported from different modules, may be imported simultaneously. Modules are considered to be units of compilation. An important aspect with respect to re-use of modules is that they may not restrict the possible language constructs. To ensure this, conflict detection and type checking are guaranteed across modules, both at compile-time and at run-time.

6.3 The Delfi3 environment components

The Delfi3 environment consists of three components, a compiler, a linker and an interpreter for the language LORE (see Figure 6.1). The *user* creates a knowledge base consisting of a number of LORE *source* modules by using an *editor*. The source modules are syntactically and semantically checked for errors by the Delfi3 *compiler*. Correct source modules are translated by the compiler into an *object* and an *export* module.

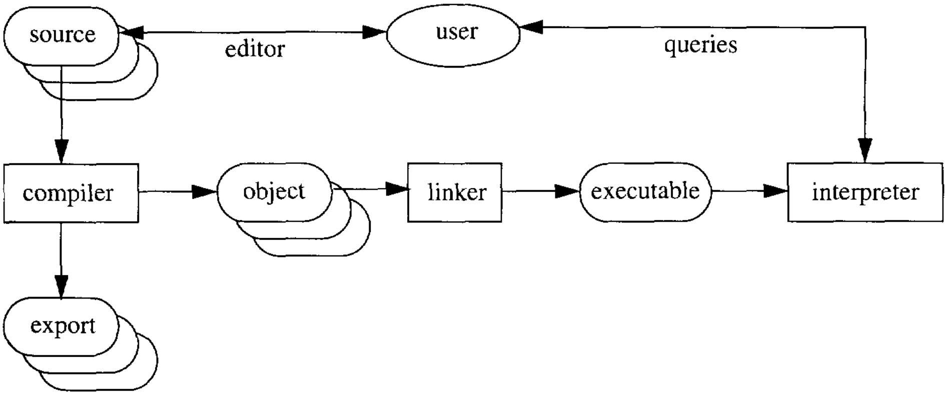

Figure 6.1 The Delfi3 system components.

The object module contains a compact description of the complete contents of the module, the export module contains a compact description of the part of the module that is qualified to export. In the compilation process, it may be necessary to consult the export modules of other modules. The Delfi3 *linker* links the object modules of all relevant modules into an *executable* module, containing the complete contents of the knowledge base. The executable module is loaded by the Delfi3 *interpreter* and can be activated by the user through the *queries*, the calls to query relations.

The Delfi3 compiler and linker are concerned with the checking of the correctness of the used language constructs in the source modules. The execution of a knowledge base is done solely by the Delfi3 interpreter.

6.4 Applicability of the environment

The knowledge engineering environment Delfi3, including the representation language LORE as described in this chapter, enables the development of knowledge-based systems in various domains [Goedhart, 1991]. Examples of such applications are Plexus [Jaspers, 1990; van Daalen, 1993] and Esats [Krijgsman, 1992]. We briefly describe the use of LORE and Delfi3 in the development of two other real-world knowledge-based systems.

Meteodes is a knowledge-based system for the automatic classification of cloud types in Meteosat imagery [van der Lubbe, 1991]. Statistical image processing techniques are not able to solve the associated problems on their own. A major reason for this is that, during classification, the meteorologist uses all sorts of meteorological knowledge concerning the scenes. Meteodes is a hybrid system consisting of a reasoning and an image processing component. A first prototype of the system has already been developed using Delfi3 [Roest, 1992; Schokking, 1992]. Example 6.9 shows an illustration of a production rule used in this prototype. Recently, a formal model of the complete classification process has been developed [Wols, 1993] that will be implemented in Delfi3.

Example 6.9

```
RULE CuContainsCb
GIVEN cs : <Cloud_segment>
IF
[  cs.number_of_pixels < 200
   AND HEAD(cs.clist) = "Cumulus"
]
THEN
ABOUT cs CONCLUDE type = "Cumulonimbus"
ERUL
```

Results from the Delfi project, and Delfi3 in particular, have been used in the field of computer integrated manufacturing [de Swaan Arons, 1991]. The knowledge-based approach has been applied in the on-line scheduling of production processes. In Esprit project 809, *Advanced Control Systems and Concepts in Small Batch Manufacturing*, an expert system was used to handle disturbances in the production process. In short, a knowledge-based system analyses the effects of an unexpected event in the production process. When the effect of some event is expected to be serious, the knowledge-based system advises the generation of a new schedule. The new schedule is based on the current schedule, the situation at hand, the estimated effect of the event on the remainder of the schedule and a set of heuristics to be used in rescheduling.

Example 6.10

```
DREL CheckOverlapping
VAR freelist      : <LIST <Orders>>
      machines  : <LIST <Machine>>
[  GetOperationsAndFreetime(CurrentProblem,'freelist,'machines)
   AND
   ForwardShiftLastOperationPossible(freelist,"Overlapping")
   AND
   DetermineBestOverlap(freelist,machines)
   AND
   ComputeStartDateTime(CurrentProblem,machines)
]
EREL
```

In Esprit project 2415, *Distributed Manufacturing, Planning and Control*, a knowledge-based system has been developed for improving schedules by applying fine-tuning techniques. These techniques exploit idle times in a schedule for overlapping operations or splitting existing orders into smaller batches. An important advantage of using these fine-tuning techniques is the fact that the original schedule is merely disturbed. As a result, the total production time of an order can be shortened, and there is a better overall performance [Jansen, 1993]. In Example 6.10, the definition relation *CheckOverlapping* is given, which is used in the latter application.

6.5 The mapping of LORE onto the general framework

When we want to study the Delfi3 environment as an implementation of the theory developed in Chapters 3 and 4, a mapping is necessary from LORE onto the formalisms of the general framework. This is not a trivial step, since LORE was not developed with the general framework in mind, both were developed independently. Therefore, the mapping is not one-to-one. For instance, LORE does not have a clear distinction between the expression, inference and control levels that are used in the general framework. We create this distinction in LORE as well, based on the semantics of the different language constructs. Developing a mapping means that the LORE constructs have to be defined in terms of propositions, expressions and inference and control steps, which are the basic constructs of the general framework. We first determine the set of atomic propositions in LORE, followed by the mapping of LORE expressions onto the sets of expressions and inference and control steps. Finally, the mapping of LORE knowledge bases onto general framework programs is described.

Mapping LORE variables

In LORE, three sorts of variables are distinguished, the attributes of individual objects and the arguments and local variables that can be defined in relations, production rules and attribute facets. Attributes of individual objects can be used throughout the entire knowledge base and can therefore be considered to be global variables. Arguments of relations and production rules are replaced by global or local variables that are used when the relation or rule is called. Therefore, arguments are nothing more than a local rename of a global or local variable, they cannot be considered explicit variables. Local variables of relations, production rules and the attribute facets can only be used in the constructs they are defined in, and are only of local importance. To conclude, the only LORE variables that are of global interest are the attributes of individual objects and therefore, the set of propositions A for a LORE knowledge base consists of the set of attributes of all individual objects.

In Examples 6.1 and 6.2, we described a definition object, *Mammal*, and two individual objects, *unknown_mammal* and *unknown_ungulate*, resulting in the set of propositions A given by

{ unknown_mammal.milk_giving, unknown_ungulate.milk_giving,
unknown_mammal.has_hooves, unknown_ungulate.has_hooves,
unknown_mammal.ungulate, unknown_ungulate.ungulate,
unknown_mammal.food, unknown_ungulate.food,
unknown_mammal.color, unknown_ungulate.color,
unknown_mammal.identity, unknown_ungulate.identity }.

Further, it must be noted that, although most LORE types allow more than two different values, variables in LORE can have only one specific value *at the time*. The propositions in the general framework do not possess something like a value, therefore the labelling in LORE is based on a pair which consists of a proposition and its current value.

Returning to Examples 6.1 and 6.2, we can give the initial labelling L_0 by

{ <unknown_mammal.milk_giving,TRUE,*t*>, <unknown_ungulate.milk_giving,TRUE,*t*>,
<unknown_mammal.has_hooves,?,*u*>, <unknown_ungulate.has_hooves,TRUE,*t*>,
<unknown_mammal.ungulate,?,*u*>, <unknown_ungulate.ungulate,TRUE,*t*>,
<unknown_mammal.food,?,*u*>, <unknown_ungulate.food,"grass",*t*>,
<unknown_mammal.color,?,*u*>, <unknown_ungulate.color,?,*u*>,
<unknown_mammal.identity,?,*u*>, <unknown_ungulate.identity,?,*u*> }.

Mapping LORE expressions

The sets of expressions, primitive inference steps and control steps are less easy to detect in LORE. This is due to the fact that the set of logical operators provided by LORE (*AND*, *OR*, *NOT*) is used to model the operators in both the set of expressions ($\wedge$, $\vee$, $\neg$) and the set of control steps (*concat*, *choose*, *denial*). We demonstrate the mapping of LORE operators by observing the context in which they are used.

A distinction can be made in LORE between simple expressions and compound expressions. Compound expressions consist of a call to a definition relation, a production rule predicate or they contain attributes that demand the execution of their facets. The remaining expressions are called simple expressions. In Example 6.7, the premise of production rule *IsZebra* consists of a simple expression, *a.ungulate* (when the value of the attribute can be given without evaluating one of its facets), and a compound expression, *Zebra(a)*, connected by the *AND* operator. The notions of simple and compound expressions play an important role in the mapping of the logical operators of LORE.

The *OR* operator in LORE is used to generate backtrack points (alternative reasoning paths in terms of the general framework). The OR operator is thus mapped onto the *choose* operator that is used in the definition of the set of control steps in the general framework. As a result of this, there is no LORE equivalent of the $\vee$ operator that is used in the definition of expressions in the general framework.

The *NOT* operator in LORE is less easy to map onto operators used in the general framework. When the NOT operator is used on simple expressions, it can be considered to model the logical operator $\neg$ that is used in the construction of the set of expressions. When the operator is used on a compound expression, the NOT operator is interpreted by the negation as failure strategy (see Chapter 2) and it can therefore be considered to model the *denial* operator from the set of control steps. Therefore, mapping the NOT operator depends on the context in which it is used.

In mapping the *AND* operator in LORE an approach similar to the mapping of the NOT operator is used. On the one hand the AND operator can be mapped straightforwardly onto the *concat* operator used in the set of control steps, when it is used to combine compound expressions. On the other hand, we can map the AND operator to model the $\wedge$ operator as well, when it is used between simple expressions.

The generation of justifications and contradictions, as proposed in the general framework, is closely related to the primitive inference step. To complete the mapping, we have to make clear what the primitive inference steps are in LORE. Intuitively, structures like definition relations and production rules, are ideal representations of primitive inference steps. However, definition relation views and production rule premises consist of LORE expressions that can contain calls to other definition relations and rule predicates as well (see Example 6.7, definition relation call *Zebra(a)* is used in production rule *IsZebra*). In the general framework this is impossible, i.e. control steps cannot be part of other primitive inference steps. Theoretically, definition relations and production rules can therefore not be used as primitive inference steps. When we take a close look at the evaluation of definition relations and production rules, we can see that during evaluation these structures are replaced by the logical expressions representing the view and the premise/conclusion respectively. As a result of this, each (compound) logical expression can in the end by replaced by a, sometimes large, simple expression. Therefore, it is concluded that definition relations and production rules[2] (and the same goes for attribute facets and external relations) can be considered the primitive inference steps in LORE.

When the interpretation of these primitive inference steps is considered, we must conclude that the type of reasoning that is supported by LORE is deductive only.

Mapping LORE knowledge bases

The final step of the mapping consists of the mapping of LORE knowledge bases onto general framework programs (see Definition 3.10). This mapping is straightforward, a knowledge base consists of a subset of all possible LORE knowledge bases, which is exactly the same demand as is made with respect to the constraint set S_0. The initial labelling L_0 in terms of LORE was distinguished when the mapping of the variables was described. Finally, the query relations in LORE must be used as the equivalents of the queries of a program in terms of the general framework. In this, LORE is more restricting than the general framework, where a query can be chosen from outside the subset S_0 as well. For practical considerations,[3] the query must be selected from the LORE knowledge base itself.

[2] Definition relations with more than one range argument, and production rules with more than one conclusion, should, of course, be modelled by a number of primitive inference steps, each having the same premise expression.

[3] As a result of this, the syntactical correctness of a query can be checked by the Delfi3 compiler and linker before execution. Otherwise, the Delfi3 interpreter must contain some sort of query parser module, which is inefficient from a practical point of view.

6.6 LORE and the model of belief and disbelief

It has already been argued that the ability to reason with uncertainty is considered of importance in the development of many real-world knowledge-based systems. Therefore, the uncertainty model reported in Chapter 4 has been incorporated into LORE. Two aspects are important here, the representation of uncertain measurements in LORE language constructs and the propagation of uncertainty measurements during reasoning. There are two situations in LORE in which uncertainty measurements can be specified, in the initial value assignment of an attribute and in primitive inference steps that enable value assignment during execution.

Initial value assignment is done through the specification of a value facet of a shared attribute, or through the specification of a value of a private attribute in an individual object definition, as demonstrated in the following example.

```
Example 6.11
DOBJ Mammal
SHARED
   milk_giving : <BOOL> VAL TRUE MB 0.9
PRIVATE
   ....
EOBJ

IOBJ
   unknown_ungulate : Mammal(TRUE,TRUE,TRUE,"grass" MB 0.8,?,?,?)
EOBJ
```

In Example 6.11, the value facet of attribute *milk_giving* is filled with a value and an uncertainty label, indicating that the value *TRUE* of this attribute can only be assured by a measurement of belief (*MB*) of *0.9*. In individual object *unknown_ungulate*, the value *"grass"* of attribute *food* is specified with a measurement of belief of *0.8*. Attributes whose values are specified explicitly in the knowledge base in this way can be considered as premises, i.e. they belong to the initial labelling.

A value can be assigned during evaluation through the evaluation of an attribute's default facet or through the evaluation of the equation operator in a definition relation view or a production rule conclusion.

```
Example 6.12
DOBJ Mammal
SHARED
   ....
PRIVATE
   ....
|  ungulate : <BOOL>
      DEF TRUE MB 0.8
EOBJ
```

In Example 6.12, the default facet of attribute *ungulate* is filled with an expression (*TRUE*) and an uncertainty measurement, indicating that the result of evaluating the default expression can only be assured with a measurement of belief (*MB*) of *0.8*. When during reasoning, the value of attribute *ungulate* cannot be found, this default facet is used to obtain the value.

Example 6.13

```
DREL AssignUngulate
DOMAIN a : <Mammal>
[ a.ungulate = TRUE MB 1.0
  AND a.has_hooves = TRUE MB 0.9
  AND a.chews_cud = TRUE MB 0.7 ]
EREL
```

In Example 6.13, we have given a definition relation that can be used to fill the values of attributes *has_hooves* and *chews_cud* of an object of type *Mammal*. When the view of this definition relation is evaluated, these attributes are given the value *TRUE*, both with different measurements of belief, *MB 0.9* and *MB 0.7*. Of course, the attribute *ungulate* is given *1.0* for the measurement of belief.

Example 6.14

```
RULE IsZebra
GIVEN a : <Mammal>
IF
[ a.ungulate AND Zebra(a) ]
THEN
ABOUT a CONCLUDE identity = "zebra" MB 0.9
ERUL
```

In production rule *IsZebra*, the conclusion on the value of attribute *identity* of a given *Mammal* being equal to *"zebra"* can be made with a measurement of belief *MB* equal to *0.9*. With this measurement, only the strength of the production rule is given. In the propagation of uncertainty, the MB of the premise, *[a.ungulate AND Zebra(a)]*, is also used.

The propagation of uncertainty during the execution of LORE knowledge bases is done according to the propagation functions of the possibilistic model of belief and disbelief, as developed in Chapter 4. Since we have described the mapping of the LORE language constructs onto the general framework, the application of the correct uncertainty propagation functions is straightforward.

6.7 LORE and reason maintenance

In order to be able to support the enhanced propagation functions of the model of belief and disbelief (developed in Chapter 4), the presence of a derivation argument is necessary. By Theorem 4.11, we described the relation between this derivation and the structures in a reason maintenance model. Further, the use of reason maintenance techniques can increase the efficiency of the inference model (problem solver). This observation (recall Chapters 2 and 3) can also be made with respect to the LORE inference model, since this model is based on chronological backtracking. Therefore, a reason maintenance model was developed for LORE, not only to support the uncertainty model, but also to enhance the inference model.

The reason maintenance model is integrated with the inference model, and has no influence on the domain model. Hence, no changes are necessary to the language LORE as such. The impact that reason maintenance has on the working of the inference and uncertainty models, however, is enormous. The impact on the uncertainty model has already been made clear in Chapter 4, here we briefly discuss what can be expected from a combination of the inference and the reason maintenance model.

This combined model, which we call the *execution model* for short, enables control strategies for intelligent backtracking and learning from inferences. In [Goedhart, 1993c], we provide a formal description of this execution model, and describe a number of test situations in which this model has been used. We here give a brief overview of the most important results.

Dependency-directed backtracking [Stallman, 1977] is the basic strategy for using the reason of a contradiction in searching for the best alternative to restart reasoning. We have used an approach that is comparable to the one given in [Drakos, 1988] and especially to the *backjumping* approach described by Dechter [1989]. Both approaches postpone the construction of the set of potential alternative search paths until a contradiction has occurred. This is contrary to the method used by Doyle [1979], in which these sets are recorded all along the reasoning process and can lead to space capacity problems.

Intelligent backtracking, as implemented in LORE, does not lead to space capacity problems, but its pruning capacity is very dependent on the structure of the search space. When knowledge bases (or logic programs or constraint satisfaction problems) are well structured, the gain will be small. Unfortunately, we think that this is the case in the majority of real-world knowledge-based systems.

Besides intelligent backtracking, a learning while searching strategy was developed that enables the inference model to foresee coming conflicts in reasoning and to avoid repeating inferences that have already been made. In [Dechter, 1989] a comparable approach is described for solving constraint satisfaction problems.

Most important results are that *learning while searching* can give great reduction of the search space, but can also lead to enormous space capacity problems (especially when the full deep learning strategy [Dechter, 1989] is used). This can be decreased by recording only a subset of the executed inferences, those containing only few variables (enabling easy pruning). Such an approach is comparable to the suggested method for decreasing the complexity of the enhanced uncertainty propagation functions, as given at the end of Chapter 4. Results that are given in [Goedhart, 1993c] underline this conclusion.

Apart from the space capacity problem, the development of a general-purpose reason maintenance model was very attractive. It can also be used to develop other facilities (for instance an explanation facility or run-time debugging options).

6.8 Closing remarks

In this chapter, we have given a demonstration of how theory developed within our general framework can be used in Delfi3 and LORE. The importance of this demonstration is only illustrative, it does not increase or decrease the capabilities of the theory delineated in Chapters 3 and 4. The main objective was to give some idea of the practical applicability of the developed theory and the contours of its implementations. The general framework can be viewed as the blueprint of an environment for constructing the implementation model of a knowledge-based system. In this chapter, we have demonstrated an implementation environment that is based on the framework. In [Jonker, 1993], the Delfi3 environment is used to demonstrate a strategy for automatically deriving implementations from formally described design models.

The description of the use of uncertainty and reason maintenance within LORE has been rather informal. A formal description of the execution model (combination of inference and reason maintenance model) and a detailed discussion of the results can both be found in [Goedhart, 1993c]. Such a formal treatment is very interesting from an implementation point of view, but clearly lies outside the scope of the thesis.

Since LORE does not support language constructs for abductive reasoning, theory given in Chapter 5 has not been used throughout the current chapter. However, one could use Chapter 5 as a guideline to introduce abduction into LORE, which would give an abductive reasoning strategy for LORE constructs. Unfortunately, due to time limitations this interesting objective could not be realised.

Chapter 7

Discussion

Specific discussions on the theory set out in this thesis have been given at the end of the relevant chapters (Chapters 3, 4 and 5). Therefore, in this chapter we discuss some fundamental questions with respect to uncertainty management in compositional reasoning. Observations on the various different ideas to deal with the term uncertainty were made in Chapter 1. How are these notions represented in the general framework? A question that naturally follows from this is: What is the role of the compositional uncertain reasoning in the construction of real-world knowledge-based systems?

In Section 1, we comment on the representation of uncertainty by the framework. In Section 2, the concept of compositional reasoning is analysed as to its capabilities as a representation formalism for reasoning with uncertainty. In Section 3, we evaluate the partial implementation of the framework in the Delfi3 environment. Section 4 is used for a brief discussion towards solution ordering in the context of the framework, a subject that was raised in Chapters 3 and 5. Finally, an indication of related open problems is given in Section 5.

7.1 Characterising uncertainty

In Chapter 1, we distinguished four notions of uncertainty: completeness, preciseness, validity and reliability. This section contains a brief evaluation of the way in which these notions are represented by the framework.

It is our conviction, and that of others in literature, that the characteristic of *completeness* in data and knowledge can best be handled by both representation and reasoning aspects. In terms of reasoning, this is realised by the use of the *choose* inference step (remember the discussion on logic programming, concluding Chapter 3). The representation aspect of incompleteness is covered by the use of a three-valued logic, including the label *unknown*, enabling the continuation of reasoning in the absence of data. Therefore, incompleteness has a strong link with the uncertainty model (we have to define the embedding of unknown), but is not considered the key subject when we think of uncertainty.

In a similar way, we have treated the notion of *validity* in the framework in the context of the reasoning process. The definition of (uncertainty) conflicts in reasoning represents this notion within the framework. Thus, we prevent invalidity to enter the uncertainty labellings and restrict validity to the inference model. Again, a link between validity and the uncertainty model is obvious, but it is not an equality relation.

We consider the notion of *impreciseness*, or vagueness, to model a characteristic aspect of uncertainty. In the framework, specification of this aspect must be done in the definition (or choice) of the uncertainty label T_{ul}. As such, the framework does not provide extensive support for this notion of uncertainty, which may seem surprising at first sight. However, as we have already argued in Chapter 3, the specification of the uncertainty type is application specific. The framework supplies an environment in which these specific uncertainty models can be used and guaranteed to respect the underlying well-founded fundamentals.

In our opinion, the main characteristic of uncertainty is in the notion of *reliability*. This notion reflects the belief or trust (and doubt) one has in the valuation of statements made by a problem solver. The model of belief and disbelief, as reported in Chapter 4, is an example of an uncertainty model, which supports this aspect of uncertainty.

To conclude, we consider preciseness and reliability to be represented best by the *uncertainty* measurement in knowledge-based systems, where completeness and validness are more concerned with the representation of, and reasoning with knowledge.[1]

7.2 In defence of compositional reasoning

The role of compositional reasoning has been a subject of discussion for a number of years. Two related subjects are of interest here, the modelling potential of the compositional reasoning formalism and its practical importance in constructing knowledge-based systems.

In the literature, the most clear analysis concerning compositional uncertain reasoning is given by Pearl [1988] in his discussion of extensional systems. The merits of compositional reasoning are the use of efficient propagation functions due to the notion of *modularity*. This modularity principle consists of the notions of *locality* (uncertainty is propagated regardless of other influences) and *detachment* (uncertainty is propagated regardless of the way in which it was derived). According to Pearl [1988] and Heckerman [1988], the deficiencies of compositional reasoning are twofold: plausible, human-like reasoning patterns cannot be modelled, and the modularity principle is unrealistic in practical situations.

This thesis contains a clear analysis of the compositional uncertain reasoning formalism, which enables a well-founded evaluation of its benefits and limitations. There is not much disagreement on the advantages of the compositional approach: ease of domain model construction, well-defined inference principles and simple computational complexity of uncertainty propagation. Concerning the limitations of compositional uncertain reasoning, we can make more subtle distinctions in our criticism.

As far as the modularity principle is concerned, our investigations have clarified the distinction

[1] In this thesis, the notions of completeness and validity of knowledge have been studied with respect to the reasoning aspect only. With respect to the representation aspect, completeness (are there inference steps missing?) and validity (are there invalid inference steps present?) are not covered here.

between the notions of locality and detachment. The notion of locality is vital to the concept of compositional reasoning, it is part of its definition. However, the notion of detachment is not crucial to this concept. The enhanced propagation functions used in the model of belief and disbelief, see Chapter 4, violate the principle of detachment by using the derivation of propositions in the propagation of uncertainty. Therefore, in our opinion, only the principle of locality is fundamental for compositional reasoning.

In Chapter 5, we showed that plausible human-like reasoning patterns can be modelled within the compositional reasoning formalism. Clearly, limitations are formulated for the expressiveness of these patterns in the compositional approach. We agree with Pearl that the modelling of bidirectional influences are beyond the capabilities of compositional reasoning. However, this does not mean that plausible reasoning has become completely impossible. The strategies called explain away and joint explain, see again Chapter 5, can be modelled in compositional reasoning and, when used in situations that fulfil the demanded restrictions, enable these plausible reasoning patterns.

A brief overview of the reasoning formalisms that have been developed in the past decades (knowledge-based systems, causal networks, neural networks, case-based systems) makes it clear that no overall, general reasoning technique has yet been found. All formalisms have their strong points and their weak points, and the development of a general purpose reasoning formalism is probably impossible. In a broader sense, we are convinced that for solving large and complex problems, a hybrid problem-solving strategy will likely give the best result. In such a multi-strategy system, the compositional reasoning formalism is considered an attractive alternative for modelling parts of the system. However, a lot of research is still needed on the cooperation of and communication between these approaches within such an environment.

7.3 Implementation aspects

In Chapter 6, the knowledge engineering environment Delfi3 was analysed in the context of the general framework of Chapter 3. This demonstrated the applicability of (a part of) the theories developed within the framework in an existing implementation environment. The enhancements primarily concerned the uncertainty model, the existing model (from the literature) could easily be replaced by the uncertainty model developed in Chapter 4.

Implementing the reason maintenance model for Delfi3, which was not available, proved to be quite a time-consuming, but not a very difficult piece of work. Once the mapping of the LORE constructs onto the general framework was defined, the structure and functionality of the reason maintenance model were given by the framework. Further, the implementation of the reason maintenance model increased the execution performance of the inference model of the Delfi3 interpreter. Strategies for intelligent backtracking and learning from past inferences were implemented [Goedhart, 1993c] and a future implementation of an explanation facility will certainly profit from the presence of the reason maintenance model.

7.4 Ordering solutions within the framework

Diagnostic problem solving is concerned with selecting explanations for a number of observed phenomena. A well-known problem in diagnostic reasoning is that more than one solution can be found for a particular problem. However, in most cases, one is interested in the most promising, or most probable one(s). In other words, some ordering or classification of the solutions is desired. In our framework, this situation occurs when more than one extension is found for a given program. However, our framework does not offer such a method for the ordering of extensions.

A trivial approach would be to assume the user of the framework to be able to develop an intelligent control structure that generates the most plausible extensions first, i.e the ordering is done through the control structure within the knowledge base. Besides the practical objections, such an approach would demand the presence of well-defined meta-reasoning constructs at the control level, which is not supported by the framework as it is.

Given the representation formalism of the framework, there are two fundamental ways to tackle this problem. Firstly, by changing the outcome of the control function $C\mathcal{F}$(see for instance Definition 3.11) from $\{\mathcal{T},\mathcal{F}\}$ to some more general outcome. This approach introduces a measurement representing the strength of the different types of inference steps, and an accompanying set of propagation functions. The extensions for the query Q can then be ordered to the results of $C\mathcal{H}$(Q) in terms of this measurement.

An alternative approach is to order the extensions to the outcome of a comparison measure between the individual extensions. The definition of a partial ordering is a common strategy to rank extensions or solutions. According to its definition, an extension L_Q is a labelling of the set of atomic propositions A, satisfying the constraints of the program. The reasoning structure that has been used in deriving this extension is represented in the underlying dependency network. Therefore, in our model, a partial ordering can be defined on this dependency network, consisting of propositions, justifications and assumptions. Taking advantage of the equivalence with dependency graphs, theory from (inexact) graph matching can be used to construct the partial ordering.

When graph matching is used, a transformation cost function must be specified between the dependency graphs. This transformation cost function can be specified by using the uncertain labelling of propositions and the strengths of the used deductive and explanatory inference steps, recorded in the justifications and assumptions. Using the uncertainty labels will provide a plausible ordering of solutions that can be further enhanced by restricting the transformation cost function to be used on selected subsets of propositions and inference steps, depending on the specific problem instance to be solved.

7.5 Open questions

It is obvious that, even in the restricted area of research covered by this thesis, numerous open questions remain to be answered. We restrict ourselves here to one aspect of (compositional) reasoning that is considered of major importance from both a theoretical and a practical viewpoint. Knowledge-based systems are used more and more often in time-critical environments. It is obvious that such systems must be able to explicitly handle the aspect of time in both representation and reasoning.

Clearly, the notion of time is not covered by the general framework. Therefore, an interesting research objective would be to study whether a temporal component can be introduced into the general framework in a way similar to the integration of the notion of uncertainty. Following this, it would be even more interesting to study a formalism using the *combination* of time and uncertainty in compositional reasoning simultaneously.

References and Literature

[Adams, 1976] Adams J.B., *A Probabilisty Model of Medical Reasoning and the MYCIN Model*, In: Mathematical Biosciences, volume 32, 1976.

[Adams, 1984] Adams J.B., *Probabilistic Reasoning and Certainty Factors*, In: [Buchanan, 1984].

[Bar, 1981] Bar A., Feigenbaum E., *The Handbook of Artificial Intelligence I*, Morgan Kaufmann Publishers, Los Altos, CA, USA, 1981.

[Bobrow, 1975] Bobrow D.G., Collins A.M. (editors), *Representation and Understanding: Studies in Cognitive Science,* Academic Press, New York, NJ, USA, 1975.

[Bonissone, 1986] Bonissone P.P., Decker K.S., *Selecting Uncertainty Calculi and Granularity: Trading-off Precision and Complexity*, In: [Kanal, 1986].

[Bonissone, 1987] Bonissone P.P., *Summarizing and Propagating Uncertain Information with Triangular Norms*, In: International Journal of Approximate Reasoning, volume 1, 1987.

[Bonissone, 1991a] Bonissone P.P., *Approximate Reasoning Systems: a Personal Perspective*, In: Proceedings of the 9th AAAI, Anaheim, CA, USA, 1991.

[Bonissone, 1991b] Bonissone P.P., Henrion M., Kanal L.N., Lemmer J.F. (editors), *Uncertainty in AI 6*, North-Holland, Amsterdam, the Netherlands, 1991.

[Brachman, 1979] Brachman R.J., *On the Epistemological Status of Semantic Networks*, In: [Findler, 1979].

[Buchanan, 1984] Buchanan B.G., Shortliffe E.H. (editors), *Rule Based Expert Systems. The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, Reading, MA, USA, 1984.

[Bundy, 1984] Bundy A., Boulay B. du, Howe J., Plotkin G., *How to get a Ph.D. in AI?*, In: O'Shea T., Eisenstadt M. (editors), *AI: tools, techniques and applications*, Harper & Row Publishers, New York, NJ, USA, 1984.

[Clancey, 1983] Clancey W.J., *The Epistemology of a Rule-based Expert System: a Framework for Explanation*, In: Artificial Intelligence, volume 20, 1983.

[Clark, 1978] Clark K.L., *Negation as Failure*, In: Gallaire H., Minker J. (editors), *Logic and Databases*, Plenum Press, New York, NJ, USA, 1978.

[Cohen, 1985] Cohen P.R., *Heuristic Reasoning About Uncertainty: An Artificial Intelligence Approach*, Pitman, London, UK, 1985.

[Collins, 1970] Collins A.M., Quillian M.R., *Facilitating Retrieval from Semantic Memory*, In: Sanders A. (editor), *Acta Psychologica 33, Attention and Performance*, North-Holland, Amsterdam, the Netherlands, 1970.

[Cooper, 1990] Cooper G.F., *Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks*, In: Artificial Intelligence, volume 42, 1990.

[van Daalen, 1993] Daalen C. van, *Validating Medical Knowledge Based Systems*, Ph.D. thesis, Delft University of Technology, the Netherlands, 1993.

[D'Ambrosio, 1990] D'Ambrosio B.D., *Process, Structure and Modularity in Reasoning with Uncertainty*, In: [Shachter, 1990].

[D'Ambrosio, 1991] D'Ambrosio B.D., Smets P., Bonissone P.P. (editors), *Uncertainty in AI 7*, Morgan Kaufmann Publishers, Los Altos, CA, USA, 1991.

[Davis, 1977] Davis R., Buchanan B.G., Shortliffe E.H., *Production Rules as a Representation for a Knowledge-based Consultation System*, In: Artificial Intelligence, volume 8, 1977.

[Dechter, 1989] Dechter R., *Enhancement Schemes for Constraint Processing: Backjumping, Learning and Cutset Decomposition*, In: Artificial Intelligence, volume 41, 1989.

[Dempster, 1967] Dempster A.P., *Upper and Lower Probabilities Induced by a Multivalued Mapping*, In: Annals of Mathematical Statistics, volume 38, 1967.

[Doyle, 1979] Doyle J., *A Truth Maintenance System*, In: Artificial Intelligence, volume 12, 1979.

[Drakos, 1988] Drakos N., *Reason Maintenance in Horn-clause Logic Programs*, In: [Smith, 1988].

[Dubois, 1992] Dubois D., Wellman M.P., D'Ambrosio B.D., Smets P. (editors), *Uncertainty in AI 8*, Morgan Kaufmann Publishers, Los Altos, CA, USA, 1992.

[Duda, 1976] Duda R.O., Hart P.E., Nilsson N.J., *Subjective Bayesian Methods for Rule Based Inference Systems*, Report 124, Artificial Intelligence Center, SRI International, Menlo Park, CA, USA, 1976.

[Elkan, 1989] Elkan C., *Logical Characterizations of Non-monotonic TMS's*, In: Kreczmar A., Mirkowska G. (editors), *Mathematical Foundations of Computer Science*, Springer-Verlag, Berlin, Germany, 1989.

[Etherington, 1988] Etherington D.W., *Reasoning with Incomplete Information*, Pitman, London, UK, 1988.

[Fikes, 1985] Fikes R., Kehler T., *The Role of Frame-based Representation in Reasoning*, In:Communications of the ACM, volume 28, number 9, 1985.

[Findler, 1979] Findler N.V. (editor), *Associative Networks: Representation and Use of Knowledge by Computers*, Academic Press, New York, NJ, USA, 1979.

[van der Gaag, 1990] Gaag L.C. van der, *Probability-Based Models for Plausible Reasoning*, Ph.D. thesis, University of Amsterdam, the Netherlands, 1990.

[Goedhart, 1991] Goedhart B., Jonker W., Swaan Arons H. de, *A Knowledge Engineering Environment for the Development of Knowledge Based Systems in Complex Domains*, In: Proceedings of the 4th AIT, Amsterdam, the Netherlands, 1991. (In Dutch)

[Goedhart, 1992] Goedhart B., Swaan Arons H. de, *Delfi3 User Manual version 2.0*, Delft University of Technology, the Netherlands, 1992. (Available from the author)

[Goedhart, 1993a] Goedhart B., Lubbe J.C.A. van der, *Probabilistic Interpretations and Internal (In)consistencies of the Certainty Factor Model*, In: Proceedings of the 13th Conference on Artificial Intelligence, Expert Systems and Natural Language, Avignon, France, 1993.

[Goedhart, 1993b] Goedhart B., *A General Framework for Compositional Uncertain Reasoning*, In: Proceedings of the 6th NAIC, Enschede, the Netherlands, 1993.

[Goedhart, 1993c] Goedhart B., *LORE and Reason Maintenance*, Internal report 93-88, Delft University of Technology, the Netherlands, 1993.

[Goedhart, 1994] Goedhart B., *Abduction and Uncertainty in Compositional Reasoning*, In: Proceedings of the 11th ECAI, Amsterdam, the Netherlands, 1994.

[Goodwin, 1987] Goodwin J.W., *A Theory and System for Non-monotonic Reasoning*, Ph.D. thesis, Linköping University, Sweden, 1987.

[Gordon, 1984] Gordon J., Shortliffe E.H., *The Dempster-Shafer Theory of Evidence*, In: [Buchanan, 1984].

[Hájek, 1992] Hájek P., Havránek T., Jirousek R., *Uncertain Information Processing in Expert Systems*, CRC Press, Boca Raton, FL, USA, 1992.

[Hayes, 1969] Hayes P., McCarthy J., *Some Philosophical Problems from the Standpoint of AI*, In: Meltzer B., Mitchie D. (editors), *Machine Intelligence*, volume 4, Edinburgh Universal Press, Edinburgh, UK, 1969.

[Heckerman, 1986] Heckerman D.E., *Probabilistic Interpretations for MYCIN's Certainty Factors*, In: [Kanal, 1986].

[Heckerman, 1988] Heckerman D.E., Horvitz E.J., *The Myth of Modularity in Rule Based Systems*, In: [Kanal, 1988].

[Heckerman, 1992] Heckerman D.E., Shortliffe E.H., *From Certainty Factors to Belief Networks*, In: Artificial Intelligence in Medicine, volume 4, 1992.

[Henrion, 1987] Henrion M., *Uncertainty in AI: Is Probability Epistemologically and Heuristically Adequate?*, In: Mumpower J.L. (editor), *Expert Judgement and Expert Systems*, Springer-Verlag, Berlin, Germany, 1987.

[Henrion, 1990] Henrion M., Shachter R.D., Kanal L.N., Lemmer J.F. (editors), *Uncertainty in AI 5*, North-Holland, Amsterdam, the Netherlands, 1990.

[Horvitz, 1986] Horvitz E.J., Heckerman D.E., *The Inconsistent Use of Measures of Certainty in Artificial Intelligence*, In: [Kanal, 1986].

[Jansen, 1993] Jansen E.P., *Improving Production Schedules without Scheduling*, Ph.D. thesis, Delft University of Technology, the Netherlands, 1993.

[Jaspers, 1990] Jaspers R.B.M., *Medical Decision Support*, Ph.D. thesis, Delft University of Technology, the Netherlands, 1990.

[Jonker, 1990] Jonker W., *The Design and Implementation of a Knowledge Representation and Processing Language*, Ph.D. thesis, University of Utrecht, the Netherlands, 1990.

[Jonker, 1993] Jonker W., Goedhart B., *KBS Development: From Design Models to Implementations*, In: Proceedings of the 5th SEKE, San Francisco, CA, USA, 1993.

[Josephson, 1987] Josephson J.R., Chandrasekaran B., Smith J., Tanner M.C., *A Mechanism for Forming Composite Explanatory Hypotheses*, In: IEEE Transactions on SMC, volume 17, number 3, 1987.

[Kanal, 1986] Kanal L.N., Lemmer J.F. (editors), *Uncertainty in AI 1*, North-Holland, Amsterdam, the Netherlands, 1986.

[Kanal, 1988] Kanal L.N., Lemmer J.F. (editors), *Uncertainty in AI 2*, North-Holland, Amsterdam, the Netherlands, 1988.

[Kanal, 1989] Kanal L.N., Levitt T.S., Lemmer J.F. (editors), *Uncertainty in AI 3*, North-Holland, Amsterdam, the Netherlands, 1989.

[Kelleher, 1988] Kelleher G., Smith B.M., *A Brief Introduction to Reason Maintenance Systems*, In: [Smith, 1988].

[Kim, 1983] Kim J.H., Pearl J., *A Computational Model for Causal and Diagnostic Reasoning in Inference Engines*, In: Proceedings of the 8th IJCAI, Karlsruhe, Germany, 1983.

[de Kleer, 1986] Kleer J. de, *An Assumption-based TMS; Extending the ATMS; Problem Solving with the ATMS*, In: Artificial Intelligence, volume 28, 1986.

[Konolige, 1988] Konolige K., *On the Relation between Default Logic and Autoepistemic Logic*, In: Artificial Intelligence, volume 35, 1988.

[Kowalski, 1974] Kowalski R.A., *Predicate Logic as a Programming Language*, In: Proceedings of the IFIP, 1974.

[Kowalski, 1979] Kowalski R.A., *Algorithm = Logic + Control*, In: Communications of the ACM, volume 22, number 7, 1979.

[Krijgsman, 1992] Krijgsman W., Reiber J.H.C., Goedhart B., Backer E., *A Knowledge Based Approach toward the Standardisation of Image Analysis in Nuclear Cardiology: the ESATS project*, In: Proceedings of the 5th NAIC, Delft, the Netherlands, 1992. (In Dutch)

[Kyburg, 1991] Kyburg H.E., *Why do we need Foundations for Modelling Uncertainties?*, In: [D'Ambrosio, 1991].

[Laskey, 1989] Laskey K.B., Lehner P.E., *Assumptions, Belief and Probabilities*, In: Artificial Intelligence, volume 41, 1989.

[Lauritzen, 1988] Lauritzen S.L., Spiegelhalter D., *Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems*, In: Journal of the Royal Statistical Society, serie B, volume 50, number 2, 1988.

[Levesque, 1979] Levesque H.J., Mylopoulos J., *A Procedural Semantics for Semantic Networks*, In: [Findler, 1979].

[Lloyd, 1984] Lloyd J.W., *Foundations of Logic Programming*, Springer-Verlag, Berlin, Germany, 1984.

[van der Lubbe, 1990] Lubbe J.C.A. van der, Backer E., Krijgsman W., *Models for Reasoning with Multi-type Uncertainty in Expert Systems*, In: Proceedings of the 3rd IPMU, Paris, France, 1990.

[van der Lubbe, 1991] Lubbe J.C.A. van der, Backer E., Goedhart B., *An Expert System for the Automatic Analysis of Cloud Types in Satellite Imagery: The METEODES Project*, In: Proceedings of the 11th Conference on Expert Systems and their Applications, Avignon, France, 1991.

[van der Lubbe, 1993] Lubbe J.C.A. van der, Backer E., *Human-like Reasoning under Uncertainty in Expert Systems*, In: Jorna R., Heusden B. van, Posner R. (editors), *Signs, Search and Communication*, Walter de Gruyter, Berlin, Germany, 1993.

[Lukaszewicz, 1991] Lukaszewicz V., *Non-monotonic Reasoning*, Prentice-Hall, New York, NJ, USA, 1991.

[Martins, 1988] Martins J.P., Shapiro S.C., *A Model for Belief Revision*, In: Artificial Intelligence, volume 35, 1988.

[McCarthy, 1980] McCarthy J., *Circumscription, a form of Non-monotonic Reasoning*, In: Artificial Intelligence, volume 13, 1980.

[McCarthy, 1986] McCarthy J., *Applications of Circumscription to Formalizing Common-sense Knowledge*, In: Artificial Intelligence, volume 28, 1986.

[McDermott, 1978] McDermott D.V., *Tarski Semantics, or no Notation without Denotation!*, In: Cognitive Science, volume 2, number 3, 1978.

[McDermott, 1980] McDermott D.V., Doyle J., *Non-monotonic Logic I*, In: Artificial Intelligence, volume 13, 1980.

[McDermott, 1982] McDermott D.V., *Non-monotonic Logic II: Non-monotonic Modal Theories*, In: Journal of the ACM, volume 29, number 1, 1982.

[McDermott, 1991] McDermott D.V., *A General Framework for Reason Maintenance*, In: Artificial Intelligence, volume 50, 1991.

[Mellish, 1987] Mellish C.S., *Abstract Interpretation of Prolog Programs*, In: Abramsky S., Hankin C.L. (editors), *Abstract Interpretation of Declarative Languages*, Ellis Horwood, Chichester, UK, 1987.

[Minsky, 1975] Minsky M., *A Framework for Representing Knowledge*, In: Winston P.H. (editor), *The Psychology of Computer Vision*, McGraw-Hill, New York, NJ, USA, 1975.

[Moore, 1985] Moore R.C., *Semantical Considerations on Non-monotonic Logic*, In: Artificial Intelligence, volume 25, 1985.

[Neapolitan, 1990] Neapolitan R.E., *Probabilistic Reasoning in Expert Systems*, John Wiley & Sons, New York, NJ, USA, 1990.

[Newell, 1972] Newell A., Simon H.A., *Human Problem Solving*, Prentice-Hall, New York, NJ, USA, 1972.

[Pearl, 1982] Pearl J., *Reverend Bayes on Inference Engines: a Distributed Hierarchical Approach*, In: Proceedings of the 5th AAAI, Menlo Park, CA, USA, 1982.

[Pearl, 1988] Pearl J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, Los Altos, CA, USA, 1988.

[Peirce, 1958] Peirce C.S., *Collected Papers of Charles Sanders Peirce*, Hartshore C., Weiss P., Burks A. (editors), 8 volumes, Cambridge Press, Cambridge, MA, USA, 1931-1958.

[Peng, 1990] Peng Y., Reggia J.A., *Abductive Inference Models for Diagnostic Problem Solving*, Springer-Verlag, Berlin, Germany, 1990.

[Poole, 1991] Poole D., *Representing Bayesian Networks within Probabilistic Horn Abduction*, In: [D'Ambrosio, 1991].

[Poole, 1993] Poole D., *Probabilistic Horn Abduction and Bayesian Networks*, In: Artificial Intelligence, volume 64, 1993.

[Pople, 1973] Pople H.E., *On the Mechanization of Abductive Logic*, In: Proceedings of the 3rd IJCAI, Stanford, CA, USA, 1973.

[Pople, 1977] Pople H.E., *The Formation of Composite Hypotheses in Diagnostic Problem Solving*, In: Proceedings of the 5th IJCAI, Cambridge, MA, USA, 1977.

[Provan, 1988] Provan G.M., *A Complexity Analysis of Assumption-based Truth Maintenance Systems*, In: [Smith, 1988].

[Provan, 1990] Provan G.M., *The Application of Dempster-Shafer Theory to a Logic-based Visual Recognition System*, In: [Henrion, 1990].

[Qiu, 1992] Qiu D., Dudeck J., *Certainty Factor Theory: Its Probabilistic Interpretations and Problems*, In: Artificial Intelligence in Medicine, volume 4, 1992.

[Quillian, 1966] Quillian M.R., *Semantic Memory*, Ph.D. thesis, Carnegie Institute of Technology, USA, 1966.

[Reinfrank, 1989] Reinfrank M., *Fundamentals and Logical Foundations of Truth Maintenance*, Ph.D. thesis, Linköping University, Sweden, 1989.

[Reiter, 1978] Reiter R., *On Closed-World Data Bases*, In: Gallaire H., Minker J. (editors), *Logic and Databases*, Plenum Press, New York, NJ, USA, 1978.

[Reiter, 1980] Reiter R., *A Logic for Default Reasoning*, In: Artificial Intelligence, volume 13, 1980.

[Robinson, 1965] Robinson J.A., *A Machine-Oriented Logic based on the Resolution Principle*, In: Journal of the ACM, volume 12, number 1, 1965.

[Roest, 1992] Roest D.M., *Using an Expert System for Cloud Classification*, M.Sc. thesis, Department of Electrical Engineering, Delft University of Technology, the Netherlands, 1992. (In Dutch)

[Sandewall, 1986] Sandewall E., *Non-monotonic Inference Rules for Multiple Inheritance with Exceptions*, In: Proceedings of the IEEE, volume 74, number 10, 1986.

[Schokking, 1992] Schokking M., *Knowledge Acquisition for METEODES*, M.Sc. thesis, Department of Electrical Engineering, Delft University of Technology, the Netherlands, 1992. (In Dutch)

[Schweizer, 1983] Schweizer B., Sklar A., *Probabilistic Metric Spaces*, North-Holland, New York, NJ, USA, 1983.

[Shachter, 1990] Shachter R.D., Levitt T.S., Kanal L.N., Lemmer J.F. (editors), *Uncertainty in AI 4*, North-Holland, Amsterdam, the Netherlands, 1990.

[Shafer, 1976] Shafer G., *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ, USA, 1976.

[Shenoy, 1990] Shenoy P., Shafer G., *Axioms for Probability and Belief-function Propagation*, In: [Schachter, 1990].

[Shortliffe, 1975] Shortliffe E.H., Buchanan B.G., *A Model of Inexact Reasoning in Medicine*, In: Mathematical Biosciences, volume 23, 1975.

[Shortliffe, 1984] Shortliffe E.H., Buchanan B.G., *A Model of Inexact Reasoning in Medicine*, In: [Buchanan, 1984].

5. There should be an explicit representation of ignorance to allow the user to make non-committing statements, that is, to express lack of conviction about the certainty of any of the available choices or events. Some measure of ignorance, similar to the concept of entropy, should be available to guide the gathering of discriminant information.

A measure of ignorance is not supported by our framework. However, lack of conviction about uncertainty can be specified by using the *unknown* label.

6. The representation must be or at least must appear to be natural to the user to enable him or her to describe uncertain input and to interprete uncertain output. The representation must also be natural to the expert to enable him or her to elicit consistent weights representing the strength of the implication of each rule.

A similar comment can be made as was made on demand number 3. When developing an uncertainty model, the choice of the basic representation measurements must appear natural to both user and expert (or knowledge engineer). In the model of belief and disbelief, no specific semantic choice is made, as previously explained in Chapter 4.

A.2 Desiderata for the inference layer

Four demands are specified by Bonissone concerning the inference layer.

7. The combining rules should not be based on global assumptions of evidence independence.

Recalling the propagation functions reported in Chapter 4, we can conclude that this demand is met.

8. The combining rules should not be based on global assumptions of hypotheses exhaustiveness and exclusiveness.

Again, recalling the propagation functions developed in Chapter 4, we state that the conditions on exhaustiveness and exclusiveness are not assumed.

9. The combining rules should maintain the closure of the syntax and semantics of the representation of uncertainty.

From the uncertainty propagation functions set out in Chapter 4 it is clear that this demand is satisfied.

10. Any function used to propagate and summarize uncertainty should have clear semantics. This is needed both to maintain the semantic closure of the representation and to allow the control layer to select the most appropriate combining rules.

During the development of uncertainty propagation functions given in Chapter 4, the semantics of these functions played a dominant role. The model of belief and disbelief itself contains selection criteria (recall Propagation function 4.7 through 4.10) to choose between the different propagation functions. We comment further on this when discussing demand number 15.

A.3 Desiderata for the control layer

Five demands are specified by Bonissone concerning the control layer.

11. There should be a clear distinction between a conflict in the information (i.e. violation of consistency) and ignorance about the information.

Since ignorance is not explicitly supported by our framework, this demand does not make much sense in its evaluation. However, we repeat our comment given on demand number 4, in which is stated that a conflict is defined explicitly in our framework. This clearly distinguishes the notion of a conflict in information from uncertainty or ignorance about the information.

12. The aggregation and propagation of uncertainty through the reasoning process must be traceable to resolve conflicts or contradictions, to explain the support of conclusions, and to perform meta-reasoning for control.

This demand can be summarised by our statement that the reason maintenance model must be part of the general framework, since all these characteristics (i.e. contradiction resolving) are encapsulated in the reason maintenance model.

13. It should be possible to make pairwise comparisons of uncertainty, as the induced ordinal or cardinal ranking is needed for performing any kind of decision-making activities.

This facility is not supported by our framework. In the closing discussion on the framework this had already been observed and some hints were given as to satisfying this demand. We agree on the usability of this demand, but we do not consider it of vital importance.

14. There should be a second-order measure of uncertainty. It is important to measure the uncertainty of the information as well as the uncertainty of the measure itself.

Again we agree on the attractiveness of the demand, but do not consider it a basic requirement of a general framework. In our framework such a second-order measurement can be related to the set of inference steps, i.e. it should be part of the control layer. This would demand a study of second-order uncertainty propagation in the control layer, a subject that is not covered in this thesis.

15. It should be possible to select the most appropriate combination rule by using a declarative form of control (i.e. by using a set of context dependent rules that specify the selection policies).

The selection of the most appropriate uncertainty propagation to be used in a specific situation is supported by the model of belief and disbelief. This means that the selection is made by the control layer, but on criteria specified in the uncertainty model. No specific formalism is introduced in our framework to explicitly represent the selection criteria. Although such a formalism would be more flexible, we claim that these selection criteria are strongly related to the uncertainty model that is used and can therefore better be formulated by the uncertainty model itself.

Appendix B

Embedding Dempster-Shafer Theory

In Chapter 4, we described the embedding of an uncertainty model in the general framework for compositional uncertain reasoning, as developed in Chapter 3. In the concluding discussion of Chapter 3, we analysed a number of existing uncertainty models when used with the general framework. Here, a closer look is taken at the embedding of the Dempster-Shafer theory in this general framework, this embedding was used in Example 3.4.

In Section 1 the basic definitions of the Dempster-Shafer theory are described. In Section 2, Dempster's rule of combination is given, followed by the deductive interpretation that is used in the Dempster-Shafer theory. The embedding of the theory in the general framework described in Chapter 3 forms the contents of Section 3. Finally, in Section 4, the appendix is concluded with a brief discussion on related work on the Dempster-Shafer theory in the context of its logical foundations and reason maintenance.

B.1 The basic definitions

We only briefly describe the basic definitions of the Dempster-Shafer theory. The basic theory is developed in [Dempster, 1967; Shafer, 1976]. For an example of the use of the theory in compositional reasoning systems one is referred to [Gordon, 1984].

The theory starts with the definition of the frame of discernment, a set of disjunct hypothesis for some variable.

> Definition B.1
>
> Let Θ be a set of disjunct hypotheses for some variable x, then Θ is called the *frame of discernment* for x.

Then, a function is defined on all subsets of the power set 2^Θ of Θ, which has the characteristics of a probability distribution. This basic probability assignment will have a pivoting role within the theory.

> Definition B.2
>
> Let Θ be a frame of discernment for some variable x.
>
> If there exists a function m: $2^\Theta \rightarrow [0..1]$ such that for each $X \subseteq 2^\Theta$
>
> (i) $m(X) \geq 0$,
>
> (ii) $m(\varnothing) = 0$,
>
> (iii) $\Sigma\, m(X) = 1$.
>
> Then m is called a *basic probability assignment* on Θ.

Based on the basic probability assignment, two functions are defined. A belief or credibility function that reflects the total belief of some subset X, and a plausibility function that reflects the total amount of belief that is not assigned to $\overline{X}$.

Definition B.3
Let Θ be a frame of discernment for some variable x and m a basic probability assignment on Θ. The *belief function* Bel: $2^{\Theta} \rightarrow [0..1]$ is defined for each $X \subseteq 2^{\Theta}$ by

$$\mathrm{Bel}(X) = \sum_{Y \subseteq X} m(Y)$$

Definition B.4
Let Θ be a frame of discernment for some variable x and m a basic probability assignment on Θ. The *plausibility function* Pl: $2^{\Theta} \rightarrow [0..1]$ is defined for each $X \subseteq 2^{\Theta}$ by

$$\mathrm{Pl}(X) = \sum_{X \cap Y \neq \varnothing} m(Y)$$

The mapping of the basic probability assignment on the belief and plausibility function is called the Möbius transform.

The following relations between belief and plausibility function can be shown to hold.

Theorem B.1
Let Θ be a frame of discernment for some variable x, m a basic probability assignment on Θ and Bel(X) and Pl(X) the belief and plausibility function. For each $X \subseteq 2^{\Theta}$ the following relations hold
(i) $\mathrm{Pl}(X) \geq \mathrm{Bel}(X)$,
(ii) $\mathrm{Pl}(X) = 1\text{-}\mathrm{Bel}(\overline{X})$.

Proof
For the proof of this theorem one is referred to [Shafer, 1976].

Based on this, the belief interval is defined for each subset $X \subseteq 2^{\Theta}$. This interval is considered to model the lower (Bel) and upper (Pl) bound as well as the uncertainty (Pl-Bel) of the belief in subset X.

Definition B.5
Let Θ be a frame of discernment for some variable x, m a basic probability assignment on Θ and Bel(X) and Pl(X) the belief and plausibility function for each $X \subseteq 2^{\Theta}$. Then [Bel(X),Pl(X)] is called the *belief interval* for X.

B.2 Dempster's rule of combination

In [Dempster, 1967] we find the definition of the propagation function for the combination of two pieces of evidence. This function is called Dempster's rule of combination, it defines an updated probability mass function for a subset $X \subseteq 2^{\Theta}$. We provide a definition similar to the one in [Shafer, 1976].

Definition B.6

Let Θ be a frame of discernment for variable x and m_1 and m_2 basic probability assignments on Θ. The *combination function* $m_1 \oplus m_2$: $2^\Theta \rightarrow [0..1]$ is defined for each $X \subseteq 2^\Theta$ by

(i) $m_1 \oplus m_2 (\emptyset) = 0$,

(ii) $m_1 \oplus m_2 (X) =$

$$\frac{\sum\limits_{Y \cap Z = X} m_1(Y)*m_2(Z)}{\sum\limits_{Y \cap Z \neq \emptyset} m_1(Y)*m_2(Z)}$$

With Dempster's rule of combination, the propagation function for the combination has been defined. Unfortunately, this is the only propagation function that is available from the theory. The missing propagation functions for the implication and the logical operators are not defined within the Dempster-Shafer theory. Various proposals of these sets of propagation functions can be found in the literature, but a definite set has not been found yet. With respect to the propagation function for the implication, there is a common agreement for the function that we define next.

Definition B.7

Let Θ be a frame of discernment for variable x and m_e and $m_{e \Rightarrow h}$ basic probability assignments on Θ. The *implication function* $m_e \otimes m_{e \Rightarrow h}$: $2^\Theta \rightarrow [0..1]$ is defined for $E,H \subseteq 2^\Theta$ by
$m_e \otimes m_{e \Rightarrow h} (H) = m_{e \Rightarrow h}(H)*\mathrm{Bel}(E)$.

This definition is intuitively attractive, the basic probability assignment of H is the result of the product of the strength of the implication e ⇒ h, given by $m_{e \Rightarrow h}(H)$ and the belief function of the premise, given by Bel(E). When there is no belief in the premise, Bel(E)=0, the resulting belief update in H will also be zero, which can be considered as an unsuccessful attempt to use the implication.

The propagation functions for the logical operators are again subject to discussion. We do not go into the discussion on these functions here. Instead, we illustrate an embedding based on the propagation functions for the implication and combination only.

B.3 Dempster-Shafer theory and the general framework

In the foregoing, we described the Dempster-Shafer theory, based on the basic probability assignment and the belief and plausibility functions. In this section, we describe the relation between the Dempster-Shafer theory and the general framework delineated in Chapter 3. Firstly, a basic probability assignment is formulated for the three-valued labelling of *true*, *false* and *unknown*. This results in corresponding belief and plausibility functions and enables a correct embedding of the three-valued model in the Dempster-Shafer theory. Finally, the uncertainty model is completed by defining a deductive inference function.

Theorem B.2

Let Θ be a frame of discernment for variable x.

Then m_{true}, m_{false} and m_{unknown} are basic probability assignments on Θ for $A, X \subseteq \Theta$.

$$m_{\text{true}}(X) = \begin{cases} 1 & X=A \\ 0 & \text{otherwise} \end{cases}$$

$$m_{\text{false}}(X) = \begin{cases} 0 & X=A \\ 1 & X=\Theta-A \\ 0 & \text{otherwise} \end{cases}$$

$$m_{\text{unknown}}(X) = \begin{cases} 0 & X=A \\ 1 & X=\Theta \\ 0 & \text{otherwise} \end{cases}$$

Proof

Straight from the definition of a basic probability assignment.

Theorem B.3

Let the *uncertainty type* T_{ul} of the Dempster-Shafer theory is equal to [Bel(X),Pl(X)], the belief interval for X according to Definition B.5.

The *uncertainty propagation functions* ($f_{\Rightarrow}$,$f_{||}$) of the Dempster-Shafer theory are defined in Definition B.6 and B.7.

Then a *correct embedding* of the three valued model in the Dempster-Shafer theory can be defined through B(*t*)=[1,1], B(*u*)=[0,1] and B(*f*)=[0,0].

Proof

According to the truth table of the propagation for the combination for [Bel(X),Pl(X)] (see Table B.1) and that of the general framework (see Table 3.4), it can be concluded that the given embedding is correct.

It should be noted that Dempster's rule of combination (see Definition B.6) is indeed undefined and results in a conflict (†) when it is calculated for a combination of the labels true=[1,1] and false=[0,0].

We can complete the model with a definition of a deductive inference function. The strength of the inference step is given by [$\text{Bel}_{\Rightarrow}$(a),$\text{Pl}_{\Rightarrow}$(a)].

Example B.15

A *deductive inference function* $I\mathcal{F}$ for the Dempster-Shafer theory is defined by

$I\mathcal{F}(s) = \mathcal{T}$ when Bel(e) > 0 and $f_{||}$([Bel(a),Pl(a)],[$\text{Bel}_{\Rightarrow}$(a),$\text{Pl}_{\Rightarrow}$(a)]) ≠ †, else $I\mathcal{F}(s) = \mathcal{F}$.

In functional notation, $I\mathcal{F}$: IS → $\{\mathcal{T},\mathcal{F}\}$.

Table B.1 Combination

‖	(1,1)	(0,1)	(0,0)
(1,1)	(1,1)	(1,1)	†
(0,1)	(1,1)	(0,1)	(0,0)
(0,0)	†	(0,0)	(0,0)

B.4 Related research on Dempster-Shafer theory

The Dempster-Shafer theory is still a major field of research in reasoning with uncertainty. We give some references to different approaches of its use that have a common aspect with our application.

In [Hájek, 1992], an approach is demonstrated which is based on an algebraic structure, the ordered Abelian group. An uncertainty model is created for compositional reasoning based on algebraic group theory with the Dempster-Shafer theory. Propagation functions for the logical operators, which are lacking from the theory, are introduced by following a defined ordering on groups. As a result of this construction, the exactness of probability theory is lost, similar to our construction of the model of belief and disbelief in Chapter 4.

Another approach is described in [Provan, 1990], where the Dempster-Shafer theory is embedded in classical propositional logic. Further, the possible use of reason maintenance in combination with the theory is described according to [Laskey, 1989]. In this, a formal equivalence is given between the theory and assumption based truth maintenance supported by probabilities. Belief and plausibility functions are defined in terms of ATMS environments and, therefore, the exponential complexity of the uncertainty calculus is replaced by the exponential complexity of constructing these environments.

Appendix C

Relabelling Algorithms

In Chapters 3 and 5, the aspect of relabelling is discussed in global terms. In this appendix, the algorithms that enable these relabelling processes are outlined. A pseudo language is used to describe the algorithms, their proofs are given informally, by considering three aspects of the algorithms: functionality, termination and complexity. The functionality aspect is concerned with the functional correctness of the algorithm. Termination describes whether the algorithm has a finite execution time, i.e. the algorithm will not enable infinite loops or endless recursion. Finally, the computational complexity of the algorithm is analysed.

C.1 Relabelling while deductive reasoning

The relabelling process in a deductive system consists of a marking and a relabelling procedure.

The marking procedure

In the marking procedure all propositions that have to be relabelled are marked, and ordered in the sense that no proposition will be relabelled until all necessary propositions in its derivation are relabelled.

```
Marking
in: $a_x$ is element of A;
out: $A_{out}$ is array of element of A;
var: $a_i$ is element of A; $A_{in}$ is array of element of A;
begin
  $A_{in} = [a_x]$  $A_{out} = []$
  while $A_{in} \neq []$ do
    take $a_i$ from $A_{in}$
    mark($a_i$)
    add $a_i$ to end $A_{out}$
    for all $j \in J$ with $a_i \in$ Ante(j) do
      if Cons(j) $\in A_{out}$
      then move Cons(j) to end $A_{out}$
      elsif Cons(j) $\notin A_{in}$
      then add Cons(j) to end $A_{in}$
      fi
    od
  od
end.
```

Functionality

The array A_{out} must contain all propositions that must be relabelled. Now assume to the contrary that there is some $a \in A$ that must be relabelled, i.e. there is some $j \in J$ which has a as its consequent and has a relabelled $a_i \in A$ in its antecedent. According to the **for all** statement, used on this proposition a_i, proposition a is either part of A_{out}, or included in A_{in}. In both cases, proposition a will eventually be part of A_{out} and will consequently be relabelled.

The second objective of array A_{out} is the fact that it is ordered in a way that the propositions it contains can be relabelled consecutively. This requires that all propositions in the antecedents of the justifications for a specific proposition have to be relabelled before the specific proposition itself can be relabelled. Observing the marking algorithm, there are two operations defined on A_{out}. Firstly, each marked proposition is added to the end of A_{out}, guaranteeing that each marked proposition will be relabelled. Secondly, within the **if** construct a proposition can be moved to the end of A_{out}. This is done when the consequent of some $j \in J$ is already part of A_{out} but is discovered to be a straight consequence of the proposition (a_i) that is currently investigated. Clearly, this current proposition a_i is now the last element of A_{out}, but has to be relabelled before Cons(j) is relabelled. This is realised by moving the proposition to the end of A_{out}.

Termination

Since deductive cycles are forbidden, each proposition can only be marked once, and the marking procedure automatically terminates within $|A|$ executions of the **while** construct.

Complexity

Clearly, the **while** construct has order $|A|$, the **for all** construct has order $|J|$ and the order of the **if** construct is given by some constant K_m. This gives for the total complexity of the marking procedure, $O(|A|*|J|*K_m)$.

The relabelling procedure

In the relabelling procedure all propositions that have been marked are actually relabelled.

```
Relabelling
in: $A_{in}$ is array of element of A;
var: $a_i$ is element of A; $J_a$ is array of element of J;
begin
  while $A_{in} \neq []$ do
    take $a_i$ from $A_{in}$
    $J_a = []$
    for all $j \in J$ with $a_i = Cons(j)$ do add j to $J_a$ od
    relabel($a_i$,$J_a$)
  od
end.
```

Functionality

The correctness of functionality is obvious. The propositions to be relabelled are given in A_{in} that is equal to A_{out} from the marking procedure. The propositions are ordered in such a way that no proposition will be relabelled until its complete derivation has been relabelled. Therefore, for each proposition, its supporting justifications are collected in J_a. The uncertainty propagation function for the combination can be used to calculate the new uncertainty label for each proposition a_i, based on this set J_a.

Termination

Trivial.

Complexity

Similar to the marking procedure, the **while** construct has order $|A|$ and the **for all** construct has order $|J|$. The computational complexity of the relabelling of a proposition is determined by the uncertainty propagation function that is used for this purpose. This results in a total complexity of the relabelling procedure of $O(|A|*|J|*K_r)$, in which K_r is the complexity of relabelling a single proposition.

C.2 Relabelling while explanatory reasoning

In Chapter 5, the relabelling process of the explanatory reasoning strategy was illustrated. This process also consists of a marking and a relabelling procedure.

The marking procedure

In the marking procedure all propositions are marked that have to be relabelled according to the fact that the proposition that they explain is relabelled. Again, these propositions must be ordered in the sense that no proposition will be relabelled until all necessary propositions in its derivation are relabelled.

Functionality

Array A_{out} contains all propositions that must be relabelled. The only difference with the marking procedure for the deductive interpretation is the addition of the **for all** construct on the consequent of an assumption. This construct is needed to inspect all propositions in an explanation, where in the deductive relabelling process the conclusion can only contain one proposition.

The ordering of array A_{out} enabling the consecutive relabelling of its propositions is realised by exactly the same approach as that used in the deductive marking algorithm.

Termination

Since cycles are forbidden in explanatory reasoning, each proposition can only be marked once, and the marking procedure automatically terminates within $|A|$ executions of the **while** construct.

Marking

```
in: $a_x$ is element of A;
out: $A_{out}$ is array of element of A;
var: $a_i$ is element of A; $A_{in}$ is array of element of A;
begin
  $A_{in} = [a_x]$  $A_{out} = []$
  while $A_{in} \neq []$ do
    take $a_i$ from $A_{in}$
    mark($a_i$)
    add $a_i$ to end $A_{out}$
    for all $u \in U$ with $a_i$ = Ante(u) do
      for all $c_u \in$ Cons(u) do
        if $c_u \in A_{out}$
        then move $c_u$ to end $A_{out}$
        elsif $c_u \notin A_{in}$
        then add $c_u$ to end $A_{in}$
        fi
      od
    od
  od
end.
```

Complexity

Similar to the deductive marking procedure, the **while** construct has order |A| and the **for all** construct has order |U|. Generally, an explanation consists of more than one proposition, therefore, the extra **for all** construct has been added. In principle, the complexity of this construct is |A| but in practice the number of propositions in an explanation will be far fewer than this |A|, say some constant K. This constant is integrated in the order of the **if** construct, some constant K_m. Therefore, the computational complexity of the first marking procedure can be given by $O(|A|*|U|*K_m)$.

The relabelling procedure

In the relabelling procedure all propositions that have been marked by the marking procedure are actually relabelled.

Functionality

Because of the close similarity to the relabelling procedure of the deductive approach, the correctness of the procedure is obvious. Again, the uncertainty propagation function for the combination can be used to calculate the new uncertainty label for each proposition a_i, this time based on the set of relevant assumptions U_a.

Because of the ordering of A_{in}, the working of the procedure is straightforward.

```
Relabelling
in: $A_{in}$ is array of element of A;
var: $a_i$ is element of A; $U_a$ is array of element of U;
begin
    while $A_{in} \neq []$ do
        take $a_i$ from $A_{in}$
        $U_a = []$
        for all $u \in U$ with $a_i \in Cons(u)$ do add u to $U_a$ od
        relabel($a_i$,$U_a$)
    od
end.
```

Termination

Trivial.

Complexity

Obviously, the computational complexity of this relabelling procedure is $O(|A|*|U|*K_r)$, in which K_r is a constant representing the complexity of relabelling a single proposition.

Samenvatting

Het onderzoek naar Artificiële Intelligentie (AI) heeft zich gedurende de afgelopen decennia ontwikkeld tot volwaardig wetenschappelijk onderzoek. Een onderwerp dat hierbij veel belangstelling heeft gekregen is de ontwikkeling van algemene redeneermethoden, welke kunnen worden gebruikt in verschillende applicatiedomeinen. De nadruk van dit proefschrift ligt op een redeneermethode genaamd *compositioneel redeneren*, hetgeen inhoudt dat het redeneerproces is opgebouwd uit elementaire redeneerstappen (de primitieven) die, wanneer in geschikte combinatie gebruikt, complexe redeneerprocessen kunnen vormen. Compositioneel redeneren is de meest gebruikte redeneervorm in bestaande kennissystemen.

De representatie en propagatie (kortweg *management*) van *onzekerheid* in compositioneel redeneren is het thema van dit proefschrift. De ontwikkeling van serieuze AI toepassingen heeft de noodzaak aangetoond van methoden voor het omgaan met onzekerheid in kennissystemen. Onzekerheid kan hierbij gebruikt worden voor het modelleren van verschillende aspecten van kennis: incompleetheid, onnauwkeurigheid, incorrectheid en onbetrouwbaarheid. Dit proefschrift beschrijft een algemeen raamwerk voor het redeneren onder onzekerheid, dat voldoet aan een aantal duidelijk omschreven ontwerpvoorwaarden.

Het algemeen raamwerk

Het ontwikkelen van een algemeen raamwerk voor compositioneel redeneren met onzekerheid is een evidente eerste stap in het onderzoek. Het raamwerk moet algemeen zijn in de zin dat geen veronderstellingen worden gemaakt met betrekking tot de oplossingsstrategie, het type onzekerheid en het toepassingsgebied. Een driewaardige logica, bestaande uit labels *waar*, *onwaar* en *onbekend*, wordt hierbij gebruikt als onderliggend mathematisch fundament, waarbij het label onbekend wordt gebruikt als een expliciete modellering van het incompleetheidsaspect. Een klassieke tweewaardige logica ondersteunt een dergelijke faciliteit niet, terwijl het gebruik van een hogere orde logica hierbij geen extra functionaliteit biedt, maar overigens wel door het raamwerk wordt ondersteund.

Formele definities worden gegeven van de basisstructuren van een kennissysteem: domein-, inferentie-, ondersteunend redeneer- en onzekerheidsmodel. Het domeinmodel beschrijft de representatietaal van het raamwerk. Hiervoor is een hiërarchische benadering gekozen, waarbij kennis in drie lagen wordt gerepresenteerd. Het inferentiemodel legt de voor dit formalisme te gebruiken oplossingsstrategie vast. Het ondersteunend redeneermodel bevat informatie over de tijdens de inferentie uitgevoerde redeneerpatronen. Zoals in de literatuur beschreven, kunnen deze patronen worden gebruikt om gelijkvormige patronen in een redeneerproces te herkennen en er op te anticiperen.

Onzekerheid wordt in het raamwerk geïntroduceerd door de driewaardige logica uit te breiden naar een type dat een onzekerheidsmaat representeert. Specificatie van dit onzekerheidstype vergezeld van een verzameling propagatiefuncties voor de compositionele operatoren, is karakteristiek voor elk onzekerheidsmodel. Het inbedden van dit type in de driewaardige logica resulteert in een fundamentele basis voor onzekerheidsrepresentatie en garandeert de correctheid van onzekerheidspropagatie in de (logische) randgevallen. De aanwezigheid van een ondersteunend redeneermodel opent de mogelijkheid tot het gestructureerd ontwikkelen van propagatiefuncties die de situatie waarin ze gebruikt worden in ogenschouw nemen.

Wanneer het raamwerk wordt gebruikt in combinatie met een deductieve oplossingsstrategie kan worden aangetoond dat alle bestaande deductieve onzekerheidsmodellen een specifiek geval zijn van het raamwerk. In meer algemene zin kan worden geconcludeerd dat het raamwerk de meest algemene formele omschrijving biedt voor het redeneren met onzekerheid in compositionele systemen.

Het model van vertrouwen en wantrouwen

Het *certainty factor model* is het meest bekende en meest gebruikte model voor compositioneel redeneren onder onzekerheid. Analyses die in het verleden zijn uitgevoerd zijn unaniem in hun verwerping van het model, waarbij het schenden van de gesuggereerde probabilistische uitgangspunten door de propagatiefuncties als de voornaamste reden geldt. Het certainty factor model heeft echter een aantal aantrekkelijke kenmerken, zoals de eenvoud van de propagatiefuncties, en heeft bevredigende resultaten opgeleverd in een groot aantal applicaties waarin het is gebruikt. Een gedeelte van het model is daarom nogmaals geanalyseerd, waarbij duidelijk wordt wat de oorzaken zijn van het slechte gedrag van het model in bepaalde situaties.

Het proefschrift beschrijft de ontwikkeling van een nieuw model voor compositioneel redeneren, het model van vertrouwen en wantrouwen, dat geen gebruik maakt van het probabilistische uitgangspunt. Onzekerheid wordt hierbij gerepresenteerd door een twee-dimensionale vector die zowel het geloof in als de twijfel aan een feit weergeeft. Hiervoor zijn propagatiefuncties ontwikkeld, die gebruik maken van informatie over de situatie waarin ze worden gebruikt. Het model is als op zich zelf staand ontwikkeld en pas daarna in het raamwerk ingebed.

De informatie die in het ondersteunend redeneermodel is opgeslagen maakt niet alleen een betere werking van het inferentiemodel mogelijk, het kan ook worden gebruikt bij de constructie van speciale propagatiefuncties. Dit idee is gebruikt bij het ontwikkelen van een verzameling propagatiefuncties die gevoelig zijn voor de situatie waarin ze worden ingezet. Deze functies zijn in staat om een klasse van problemen op te lossen die niet kunnen worden opgelost in bestaande deductieve onzekerheidsmodellen. De berekenbaarheid van deze functies is, hoewel in theorie exponentieel, acceptabel in praktische toepassingen.

Dat het model van vertrouwen en wantrouwen eenvoudig te implementeren is wordt gedemonstreerd aan de hand van de implementatie in een bestaande omgeving voor het ontwikkelen van kennissystemen, Delfi3. Het representatieformalisme van deze omgeving is afgebeeld op het domein- en inferentiemodel van het raamwerk. Een ondersteunend redeneermodel is ontwikkeld voor de implementatie van de propagatiefuncties, en is ook gebruikt voor het verbeteren van de werking van de Delfi3 interpreter. De totale implementatie onderstreept de praktische waarde van de ontwikkelde theorie.

Verklarend redeneren

Bestaande compositionele redeneerstructuren gebruiken een *deductieve* oplossingsstrategie. Het gebruik van *abductief* redeneren staat echter in toenemende mate in de belangstelling binnen het AI onderzoek. Abductie wordt gerepresenteerd in het volgende redeneerschema.

$$\begin{array}{ll} \text{Feit:} & B' \\ \text{Relatie:} & A \rightarrow B \\ \hline \text{Verklaring:} & A' \end{array}$$

Deze redeneervorm wordt veelvuldig gebruikt in het menselijk redeneerproces en maakt een flexibele vorm van redeneren mogelijk waarbij verklaringen kunnen worden verondersteld en later eventueel weer worden afgezwakt of verworpen. Een aantal formalismen is ontwikkeld om deze redeneervorm te ondersteunen, geen van deze formalismen gebruikt echter een compositionele architectuur. Dit proefschrift bespreekt de bruikbaarheid van een abductieve oplossingsstrategie voor compositioneel redeneren.

Als gevolg van de veronderstellende stijl is abductief redeneren moeilijker in het raamwerk te integreren dan een deductieve strategie. Informele definities van abductie mikken op de beste of meest plausibele verklaring(en) voor een verzameling observaties. In dit proefschrift wordt een abductieve oplossingsstrategie beschreven voor compositioneel redeneren, deze strategie wordt *verklarend redeneren* genoemd.

Een verklarende interpretatie is gedefinieerd voor de compositionele redeneerstappen waarin een combinatie van feiten wordt verondersteld een gegeven observatie te hebben veroorzaakt. Twee typen verklaringen worden onderscheiden: convergerende en divergerende verklaringen. In een convergerende verklaring moeten alle feiten tegelijkertijd optreden om de observatie te verklaren. In een divergerende verklaring is het optreden van tenminste één feit voldoende.

Het gebruik van onzekerheid maakt verklarend redeneren nog interessanter. Bij het uitvoeren van een verklarende redeneerstap wordt een initiële verdeling van onzekerheid toegekend aan de feiten in een verklaring. Wanneer extra informatie beschikbaar komt voor feiten in de verklaring, wordt deze initiële verdeling aangepast, rekening houdend met het type van de verklaring. In een divergerende verklaring heeft een toegenomen vertrouwen in een bepaald feit tot gevolg dat het vertrouwen in de resterende feiten afneemt. Dit gedrag wordt *explain away* genoemd. In een convergerende verklaring heeft een toegenomen vertrouwen in een bepaald feit juist tot gevolg dat het vertrouwen in de resterende feiten eveneens toeneemt. Dit gedrag is tegenovergesteld aan explain away en wordt daarom *joint explain* genoemd.

Deze, min of meer informele, beschrijving van verklarend redeneren is formeel beschreven in termen van het ontwikkelde raamwerk. Het model van vertrouwen en wantrouwen is gebruikt om een verzameling propagatiefuncties te ontwikkelen voor verklarend redeneren. De werking van deze functies weerspiegelt de begrippen explain away en joint explain. Echter, het stellen van een aantal restricties is noodzakelijk bij het concept van verklarend redeneren. De meeste restricties zijn niet zwaar, en hebben betrekking op het applicatiedomein. Een belangrijke restrictie daarentegen, het modelleren van bi-directionele invloeden, doet afbreuk aan de capaciteiten van het principe van verklarend redeneren met betrekking tot het modelleren van abductie.

Concluderend kan worden gesteld dat met verklarend redeneren een formalisme is ontwikkeld dat aantrekkelijke karakteristieken heeft waar het gaat om redeneerprocessen (explain away en joint explain, simpele berekenbaarheid). Echter, de expressiviteit van het ontwikkelde formalisme is, door de conceptuele onmogelijkheid om bi-directionele invloeden te modelleren, te gelimiteerd om algemene abductieve redeneerpatronen te ondersteunen.

Het gebruik van compositioneel redeneren in kennissystemen

Zoals elk redeneersysteem heeft de compositionele aanpak sterke en zwakke punten. Voordelen van compositioneel redeneren zijn dat domeinmodellen eenvoudig te construeren zijn, dat inferentie duidelijk gedefinieerd is en dat de berekenbaarheid van propagatiefuncties simpel is. Praktische toepassingen, gebaseerd op (deductieve) compositionele formalismen, hebben bewezen dat zo'n aanpak tot bevredigende resultaten kan leiden.

Restricties aan compositioneel redeneren worden gesteld door het onvermogen om de eerder genoemde bi-directionele invloeden te verwerken en door eisen die aan het applicatiedomein worden gesteld. De slotconclusie is echter dat compositioneel redeneren een aantrekkelijk formalisme blijft om (gedeelten van) een kennissysteem te modelleren. Dit proefschrift geeft een duidelijk overzicht van zowel de theoretische als de praktische randvoorwaarden van het formalisme.

Dankwoord

Het schrijven van een dankwoord bij een proefschrift is verre van gemakkelijk, zeker wanneer de promovendus in meerdere vakgroepen en faculteiten werkzaam is geweest. Op het gevaar af mensen te vergeten volgt hier een kort maar persoonlijk dankwoord.

In de vier jaar dat ik, al dan niet in deeltijd, bij de vakgroep Informatietheorie betrokken ben geweest heb ik mij er altijd zeer thuis gevoeld. Ik bedank alle leden van de vakgroep voor het scheppen van de juiste sfeer en de hulp bij alle kleine en grote problemen. Prof. Eric Backer bedank ik voor de kans die hij mij heeft gegeven dit promotieonderzoek uit te voeren en tot een goed einde te brengen. Discussies met hem waren altijd inspirerend. Jan van der Lubbe heeft, in zijn rol als directe begeleider, grote invloed gehad op het werk. Daarnaast heeft hij, als voortrekker in het METEODES project en de afgelopen tijd als kamergenoot, blijk gegeven van een enorm brede deskundigheid. Van talloze gesprekken en discussies met hem over zeer diverse onderwerpen heb ik veel opgestoken. Erica van de Stadt heeft met haar kritische leeswerk zeker een bijdrage geleverd aan het eindresultaat. Haar, soms zeer originele, inzichten in de materie zullen binnenkort zeker tot uiting komen in haar eigen proefschrift. Cees Witteveen heeft geheel belangeloos zijn licht laten schijnen over de eerste versies van hoofdstuk 2. Tim Bedford voorkwam een fundamentele misstap in het eerste gedeelte van hoofdstuk 4. Tenslotte bedank ik mw. Zaat-Jones voor haar hulp bij het leesbaar maken van mijn Engelse teksten.

Gedurende een periode van ongeveer vijf jaar heb ik, bij de vakgroep Technische Toepassingen van de Informatica, gewerkt in het kader van een tweetal ESPRIT projecten. In die tijd heb ik Henk de Swaan Arons leren kennen als een voortreffelijke baas en collega. Met name wanneer wij vanwege projectverplichtingen in het buitenland verbleven, wisselden scherts en serieuze overpeinzingen elkaar in hoog tempo af. Ik herinner mij dat het onderwerp *stropdas* hierbij lange tijd een rode-draad-functie vervulde. Naast Henk heeft Wim Jonker mij, misschien wel als eerste, geïnspireerd om gericht naar deze promotie toe te werken. Onze samenwerking in Delft resulteerde niet alleen in de ontwikkeling van het vermaarde Delfi3, het bezorgde mij later ook nog een heel plezierig conferentiebezoek aan San Francisco. Een gemeenschappelijke literaire interesse heeft nog regelmatig een (electronisch) postpakketje tot gevolg. Eveneens in ESPRIT verband was Eric-Paul Jansen gedurende twee jaar mijn kamergenoot op de Delftse Julianalaan. Zijn slimme strategieën voor het verbeteren van werkplannen waren stof tot nadenken, discussie en zelfs weddenschappen. Op voetbalgebied bleek maar weer eens dat het hebben van een gemeenschappelijke vijand mensen tot elkaar kan brengen: de rivaliteit tussen onze beide favoriete clubs verbleekte bij de gedeelde vreugde over *weer* een nederlaag voor de Lichtstad.

Tenslotte bedank ik natuurlijk mijn ouders, die altijd het (toekomstig) nut van doorstuderen hebben benadrukt. De laatste jaren hebben ze zich misschien afgevraagd of ze hierin niet te ver waren gegaan, getuige de vraag wanneer ik nu eindelijk eens ging *werken*. Geheel volgens de traditie van dit soort bedankjes sluit de partner van de promovendus de rij. Dat deze er in de praktijk de meeste last van heeft, daar weet Will inmiddels alles van ("Nee, we kunnen niet, Bob moet leren."). Ondanks het feit dat ze het ook op andere vlakken voor de kiezen heeft gehad de laatste twee jaar, heeft ze mij toch die steun kunnen geven die nodig was. Dat dit simpele bedankje hiermee in geen verhouding staat moge duidelijk zijn, ik hoop dan ook dat zij, in de nabije en verre toekomst, nog net zo veel plezier aan dit proefschrift mag beleven als ik.

Curriculum Vitae

Bob Goedhart werd geboren op 11 juni 1963 te Delft en woonde gedurende zijn gehele jeugd in Voorburg. In mei 1981 behaalde hij het VWO diploma aan de openbare Dalton scholengemeenschap Voorburg-Leidschendam. Daarna studeerde hij Wiskunde (5-jarige opleiding) aan de Technische Universiteit Delft, in juni 1988 afgerond in de informatica afstudeervariant. Het afstudeerwerk betrof onderzoek naar en implementatie van een ontwikkelomgeving voor kennissystemen gebaseerd op productieregels.

In september 1988 werd hij aangesteld bij de TU Delft, faculteit der Technische Wiskunde en Informatica, in de functie van toegevoegd onderzoeker verbonden aan ESPRIT[1] project 809 *Advanced Control Systems and Concepts in Small Batch Manufacturing.* Doel van het project was de ontwikkeling van een kennisgestuurd planning- en controlsysteem voor het productieproces van bedrijven in de metaalnijverheid, bij kleine seriegroottes. Partners in dit project waren de Universiteit Twente, ICL (Manchester), Krupp Atlas Datensystemen (Essen) en Morskate Aandrijvingen (Hengelo).

In september 1989 trad hij in dienst van de faculteit Elektrotechniek van de TU Delft als erkend gewetensbezwaarde militaire dienst. In de vakgroep Informatietheorie was hij betrokken bij het METEODES project, betreffende het gebruik van kennissystemen bij de automatische detectie en classificatie van wolken in METEOSAT satellietbeelden. Het project is een samenwerkingsverband van de TU Delft, het Nationaal Lucht- en Ruimtevaartlaboratorium en het Koninklijk Nederlands Meteorologisch Instituut.

Vanaf maart 1991 was hij opnieuw werkzaam als toegevoegd onderzoeker bij de faculteit der Technische Wiskunde en Informatica van de TU Delft. Deze aanstelling was in het kader van het ESPRIT project 2415 *Distributed Manufacturing, Planning and Control.* Doel van het project was de ontwikkeling van een gedistribueerd planning- en controlsysteem voor het productieproces van middelgrote bedrijven. Partners in dit project waren Imperial College (Londen), Atlas Datensystemen (Essen) en Harmonic Drive Antriebstechnik (Limburg a/d Lahn). Vanaf februari 1992 was hij tevens lid van de organisatiecommissie van de Nederlandse AI Conferentie (NAIC) die in november 1992 aan de TU Delft werd gehouden.

Vanaf juli 1992 was hij als Assistent In Opleiding (AIO) werkzaam bij de vakgroep Informatietheorie van de faculteit Elektrotechniek van de TU Delft. Doel van deze laatste aanstelling was het schrijven van het onderhavige proefschrift.

[1] European Strategic Program of Research and development in Information Technology.