

Belief revision with explicit justifications: an exploration in type theory

Citation for published version (APA):

Borghuis, V. A. J., & Nederpelt, R. P. (2000). *Belief revision with explicit justifications: an exploration in type theory*. (Computing science reports; Vol. 0017). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/2000

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Technische Universiteit Eindhoven
Department of Mathematics and Computing Science

Belief Revision with Explicit Justifications:
an Exploration in Type Theory

by

Tijn Borghuis and Rob Nederpelt

00/17

ISSN 0926-4515

All rights reserved

editors: prof.dr. J.C.M. Baeten
prof.dr. P.A.J. Hilbers

Reports are available at:
<http://www.win.tue.nl/win/cs>

Computing Science Reports 00/17
Eindhoven, October 2000

Belief Revision with Explicit Justifications: an Exploration in Type Theory

Tijn Borghuis and Rob Nederpelt

October 27, 2000

Abstract

This paper explores belief revision for belief states in which an agent's beliefs as well as his justifications for these beliefs are explicitly represented. Treating justifications as first-class citizens allows for a deductive perspective on belief revision. We study the belief change operations emerging from this perspective in the setting of typed lambda calculus, and situate these operations with respect to standard approaches.

1 Introduction

An agent who keeps expanding his belief state with new information may reach a stage where his beliefs have become inconsistent, and his belief state has to be adapted to regain consistency. In studying this problem of “belief revision”, the justifications an agent has for his beliefs are not considered to be first-class citizens. At first sight, the two main approaches in literature seem to cover all possible stances on dealing with justifications ([6]): according to “foundations theory” one needs to keep track of the justifications for one's beliefs, propositions that have no justification should not be accepted as beliefs, whereas “coherence theory” holds that one need not consider justifications, what matters is how a belief coheres with the other beliefs that are accepted in the present state. However, there are different ways in which one can “keep track” of justifications. In foundations theory, beliefs are held to be justified by one or several other beliefs (and some beliefs are justified by themselves). In this view, justifications are implicitly present as relations between beliefs, rather than as objects in their own right which are explicitly represented in the formalisation of belief states and belief change operations. In this paper, we explore belief revision for belief states in which justifications are first-class citizens.

Our motivation for investigating belief revision along these lines stems from working on knowledge representation in type theory (more specifically Pure Type Systems,[2]) in the DenK-project. In this project a formal model was made of a specific communication situation, and based on this model human-computer interface was implemented (for a description of the project see [3]). Both in the model and in the system, the belief states of agents were formalised as type theoretical contexts. This means that an agent's beliefs are represented in a binary format, where one part of the expression is the proposition believed by the agent and the other the justification the agent has for this particular belief. Both parts are syntactic objects in their own right, and can be calculated upon by means of the rules of the type theory. This way of representing beliefs turns justifications into first-class citizens, and proved to be very fruitful for the purposes of the project.

At that time mechanisms for belief revision were not investigated or implemented, but it became clear that given this formalisation of belief states there is a straightforward deductive approach to the problem: since every belief is accompanied by its justification (and the rules of the calculus operate on both), every inconsistency that surfaces in the agents belief state has its own (complex) justification containing the justifications of the beliefs that together cause the inconsistency. This makes it easy to identify and remove the “suspects” among the beliefs in the agent’s belief state. Although, technically speaking, this is a direct consequence of the so-called Propositions As Types-principle (as will be explained in sections 2 and 3), this simple idea seems not to have been explored before. We feel that is of a more general interest for two reasons. Firstly, our type theoretical case study shows that explicitly represented justifications have clear advantages: a number of drawbacks traditionally associated with foundational approaches disappear. As such, it may serve as a precursor to a more general account in the setting of Labelled Deductive Systems, of which typed λ -calculi are a simple case. Secondly, it may contribute to a more computational account of belief revision, one which is applicable to agents that have finite information and finite reasoning powers.

In developing the idea, we will come across other well-known issues in this field of research besides the one between coherence and foundations theory mentioned above. For instance the question whether belief states should be taken to be logically closed sets or rather a base set of beliefs which is not closed under logical consequence ([8]), and the question whether an agent should always accept new information (revision versus semi-revision,[10]). In addition, we question a number of assumptions that are traditionally made such as the assumption that an agent has infinite reasoning powers, and that an agent has to solve the revision problem “in splendid isolation”, i.e. without going back to his sources of information via observation and communication.

The paper is structured as follows: we start out by explaining how belief states can be captured in type theory in section 2. To keep the paper self-contained, this section also introduces the (rather minimal) type theoretical apparatus needed. Section 3 shows how type theoretical knowledge states develop as new information becomes available, and gives an informal statement of the problem of revision in type theory. This account of type theoretical revision is formalised in section 4. In sections 5 and 6 we situate our approach with respect to standard approaches from literature, and make a comparison on the level of belief change operations. As it turns out, our revision procedure is particularly close to so-called consolidation operations. This correspondence is worked out in detail in the Appendix. The paper closes with concluding remarks in section 7.

2 Type theory for knowledge representation

This section sets the stage for our account of belief revision with explicit justifications. First the basics of type theory needed for this paper are introduced. Then we give our definition of knowledge and knowledge state, and explain how such knowledge states can be formalized in type theory.

2.1 Type theory

Judgements

The basic relation in type theory is the *judgement*

$$\Gamma \vdash a : T,$$

which can be read as ‘term a has type T in context Γ ’. Here ‘ a ’ and ‘ T ’ are both formulas written according to a well-defined syntax (on the basis of lambda calculus). The expression $a : T$ is called a *statement*, term a is *the subject* of the statement. One also says that term a is an *inhabitant* of type T .

The context Γ is a list of statements with *variables* as subjects, e.g. $x_1 : T_1, \dots, x_n : T_n$. The above judgement can then be read as follows: “If x_1 has type T_1 , ..., and x_n has type T_n , then term a has type T ”. Note that a may *contain* x_1, \dots, x_n , so a *depends on* x_1 to x_n . The set of subject variables $\{x_1, \dots, x_n\}$ is called the *domain* of Γ , or $\text{dom}(\Gamma)$.

Statements

The intuitive notion ‘has type’ has a direct counterpart in naive set theory, viz. ‘is element of’. For example, consider the statement ‘ $a : \mathbf{N}$ ’ (‘term a has type \mathbf{N} ’). Assuming that \mathbf{N} is a symbol representing the set of natural numbers, this statement can immediately be interpreted as ‘ $a \in \mathbf{N}$ ’ (‘the object represented by a is *element of* the naturals’).

The notion of having a type, however, is more general than the notion of set-theoretical elementhood. This is because a type T can represent not only some kind of set, but also a *proposition*. In the latter representation, the statement $a : T$ expresses: ‘ a is (a term representing) a *proof* of the proposition T ’. One speaks of ‘propositions as types and proofs as terms’ (together abbreviated as *PAT*) in order to emphasize this special usage of types.

The advantage of PAT is that proofs belong to the object language, not the meta-language. That is, proofs are ‘first class citizens’ in the syntactical world of type theory. This, combined with the strength of the standard lambda calculus operations, makes type theory a powerful mechanism.

A ‘proof’ is generally considered to be a mathematical notion, but in the PAT-style a proof is anything *justifying* a proposition. This can be a proof in the mathematical sense, but also any other acceptable justification. Let T represent a proposition and let $a : T$. Then:

- If a is an *atomic term* (think of a constant or a variable), then a encodes a justification which cannot be further analysed:
 - It can stand for an axiomatic justification of a proposition: T is an *axiom* and a expresses that the axiom ‘holds’.
 - The validness of proposition T can also come from a *reliable source*. In this case the proof a itself cannot be inspected, but the reliability of the source is enough guarantee to accept the proof. The origin of the knowledge can be any source, either virtual: e.g. a knowledge base, or real: a reliable (community of) person(s).
 - Proposition T can also be justified by *observational evidence*. For example, the proposition that a certain body is yellow can be justified by an atomic term representing the observation that this is the case.

- Finally, proposition T can be an *assumption*. This case is dealt with in type theory by introducing a variable (say x) as an arbitrary (but fresh) inhabitant for the proposition: the statement $x : T$ then expresses: ‘Let x be a proof of T ’. Since x is an unspecified variable, this amounts to: ‘Assume T ’.
- If a is a *composite term*, composed according to the (type-theoretical) syntax, it embodies a complex justification. In this case the precise structure of a expresses how the evidence for T is constructed. For example, under the PAT-interpretation a complete mathematical proof (of a theorem) is coded in one, possibly large, composite term. But also a justification that combines knowledge obtained from observing a certain object with general rules about its behaviour, will lead to a composite term.

Contexts

The context Γ in a judgement $\Gamma \vdash a : T$ contains the ‘prerequisites’ necessary for establishing the statement $a : T$. As mentioned above, a context Γ is a list of statements with variables as subjects, like $x_1 : T_1, \dots, x_n : T_n$.

A context statement $x_i : T_i$ can express several kinds of prerequisites, the simplest being:

1. x_i is an element of the set T_i ,
2. T_i is an assumption (a proposition) and x_i is its atomic justification.

However, in type theory there are different ‘levels’ of typing: a type can have a type itself. Statements expressing the typing of types are concerned with the well-formedness of these types. For the T_i occurring in 1. and 2. such statements have the form:

1. $T_i : \mathbf{set}$, to express that T_i is a well-formed formula representing a set,
2. $T_i : \mathbf{prop}$, to express that T_i is a well-formed formula representing a proposition.

The last-mentioned statements can also be part of a context. So a context could look like: $T_1 : \mathbf{prop}, T_2 : \mathbf{set}, x_1 : T_1, x_2 : T_2$. The terms **set** and **prop** are examples of so-called *sorts*, predefined constants on which the type system is based. Every type system has a specific set of sorts, which we denote by \mathcal{S} .

Note that the statements in the context are *ordered*: first the well-formedness of T_1 and T_2 is established, before their inhabitants x_1 and x_2 are introduced. This is a general principle in contexts: every variable (except the sorts) used in a type must be introduced as the subject of a preceding statement. As a matter of fact, a similar consideration applies to *judgements*: in $\Gamma \vdash a : T$ all variables and constants used in a and T must be introduced as subjects in Γ .

Theories

The PAT-interpretation enables a well-established connection between mathematics and type theory, as has been shown already in the Automath project (see [4]), in which large parts of mathematics have been formalized in type theory: an entire mathematical theory was rendered as a list of judgements. The great importance of such a type-theoretical formalization is that it makes it possible to check whether a given proof of a certain theorem does indeed *prove* the theorem. In fact, it turns out that *syntactical correctness* of the list of judgements is enough to establish the mathematical correctness of the mathematical theory. And the check

on syntactical correctness is relatively easy, since the question whether a certain term is of a certain type in a certain context is decidable. This check on syntactical correctness can be performed by man, but also by a straightforward computer program. In the Automath project, this has already been done with the computer technology of the seventies.

A second advantageous and long-standing connection is the one between logic and type theory. The ‘reasoning power’ of logic finds a very natural counterpart in the operations of lambda calculus underlying type theory. A well-known result is that logics of arbitrarily high order can be expressed in type theory. In the PAT-interpretation of logic, terms capture the full proof *process*: from a proof term one can reconstruct not only which premisses were used in the proof, but also the order in which they were used and the logical rules used to combine them.

Hence, PAT is suitable to express the proof as an object *embodying* its developmental history. As a consequence, type theory embodies an excellent machinery for storing (various kinds of) information, including knowledge. The connection between type theory and knowledge is the subject of the following section.

2.2 Knowledge and type theory

We do not intend to present a philosophical or psychological theory of knowledge, but simply identify three characteristics of knowledge which, according to us, should be taken into account in any attempt to formalize knowledge:

- *Subjectivity* Knowledge is formulated in terms of *concepts*. We assume that these concepts are subjective in the sense that one person may judge something to be an instance of a certain concept, while another person would not recognize this as such. Another aspect of subjectivity is that the knowledge of a person is *partial*: no one knows everything, and persons differ in what they know and don’t know.
- *Justification* Knowledge is justified: persons not only *know* things, but they have *reasons* for knowing them. Generally, parts of knowledge are justified in terms of more basic parts; a person’s body of knowledge is structured. And even atomic justifications are supports for the knowledge, since they point at an origin (an axiom, an observation, etc.; see the previous section).
- *Incrementality* The knowledge of a person can be *extended* as new information becomes available. Whether this information can be incorporated by the person depends on the possibility to tie this information to the knowledge that is already present. This may lead to simply adding the new information, but also to dismissing it (for instance because it is incomprehensible) or even to a reorganization of the existing knowledge.

Under an account of knowledge satisfying these requirements, the traditionally made distinction between knowledge and belief disappears: there can be no knowledge which is true in any absolute sense, since an agent’s knowledge depends on his subjective conceptualisation of the world. At best some pieces of knowledge turn out to be more reliable than others and some things can be agreed upon by more agents than others.

There is a natural way to capture the above characteristics in type theory:

- *Subjectivity is captured by types* Each concept is formalized as a type, each instance of the concept is a term inhabiting this type. A person’s subjective ability to recognize

something as an instance of a concept, is mirrored in the ability to judge that the corresponding term inhabits the corresponding type.

Note that ‘having a concept’ is also subjective in the sense that different people may have formed different concepts in the course of time. This means that one person can have a concept, whereas another person has no comparable concept. And in case persons *do* have comparable concepts, they may differ in what they recognise as belonging to this concept. In case the type formalizing the concept is a ‘set-type’, this means that they may differ in what they regard as elements of the set (a rhododendron may be a tree for the one, but a shrub for the other). In case this type is a ‘proposition-type’, they may differ in what they accept as a justification for that proposition.

- *Justification is captured by terms* As said before, by the PAT-principle, justifications are first-class citizens, formalized in the type-theoretical syntax as terms. The fact that term a justifies proposition T , is expressed as the statement $a : T$. The rules of type theory allows these terms to be combined into complex terms, which reflects that parts of knowledge may be a structured combination of more basic parts of knowledge.
- *Incrementality is captured by contexts* As we will explain below, a person’s knowledge state can be formalized as a type-theoretical context. Addition of new information to the knowledge state can be formalized by adding statements to the context, dismissing information amounts to reducing the context. Information may only be added if it ‘matches’ a person’s knowledge state. Type theory has an innate notion of ‘matching’: a statement can only extend a context if it obeys certain well-formedness restrictions.

2.3 Formalization of the knowledge state

The knowledge *state* of a person consists of ‘everything he knows’ at a certain instant. Given our characterization of knowledge, this means that everything in a knowledge state is formulated in terms of the person’s concepts. This has several aspects:

- *Meaningfulness* A person has formed his own, private concepts, and only things which are formulated by means of these concepts can be meaningful to him. Whether or not information coming from outside (by observation or communication) makes sense, depends on the concepts that are already available. (Throughout this paper we will assume that the entirety of concepts of a person is fixed.)
- *Inhabitation* Whatever a person knows about the world surrounding him is recorded in a knowledge state in the form of meaningful expressions that he accepts. This includes expressions about which objects ‘inhabiting’ the concepts there are in the world and which propositions hold in the world, according to the person.

If we take the following (very simple) context as representing a person’s knowledge states: $T_1 : \mathbf{prop}, T_2 : \mathbf{set}, x_1 : T_1, x_2 : T_2$, we can see:

- *Meaningfulness is captured by statements of the form $T : \mathbf{prop}$ or $T : \mathbf{set}$.* That is to say, in this example the person has two concepts, viz. T_1 , which is a proposition to him, and T_2 , which is a set. (Note that the statements $T_1 : \mathbf{prop}$ by itself does not imply that the proposition T_1 holds according to the person, nor does $T_2 : \mathbf{set}$ imply that the set T_2 is non-empty.) At this stage, there are no other concepts, i.e. all sets and propositions which are not constructed out of T_1 and/or T_2 are not meaningful to him.

- *Inhabitation is captured by statements of the form $x : T$, where T is meaningful* In the example context, the inhabitant x_1 of T_1 represents the person's justification for the holding of T_1 , and the inhabitant x_2 of T_2 is an element of the set T_2 which is recognized as such by the person¹.

'Everything a person knows' at a certain instant can be divided into two categories:

- *Explicit knowledge* is expressed by the statements in the context Γ . These are explicitly represented pieces of knowledge which are directly available to the person.
- *Implicit knowledge* is expressed by statements *derivable* on the context Γ . These are consequences of a person's explicit knowledge which he can obtain by doing inferences.

Hence, in a judgement of the form $\Gamma \vdash a : T$, the explicit knowledge can be found to the left of the symbol \vdash , and the implicit knowledge to the right of \vdash .

Note that the knowledge state is not deductively closed, i.e. deriving consequences requires 'work', which is reflected in the construction of a compound justification a for T . Such a construction is a derivation using the rules of type theory; it consists of a *sequence* of judgements of which the just-mentioned compound justification is the final one. We come back to this in the next section.

In order to derive *all* consequences of his explicit knowledge, a person would have to be able to perform possibly infinite derivations. Since this is not feasible, we assume that there is a certain 'bound' on the derivation depth.

3 Development of the knowledge state

The knowledge state of a person is not static. As time goes by, new information comes to the person's attention and has to be dealt with. With the conception of knowledge states as type-theoretical contexts in mind, as explained in the previous section, we distinguish several stages in the treatment of new information by a person, marked by decisions which the person has to make. We describe these stages below.

Meaningfulness

In the first stage, the meaningfulness of the new information is at stake. New information may or may not be meaningful to a person depending on his current knowledge state. Type-theoretically, new information manifests itself in the form of a (sequence of) statement(s). Whether these statements are meaningful with respect to a knowledge state, can be syntactically decided. In section 2.3 we noted that type theory has an intrinsic notion of meaningfulness. Below we explain how this notion can be extended to statements of the form $x : T$, expressing the inhabitation of a proposition or set T .

We presuppose that a person only processes new information that is meaningful (makes sense) to him, i.e. meaningful with respect to his current knowledge state, and that he *decides* to dismiss this information otherwise. (In a communication setting, we expect the person to search for clarification, either by questioning his dialogue partner, or by (re-)inspecting his environment.)

¹Syntactically, x_1 and x_2 are *variables*. However, as we see later, each of these 'variables' may in fact be a *defined constant*, abbreviating a term which codes all details of the justification.

Expanding the knowledge state

If the information *is* meaningful, the person adds the new information provisionally to the knowledge state: Γ is extended to e.g. $\Gamma_1 \equiv \Gamma, y_1 : T_1, y_2 : T_2$.

The resulting knowledge state can turn out to be consistent, that is to say, the person cannot construct a term M such that $\Gamma_1 \vdash M : \perp$, where \perp is falsum (the logical constant ‘falsity’). As we explain below, we assume that the person has a limited deductive power, so he can only construct terms by derivations up to a certain length. Intuitively this means that the person has a ‘horizon’ behind which he cannot see the consequences of his knowledge state. Hence, the person’s notion of ‘consistency’ is bound by his horizon. (As a consequence, a knowledge state can be inconsistent without the person being able to find this out at the current point in time.)

If the obtained knowledge state does not give contradictions *within the horizon*, then Γ_1 is accepted as the new context.

Revising the knowledge state

There is, however, also the possibility that the person has found an inconsistency, i.e. he has constructed in his newly expanded knowledge state some term M such that $\Gamma_1 \vdash M : \perp$. In that case, he can decide to reject the new information and return to the previous knowledge state. But he can also decide to revise his knowledge state (including the new information) in order to restore consistency. (The person may actually be able to construct more than one inhabitant of falsum; we assume that he concentrates on one of these.) The most natural thing to do, is to find one or more statements in the context representing his knowledge state, which enabled the construction of M . These statement can be located in the ‘old’ context, but also in the newly added piece of context, or in both. By removing one or more of these statements from his context, consistency may be regained, since this particular proof of falsum, M , cannot be constructed any more. Below we explain this in more detail: we propose a syntactical iterative procedure which restores consistency. (In general, there is more than one way to regain consistency by removing statements from the knowledge state.)

The stages and decisions we distinguished above, are not intended to capture actual cognitive processes, but merely to state as clearly as possible which aspects of belief revision we do and do not consider in our formalization. For instance, the fact that the person decides which statements to remove, means that this is not decided by the formalism, in other words, we do not postulate so-called *epistemic entrenchment*. (For a comparison with standard theories of belief revision, see section 5.)

In sections 3.1 and 6.3 we discuss the various stages of dealing with information as explained just now, in more detail. We give special attention to the representation in Type Theory.

3.1 Adding information

The knowledge state of a person changes as new information becomes available to him. Since knowledge states are modeled by type-theoretical contexts, this means that contexts should change accordingly. In this subsection we demonstrate that type theory has the possibility to accommodate a simple form of such a change in the knowledge state, viz. the addition of new information to the knowledge state.

Adding information to a type-theoretical context amounts to adding statements to this context. This does not mean that arbitrary information may be added, addition is subject to syntactical restrictions. We discuss this below, distinguishing between the addition of information originating from inside and from outside the knowledge state of the person.

Adding information from inside

A person is able to reason with his knowledge. For example, let us assume that the statements $A \rightarrow B : \text{prop}$ and $A : \text{prop}$ are meaningful to the person. Moreover, let us assume that the person has justifications for both propositions, since $A \rightarrow B$ and A are inhabited (e.g. $x : A \rightarrow B$ and $y : A$ occur in the context Γ representing his knowledge state). Then the person can infer that B holds, as well, expressed by the statement $xy : B$. This is the case since one of the rules in type theory is the so-called *application rule*, which in this specific instance looks like:

$$\frac{\Gamma \vdash x : A \rightarrow B \quad \Gamma \vdash y : A}{\Gamma \vdash xy : B}$$

This inference allows the person to combine his justification x for $A \rightarrow B$ with his justification y for A into a complex justification xy (pronounced as ‘ x applied to y ’) for the proposition B . (See [2].)

We do not treat all the rules of type theory in detail in this paper. We only mention that there are no more than a small number of rules, which are all like the above rule in that they enable to derive a new judgement from one or more judgements which are given or derived earlier.

The judgement $\Gamma \vdash xy : B$ resulting from the person’s inference as explained above, shows that the person is able to construct a justification for B on his knowledge state Γ . However, the statement $xy : B$ is not yet part of his knowledge state. To incorporate this statement, it would simply be sufficient to append it to Γ . However, for technical reasons only statements with variables as subject are allowed in the context. In order to circumvent this (technical) problem, we expand our notion of ‘context’ as described above, by allowing also a new kind of statements, called *definitions*, in the context. A definition is a statement of the form $z := E : T$, expressing that z is a name for the term E of type T . The new name z is the *subject* of the definition $z := E : T$. Formally, z is a *variable*. (This is in contrast with the good habit of calling such a defined name a *constant*.) By means of definitions, complex justifications can be abbreviated and recorded in the context. This definition mechanism is essential in the practical use of type theory for the formalization of ‘bodies of knowledge’, as has been shown e.g. in the Automath project ([4]).

A definition $z := E : T$ may be added to a context Δ whenever z is fresh with respect to Δ and $E : T$ is derivable on Δ . In the example above, this enables the person to record the inferred $xy : B$ in his knowledge state by adding the definition $u := xy : B$, using some fresh variable u . Hence, the context Γ has evolved into the context $\Gamma, u := xy : B$, reflecting the development of the person’s knowledge state brought about by his reasoning. The proposition B (and its justification), which was implicit knowledge of the person (since it occurred at the right hand side of the \vdash), has now become explicit knowledge.

From a purely logical point of view, it may seem that adding a derived proposition to the knowledge state (making it *explicit*) does not contribute to the person’s *implicit* knowledge. However, this is not the case since we assume that there is a bound to the depth of derivations

a person can perform. Under this assumption, the implicit knowledge is limited: it consists of everything a person can derive on his context *within a certain number of derivation steps*. As soon as the explicit knowledge has grown, in general there is more that can be derived by the person in the same number of steps, so the implicit knowledge has grown as well: the person's 'deductive horizon' has broadened.

Adding information from outside

The knowledge state of a person can change by reasoning (which he does himself, from the *inside*), but it can also change by information originating from the *outside*. For the latter there are two important knowledge sources: observational and communicational.

- *Observation* A person can *recognize an object* (visually, or by any other sensory perception) in his world as belonging to a certain *set*. For example, he sees an object which he characterizes as being a ball. But he can also *obtain evidence* for *propositions* by looking at the outside world. For example, he sees that the ball is yellow.

In both cases, the new information can be added to the context of the person by the addition of a new statement with a fresh *atomic* subject, acting as the justification. The atomic character of this justification is caused by the impossibility to decompose the observation into smaller parts.

The two observations in the example above could e.g. be combined into the context extension $b : ball, o : yellow b$.

- *Communication* Another manner in which a person can change his knowledge state is by information passed to him by another person. Again, this information can involve (the existence of) objects as well as (the holding of) propositions.

For this communication it is necessary that both persons share a language in which they communicate. We assume that each person speaking this language has a mapping between the words of the language and the subjective concepts present in his knowledge state, and vice versa. In [1] a type theoretical model of communication is developed based on this assumption. In this model, the types in a person's knowledge state are communicable via the (mappings to) the common language, but the inhabitants of these types (justifications) are not. Hence the contents of a communication take the form of a (sequence of) statement(s) of which the subjects are *atomic*, since the original justifications of the 'sender' are not communicable to the 'receiver'.

Example: in a situation after the observation of the previous example, the utterance 'The yellow ball is hollow' can lead to the following extension of the person's context: $c : hollow b$, provided that 'hollow' is a concept known to the person, and he is able to correctly match the definite description to the objects b and o in his context.

Hence, be it either observation or communication, the information to be added to a person's context has the form of a sequence of statements with atomic subjects. However, as we said earlier, the types of the statements in the context give rise to a notion of meaningfulness. Only types 'constructable' from the statements already present in the context of a person are meaningful to him. This restricts the addition of statements originating from the outside.

Technically, this has the following form. Let Γ be the original context of the person and assume that the sequence $x_1 : T_1, \dots, x_n : T_n$ is the information from the outside (with

fresh subjects x_1, \dots, x_n). Then these statements are added one by one, thus changing the knowledge state incrementally. That is to say, for each $1 \leq i \leq n$, the statement $x_i : T_i$ may only be added if

$$\Gamma, x_1 : T_1, \dots, x_{i-1} : T_{i-1} \vdash T_i : s$$

with $s \equiv \text{set}$ or $s \equiv \text{prop}$. In other words, a statement may only be added if its type is well-formed with respect to the current knowledge state. This shows, as we said before, that new information (a sequence of statements) can only be absorbed in a step-by-step fashion (statement by statement), where the possibility to append a new statement *depends on* the information available in the context at that stage, i.e. the original context *plus* the already appended statements.

This embodies precisely the notion of incrementality, discussed in subsection 2.2, which not only applies to the case of only one ‘chunk’ of information from the outside (i.e. one sequence of statements) as above, but also to subsequent additions of such chunks of information. For instance, if a person is in a dialogue with another person, each new utterance he receives will be added only if it is meaningful against the background of the utterances accepted before.

Technical note: In treating observation and communication, we extended the use of type theory as it is traditionally described in the literature: one usually does not take into account that information can come from outside the context. When type theory is applied to knowledge representation, one usually models (the progress of) a solitary reasoning person, who can only extend his knowledge from the inside. However, since we adopted the same well-formedness criteria as usual to adding information from the outside, the resulting context in our extension will always be syntactically correct with respect to the original type-theoretical standards. Hence, this extension of the use of type theory does not lead to an extension of the formalism. (Even the complete process of adding information from the outside can be justified in type-theoretical sense. We will not go into that here.)

3.2 The problem of revision

As we saw in the previous section, a situation in which a person has to revise his knowledge state can be characterized as follows. The person is confronted with new information (which is meaningful to him), and decides to accept it. When it turns out that the incorporation of this new information leads to inconsistency of the resulting knowledge state, the person has to remove information from this new knowledge state to restore consistency. Below we describe how this can be done by means of type theory.

Revision from a type-theoretical perspective

The need for revision can originate both from the inside and from the outside. We begin by describing the situation where new information is added from outside.

Suppose that the context Γ represents the person’s current knowledge state (which is consistent within his horizon) and the sequence $x_1 : T_1, \dots, x_n : T_n$ represents the new information from the outside. Hence the resulting context is $\Gamma_1 \equiv \Gamma, x_1 : T_1, \dots, x_n : T_n$. The inconsistency of Γ_1 manifests itself in the existence of an inhabitant of falsity which the person can construct within his horizon: there is an M such that $\Gamma_1 \vdash M : \perp$. There may be more than one such an inhabitant, but for the moment we assume that the person has chosen one of these. (We come back to this in section 4.)

The fact that all justifications are explicitly present enables the person to identify all ‘suspects’: the beliefs in Γ_1 that together cause the inconsistency. Since M embodies a derivation of falsity in the sense explained earlier, we find in M the justifications of all beliefs that are part of this derivation (M contains the full developmental history of the derivation!). The suspect justifications occur as free variables in M , since these free variables point exactly at the premisses of the derivation of falsity: such a premiss $x : T$ gives rise to a free x in M . This is a property of the proposition as types interpretation of type theory. Moreover, the rules of type theory ensure that all free variables of M occur as subjects in Γ_1 .

We give an example to make this clear. Let $A : \text{prop}$ and $B : \text{prop}$ be statements belonging to the knowledge state (the context) and assume that the person has proofs of A , of $A \rightarrow B$ and of $\neg B$ (abbreviating $B \rightarrow \perp$). This is represented in the knowledge state by statements say $x : A$, $y : A \rightarrow B$ and $z : \neg B$. The rules of Type Theory then enable the derivations of first $\Gamma \vdash yx : B$ and second $\Gamma \vdash z(yx) : \perp$. Now the free variables x , y and z in the ‘proof object’ $z(yx)$ point precisely at the propositions A , $A \rightarrow B$ and $\neg B$, which together enable the construction of the inconsistency.

Note that, given the consistency of Γ , there have to be free variables in M which occur as subjects in the *new* information $x_1 : T_1, \dots, x_n : T_n$. (Otherwise, $M : \perp$ could already be constructed on Γ itself; this is a consequence of the so-called Strengthening Lemma of type theory. See also section 4.2.)

New information can also originate from the inside, when a person adds a derived consequence to his knowledge state by means of a definition. This broadens his horizon and hence contradictions which were previously out of sight can now come into view. (See section 3.1) I.e. suppose Γ is consistent and $\Gamma \vdash N : P$ within the horizon. The result of adding $N : P$ to Γ by means of a definition is $\Gamma' \equiv \Gamma, u := N : P$. Now it is possible that there exists an M such that $\Gamma' \vdash M : \perp$ within the new horizon. As above, this M contains inhabitants of all ‘suspects’ as its free variables.

This shows that there is, technically speaking, no difference at all between revision due to information from outside and revision due to information from inside. Intuitively it may seem strange that a person can be forced to revise his knowledge state by only adding a consequence of what he already knows to his knowledge state, that is to say without any external reason. However, if we take the idea of limited deductive power seriously, this is inevitable.

Restoring consistency by removing information

In the situation described above (i.e. there is an M such that $\Gamma_1 \vdash M : \perp$), the person can try to regain consistency by removing one or more of the ‘suspects’ from Γ_1 , being some of the statements $x_i : T_i$ occurring in Γ_1 where x_i occurs free in M . As we pointed out before, we assume that *the person decides* which statements he chooses to remove. Before making this choice, the person probably reconsiders the suspects, with the help of new observations or communications with others.

However, it is generally not sufficient to simply erase the chosen suspects from the knowledge state, since there may be beliefs depending on the ‘suspect’ beliefs. Such a dependent belief should be removed as well, since it is no longer meaningful on the knowledge state from which the suspect(s) have been erased.

A belief can depend upon another belief in two ways. First, a belief B may contain a free variable x which is the subject variable of a statement $x : A$ preceding $y : B$ in the context.

Second, if $x : A$ precedes a definition statement $z := E : C$, both E and C may contain such a free variable x . In these cases, $y : B$ and $z := E : C$ depend on x for their well-formedness. Hence, removal of $x : A$ from the context has consequences for these statements as well. The most natural solution is to remove them.

There is a relatively simple, syntactical procedure for removing suspect beliefs *and* the beliefs depending on them, which we describe in section 4.1. The result of this procedure is a new knowledge state, Γ_2 . It is, however, not necessarily the case that this Γ_2 is consistent within the person's horizon. Although the justification M of falsity is no longer constructable on Γ_2 , there may have been more than one justification for falsity on Γ_1 . Some of these justifications of falsity may still be constructible on Γ_2 . In that case, the person chooses one of these justifications and selects a new set of suspects on which the procedure described above is repeated. Iteration leads to a sequence of knowledge states Γ_1, \dots which is finite, since in every iteration step at least one of the (finite number of) justifications of falsity is removed. So there is a final knowledge state Γ_n , on which no justifications of falsity are constructable. Hence, Γ_n is consistent within the person's horizon. This Γ_n is then the resulting *revised* knowledge state.

4 Belief revision

In this section we give a formal description of the process of belief revision in type theory, as described above. First we define the syntactical procedure for removing 'suspect' beliefs and the beliefs depending on them (section 4.1). Next we state some properties of this removal procedure (section 4.2). Finally, we discuss the full revision procedure, which may involve iterative removal of suspect beliefs, and we investigate the properties of the procedure.

4.1 The removal operation

We start with a knowledge state represented by a context Γ and new information represented by the sequence $x_1 : T_1, \dots, x_n : T_n$. We add the new knowledge to the original knowledge state, obtaining $\Gamma_1 \equiv \Gamma, x_1 : T_1, \dots, x_n : T_n$. We assume that this 'new' context Γ_1 turns out to be inconsistent and we assume that the person has chosen one or more suspect beliefs in Γ_1 which he wants to remove. Note the assumption that the suspect beliefs can be found in the *entire* Γ_1 , so also among the new information: contrary to standard accounts of belief revision we do not award a special priority to the new information (this point will be discussed in section 6.3).

The removal operation that we describe below results in the transformation of Γ_1 into a new context Γ_2 . However, as we discuss below, regaining consistency may involve more than one such transformation, hence in our definition we define the transformation as leading from Γ_i to Γ_{i+1} .

In order to give a general definition of removal, we write a context as if all statements in the context were definitions: $y_1 := E_1 : T_1, \dots, y_m := E_m : T_m$, with the convention that $y_l := E_l : T_l$ must be read as $y_l : T_l$ if it is not a definition and we take $FV(E_l) = \emptyset$ in the last mentioned case. ($FV(M)$ is the set of all variables occurring free in M .)

We assume that V is the set of variables which are the subjects of suspect beliefs $y_k := E_k : T_k$ in Γ_i which the person has chosen to remove. As we explained at the end of section 3, also beliefs $y_l := E_l : T_l$ *depending* on the variables in V must be removed. Below we characterize the set $\text{dep}_\Gamma(V)$ consisting of V plus all subject variables of statements depending on V .

We start with the definition of the notion ‘subcontext’.

Definition 1 Let $\Gamma \equiv \Delta_1, y := E : T, \Delta_2$ and $\Gamma' \equiv \Delta_1, \Delta_2$. Then $\Gamma' \subset \Gamma$. The relation \subseteq is the reflexive and transitive closure of \subset . If $\Gamma_1 \subseteq \Gamma_2$ we say that Γ_1 is a subcontext of Γ_2 .

Next we define the dependency relation \leq , a partial order between subject variables of a context Γ .

Definition 2 Let $\Gamma \equiv \Delta_1, y := E : T, \Delta_2$. Then $\text{def}_\Gamma(y) = E$, $\text{type}_\Gamma(y) = T$ and $\text{stat}_\Gamma(y) = (y := E : T)$. For y and $z \in \text{dom}(\Gamma)$ we say that $y < z$ if $y \in \text{FV}(\text{def}_\Gamma(z) \cup \text{type}_\Gamma(z))$. (For convenience, we write ‘ $<$ ’ instead of $<_\Gamma$.) The relation \leq is the reflexive and transitive closure of $<$. The set $\text{dep}_\Gamma(y)$ is $\{z \in \text{dom}(\Gamma) \mid y \leq z\}$. Moreover, $\text{dep}_\Gamma(V)$ is $\bigcup_{y \in V} \text{dep}_\Gamma(y)$, for $V \subseteq \text{dom}(\Gamma)$.

Note that the set of variables depending on a set of variables V , includes V itself.

Next, we define a deletion operator del , erasing statements from a context, and the removal operator ‘ \setminus ’.

Definition 3 For domain variable y of $\Gamma \equiv \Delta_1, y := E : T, \Delta_2$, we define $\Gamma - \text{stat}_\Gamma(y)$ as Δ_1, Δ_2 . For a set W of domain variables of Γ , we define $\text{del}_\Gamma(W)$ as $\Gamma - \bigcup_{y \in W} \text{stat}_\Gamma(y)$. For a context Γ and a set $V \subseteq \text{dom}(\Gamma)$, the removal operation ‘ \setminus ’ is defined by $\Gamma \setminus V = \text{del}_\Gamma(\text{dep}_\Gamma(V))$.

So, $\Gamma \setminus V$ is the context resulting from removing all statements depending on the set V of chosen subject variables, from Γ .

4.2 Properties of the removal operator

As we explained in section 3.1, knowledge states are incremental, in the sense that the type of each statement should be meaningful given the statements preceding that statement. In type theory this is expressed by the notion *legality*:

Definition 4 A context $\Gamma \equiv y_1 := E_1 : T_1, \dots, y_n := E_n : T_n$ is legal, if for all $1 \leq i \leq n$: $y_1 := E_1 : T_1, \dots, y_{i-1} := E_{i-1} : T_{i-1} \vdash T_i : s$ for some $s \in \mathcal{S}$ and moreover, if $E_i \neq \emptyset$, then: $y_1 := E_1 : T_1, \dots, y_{i-1} := E_{i-1} : T_{i-1} \vdash E_i : T_i$.

The removal operator applied to a legal context, results in a new, legal subcontext:

Lemma 1 Let Γ be a context and $V \subseteq \text{dom}(\Gamma)$. Then $\Gamma \setminus V \subseteq \Gamma$. Moreover, if Γ is legal, then $\Gamma \setminus V$ is legal.

PROOF For the second part: Subsequently delete all $\text{stat}(y)$ for $y \in \text{dep}_\Gamma(V)$ from Γ , from right to left, using the Strengthening Lemma²:

For $\Gamma_1, y := E : T, \Gamma_2$ a legal context and M and B terms: if $\Gamma_1, y := E : T, \Gamma_2 \vdash M : B$ and $y \notin \text{FV}(\Gamma_2) \cup \text{FV}(M) \cup \text{FV}(B)$, then $\Gamma_1, \Gamma_2 \vdash M : B$.

The removal operator has the nice property that the result of subsequent applications to V and W is the same as applying it in the reverse order, or by applying it to the union of V and W :

²Lemma due to Geuvers, Nederhof and Van Benthem Jutting, see [5], pp. 74.

Lemma 2 For a legal context Γ and subsets V and W of $\text{dom}(\Gamma)$:
 $(\Gamma \setminus V) \setminus W = (\Gamma \setminus W) \setminus V = \Gamma \setminus (V \cup W)$.

PROOF By the definition of \setminus and simple set theory.

4.3 The revision procedure

In this section we show how the removal operator can be used to regain consistency. We assume that a person has originally a legal and *consistent* knowledge state Γ . He extends his context Γ with new information $x_1 : T_1, \dots, x_n : T_n$, obtaining $\Gamma_1 \equiv \Gamma, x_1 : T_1, \dots, x_n : T_n$. Let's assume that the extended context is legal again, but the extension makes his context inconsistent: he can now construct an M such that $\Gamma_1 \vdash M : \perp$.

We consider several cases.

- Consider the case that the person chooses to remove a *single* subject variable z occurring freely in M , plus all statements depending on this z . Hence, he obtains $\Gamma_2 \equiv \Gamma_1 \setminus \{z\}$ as his new context.

Note that the chosen variable z may be the inhabitant of a statement in the original context Γ or of a statement $x_i := E_i : T_i$ which is part of the extension. In the latter case, $\text{dep}_{\Gamma_1}(z)$ contains only variables occurring as subjects in the extension. In the former case, however, $\text{dep}_{\Gamma_1}(z)$ may contain subject variables of Γ *as well as* subject variables of the extension. Hence, the removal operation may change the new information in *both* cases.

- If the person chooses a non-empty *set* V of variables occurring freely in M , then he obtains $\Gamma_2 \equiv \Gamma_1 \setminus V$ as his new context. Note that the removal of V has the same effect as removing the separate elements of V , one by one, in *any* order. This is a consequence of lemma 2. (This also holds if V is the set of *all* free variables in M .)

In either of these cases, the proof M of falsity is no longer derivable on the resulting context Γ_2 . However, this does not guarantee that Γ_2 is consistent: it may be the case that the person can still construct a proof of falsity, say M' , on Γ_2 . Then the person can repeat the removal operation with one or more free variables occurring in M' , and so on. Thus he obtains a sequence of contexts $\Gamma_1, \Gamma_2, \dots$, where each Γ_{i+1} is a legal subcontext of Γ being properly 'smaller' (i.e. contains fewer statements) than Γ_i . It follows that the sequence $\Gamma_1, \Gamma_2, \dots$ is finite, so that the person finally obtains a context Γ_n which is consistent. (In the extreme case $\Gamma_n = \varepsilon$, but there is no proof of falsity on the empty context ε by the consistency of type theory.)

This implies:

Lemma 3 *The iterated application of the removal operation terminates and results in a consistent knowledge state.*

In other words, it is a *revision procedure*.

It is interesting to note that this iteration can be summarized in a single application of the removal operation: Let's call the non-empty set of variables that the person chooses to remove in the transition from Γ_i to Γ_{i+1} , V_i (which can be a singleton set). Then $\Gamma_{i+1} = \Gamma_i \setminus V_i$. However:

Lemma 4 *Successively removing V_i from Γ_i for $i = 1, \dots, n - 1$, leads to the same result as removing the union of all V_i s from Γ_1 :*

$$\Gamma_n = \Gamma_1 \setminus \bigcup_{i=1}^{n-1} V_i.$$

PROOF *This is again a consequence of lemma 2.*

In this section we assumed that it is the *person* who makes the decision about which statements to remove, and not the formalism. We gave arguments for this point of view in section 3. However, in comparing our system with systems in the literature we will briefly discuss formal *heuristics* for making these decisions (see section 6.4).

4.4 Revision with horizon

In the previous subsection we assumed that the person is ‘omniscient’ in the sense that he is able to provide a proof of falsity at any time, if there *exists* one. This, of course, is not realistic. For this reason we introduced in the beginning of section 3 the notion of ‘horizon’ for the person. If we look at the revision procedure, the presence of a horizon has important consequences.

Firstly, a knowledge state Γ has only a limited (finite) number of consequences within a given horizon. We formulate this as a theorem, provable by combinatorial arguments:

Theorem 1 *Given a context Γ and a number h limiting the derivation depth of derivations on Γ (‘the distance to the horizon’), there is a finite number of statements derivable on Γ (modulo α -conversion).*

Note that we do *not* consider the full deductive closure of Γ , which possibly corresponds with an ‘infinite horizon’, which is no horizon at all.

For convenience, we denote the finite set of derivable statements from context Γ (the set of consequences of Γ) within horizon distance h by $Conseq_h(\Gamma)$.

Corollary 1 *Given a context Γ that is inconsistent within horizon distance h , there is a finite number of inhabitants of falsity (‘proofs of falsity’) (modulo α -conversion). I.e., there are finitely many terms M such that $M : \perp \in Conseq_h(\Gamma)$.*

By application of the revision procedure, statements are removed from the context Γ . This will eliminate a (number of) proof(s) of falsity, but the question arises whether there are *new* proofs of falsity on the revised (smaller) context. This is not the case:

Theorem 2 *If $\Gamma \setminus V$ is the result of revising Γ with respect to V , there is no statement derivable within horizon distance h on $\Gamma \setminus V$ which was not already derivable within horizon distance h on Γ . I.e., $Conseq_h(\Gamma \setminus V) \subseteq Conseq_h(\Gamma)$.*

PROOF *Note that $\Gamma \setminus V \subseteq \Gamma$, i.e. every statement occurring in $\Gamma \setminus V$ also occurs in Γ , by lemma 1. For any two PTS-contexts Δ and Δ' the so-called Thinning Lemma holds ([5], Lemma 4.4.24): if $\Delta' \subseteq \Delta$ and $\Delta' \vdash A : B$, then $\Delta \vdash A : B$. Hence if $\Gamma \setminus V \vdash A : B$ then $\Gamma \vdash A : B$. However, if we regard the horizon distance, it might still be possible that there exists a statement $A : B$ which is derivable on $\Gamma \setminus V$ in at most h steps, and on Γ in more than h steps (due to extra steps needed to ‘retrieve’ the premisses on the larger context). We assume, however, that the axiomatization of Type Theory is such that the Start-rule allows any number of Weakenings. In that case, a derivation of $\Gamma \setminus V \vdash A : B$ can always be ‘copied’ into a derivation of $\Gamma \vdash A : B$ with the same number of derivation steps.*

Corollary 2 *The removal procedure does not allow the introduction of new proofs of falsity.*

Corollaries 1 and 2 imply the following theorem, which says that we can always reach a consistent context *in one revision step* (albeit that this appears to be a rather crude one):

Theorem 3 *Given an inconsistent context Γ and a horizon distance h , there exists a set of variables V such that $\Gamma \setminus V$ is consistent within the same horizon distance.*

PROOF *Take V to be the set of all free variables occurring in all proofs of falsity which can be derived on Γ within horizon distance h . By Corollary 1, this set is finite and by the definition of the revision procedure, none of these proofs of falsity are constructable on $\Gamma \setminus V$. By Corollary 2, there are no new proofs of falsity on $\Gamma \setminus V$, hence $\Gamma \setminus V$ is consistent within horizon distance h .*

5 Situating our approach

In the previous sections we presented an approach to belief revision based on type theory. As far as we know, this approach is new. In the setting of type theory, justifications of beliefs are ‘first class citizens’, which is not the case in current approaches to belief revisions.

In this section we discuss the relations between our approach and approaches from the literature which are well-known. We take the Handbook-article of Gärdenfors and Rott ([7]) as our guideline for this discussion.

5.1 Belief bases with justifications

Following the methodological taxonomy of [7], our approach has the following characteristics:

- Beliefs are represented as statements in type theory, a person’s belief state as a type-theoretical context (section 3). The result of a belief change operation is again a type-theoretical context (section 4.3).
- The statements that are elements of the context representing a person’s belief state, represent the *explicit* beliefs of the person. Beliefs that can be derived from these statements are his implicit beliefs (section 2.3). Contrary to standard practice, we assume that the deductive powers of the person are limited: a person has a deductive horizon and only statements that are derivable within this horizon count as his implicit beliefs.
- Our theory does not prescribe how choices are made concerning what beliefs to retract. It provides a set of candidates for retraction, but leaves the actual choice to the person (section 4.3). At best, we can give heuristics for this choice (see section 6.4).

Gärdenfors and Rott mention four *integrity constraints* guiding the construction of belief revision formalism:

- *The beliefs in the data base should be kept consistent whenever possible.* We adhere to this constraint, with the annotation that we take ‘consistent’ to mean: ‘consistent with respect to the limited deductive powers of the person’.

- *If the beliefs in the data base logically entail a sentence, then this sentence should be included in the data base ('deductive closure').* It will be clear from our earlier comments (sections 3 and 4.4) that we do not subscribe to this point of view. However, it is possible to explicitly include a derived belief (to be precise: derived within the person's horizon) in the knowledge state by means of a definition (section 3.1).
- *The amount of information lost in a belief change should be kept minimal.* In accordance with the fact that our theory says nothing about extra-logical factors governing the choice of beliefs-to-be-retracted, there is no notion of minimality inherent in our theory.
- *In so far as some beliefs are considered more important or entrenched than others one should retract the least important ones.* In line with our previous comment, a notion of extra-logical preference like entrenchment should in our opinion not be part of a theory. Preferences like entrenchment belong, again, to the realm of heuristics.

The choices we made above imply that we work with so-called *belief bases*: the knowledge state of a person is represented by a finite set of sentences, a context Γ . The belief set of the person consists of his explicit beliefs (statements in Γ) and his implicit beliefs (statements derivable on Γ within the horizon, i.e. $Conseq_h(\Gamma)$). Note that $\Gamma \subseteq Conseq_h(\Gamma)$: every explicit belief in the context Γ is derivable on Γ , and is hence *also* implicit. Therefore we can represent a person's belief set by $Conseq_h(\Gamma)$.

Since we choose to represent justifications for beliefs explicitly, as inhabitants, in the knowledge state, our approach is closely related to what is called *Foundations Theory* in the literature, see e.g. [6].

5.2 The relation with Foundations Theory

Foundations Theory is based on the principle that belief revision should consist in giving up all beliefs that do no longer have a satisfactory justification, and in adding new beliefs that have become justified. This principle has a number of consequences:

- *Disbelief propagation* If in revising a knowledge state a certain belief is retracted, not only this belief should be given up, but also all beliefs depending on this belief for their justification. Since our theory has an explicit representation of justifications, this propagation can be captured syntactically, as was shown in definition 2, by means of the relation \leq . Hence, our approach does not have the drawbacks that are often associated with disbelief propagation, viz. 'chain reactions' and 'severe bookkeeping problems'.
- *Non-circularity.* Since beliefs can depend on other beliefs for their justification, we should be careful that the dependency graph is well-founded, i.e. does not contain circularities. In our approach such circularities cannot occur, since they are ruled out by the well-formedness requirements for the type-theoretical contexts (section 3).
- *Multiple justifications.* A belief may be supported by several independent beliefs. The removal of one of those justifications does not automatically lead to giving up the belief. This characteristic is reflected in our approach, where a belief may have more than one inhabitant. Suppose that a person has two justifications for the belief that A holds on his knowledge state Γ , for example: $\Gamma \vdash M : A$ and $\Gamma \vdash N : A$. Since the free variable sets of M and N may be disjoint, it may be possible to retract the justification M of A , while retaining N and hence the belief that A (see section 4.3).

There is a well-known problem in Foundations Theory, following from the hypothesis that all beliefs must have a justification. This induces a distinction between beliefs: some beliefs are justified by one or more other beliefs, but there must also exist beliefs which are justified ‘by themselves’. These so-called *foundational beliefs* are considered to be ‘self-evident’, they need no further justification.

In Foundations Theory, justification is a relation on the level of the beliefs. In type theory, however, justifications are explicitly represented by terms *inhabiting* the beliefs they justify. The distinction between foundational and other beliefs is reflected in type theory in the structure of the term inhabiting the belief:

- *Atomic justifications.* If the term inhabiting the belief is a constant or a variable, the justification cannot be further analyzed. This corresponds to the foundational beliefs, but only to a certain extent: it does not imply that these beliefs are necessarily self-evident. The atomic justification simply reflects the person’s decision to adopt the belief in its own right, e.g. on the basis of an observation, communication or an act of will. (See also subsection 2.1)
- *Composite justifications.* If the term inhabiting the belief is a composite term, the justification can be analyzed according to the structure of the term. These terms occur in the context in *definitions*, e.g. in the statement $y := E : T$, where E is a composite justification for T . One can find the inhabitants of the other beliefs supporting the belief that T , as the free variables occurring in E .

Thus the justification relation from Foundations Theory becomes a relation between inhabitants of beliefs in type theory. This relation is captured by the dependency relation \leq of definition 2.

6 Comparing operations for belief change

Before we can compare the formal properties of our revision procedure with those described in the literature, we must formulate our equivalents of the three standard belief change operations: expansion, contraction and revision.

- *Expansion:* Adding a new sentence A to the belief base K , regardless of the consistency of the resulting belief base. The result is usually denoted by $K + A$.

In our type-theoretical setting, expansion is just addition of either a statement or a definition to the context: Γ changes into $\Gamma, x : A$ (with x fresh), or into $\Gamma, x := M : A$. In the first case new information originating from outside is added, in the second case a consequence of the belief base is made explicit by adding it to the context.

Note that, in both cases, the type A must already be well-formed with respect to Γ , i.e. $\Gamma \vdash A : s$ with s a sort in the set of sorts \mathcal{S} of the type system (cf. section 2.1). In the second case, $x := M : A$ may only be added when $\Gamma \vdash M : A$ is derivable. This again gives a well-formedness guarantee.

NOTATION: The type-theoretical analogue of Expansion will be denoted by $Exp_{x:=M:A}(\Gamma; \Gamma')$ if the expansion of Γ with the statement or definition $x := M : A$ yields Γ' . Hence, $\Gamma' \equiv \Gamma, x := M : A$.

- *Contraction*: Retracting some sentence A from the belief base K , as well as sentences depending on A (without adding new beliefs). This is denoted by $K \ominus A$.

In type theory, retracting has to be done with statements instead of formulas. Moreover, given a context Γ and a horizon depth h , there can be several terms inhabiting a belief A that is to be retracted. There is a set of terms t such that $t : A \in \text{Conseq}_h(\Gamma)$. If we take retraction to mean that *no* statement $M : A$ should be derivable any more, we need a retraction procedure similar to the one described in section 4.3. That is, the person iteratively chooses variables occurring free in such terms t inhabiting A and removes them from Γ , in order to eliminate evidence for A .

Formally, we can say that there is a set $V_A := FV\{t \mid t : A \in \text{Conseq}_h(\Gamma)\}$. The variables chosen by the person together constitute a subset V of V_A (cf. Lemma 4). Retraction of A with respect to Γ then amounts to a removal $\Gamma \setminus V$ with V chosen such that $\neg \exists_t (t : A \in \text{Conseq}_h(\Gamma \setminus V))$.

Note: In its generality, this procedure always gives the desired result. There is, however, a slight complication: there are sentences which we never want to be contracted, for example tautologies. How we can prevent in type theory that this kind of sentences can be retracted, is discussed in section 6.2.

NOTATION: The type-theoretical analogue of Contraction is denoted by $\text{Ctr}_A(\Gamma; \Gamma')$, if Γ' is the result of contracting Γ with respect to A . In case $A \notin \text{Conseq}_h(\Gamma)$, we take Γ' to be Γ .

- *Revision*: Adding a new sentence A to the belief base K while maintaining consistency, by (possibly) deleting a number of sentences in K . This is denoted by $K * A$.

In the standard account, revision is related to contraction and expansion by means of the so-called Levi-identity: $K * A = (K \ominus \neg A) + A$. This implies, that for belief bases revision can be defined as a two step procedure:

1. *Contract* by $\neg A$
2. *Expand* by A

We can match this so-called internal revision ([10]) by means of the two typetheoretical operations defined above:

1. $\text{Ctr}_{\neg A}(\Gamma; \Gamma')$
2. $\text{Exp}_{x:=M:A}(\Gamma'; \Gamma'')$

Note that this procedure will always lead to a context (Γ'') containing the new information ($x := M : A$), whereas the procedure described in sections 4.3 and 4.4 did not, since there it was possible that (parts of) the new information were removed as well, if this information contributed to the inconsistency. In literature, this alternative approach is known as ‘semi-revision’. In section 6.3 we will show that the typetheoretical version of revision developed in this paper closely resembles the semi-revision operation *consolidation* of [10]. Anticipating on this, we introduce the following.

NOTATION: The type-theoretical analogue of Revision (i.e., Contraction by $\neg A$ and Expansion by A) is denoted by $\text{Rev}_{x:=M:A}(\Gamma; \Gamma')$, if Γ' is the result of revising Γ with respect to $x := M : A$.

Finally we note that the operations of expansion and contraction, and hence revision, described above can also be executed with new information consisting of a *sequence* of statements $(x_1 := M_1 : A_1, \dots, x_i := M_i : A_i)$, rather than a single statement $(x := M : A)$. From a type-theoretical point of view, this is a natural generalization. Moreover, experiences obtained in formalizing the addition of outside-information (as described in section 3.1) to type-theoretical knowledge states, suggests that such information generally takes the form of a sequence of statements.

Now we have given our equivalents of the standard belief change operations, expansion, contraction and revision, we give a more detailed comparison between the two approaches in order to position our approach with respect to the literature. We concentrate on the results of Gärdenfors ([7]) and Hansson ([10]).

6.1 Expansion

Since expansion is not problematic, neither in the standard approach, nor in the type-theoretical analogue, there is no reason to compare these two approaches in more detail.

6.2 Contraction

We now look at the rationality postulates for contraction as they are reformulated for belief bases in [7]. As already remarked earlier, our approach is more fine-grained than that of Gärdenfors, because we deal with specific proofs of propositions, whereas the standard approach does only considers (sets of) propositions. Hence, when Gärdenfors contracts with respect to a proposition A , from our perspective, he implicitly quantifies over *all* proofs of A . This difference also plays a role in the formulation of the postulates themselves.

In some of the Gärdenfors postulates, conditions occur of the form $\vdash A$ and $\not\vdash A$. Type-theoretically, we take these to state that there *exists* respectively *doesn't exist* a proof object for the type A within the horizon. Moreover, the fact that A is or isn't a tautology, suggests that this proof object can (or cannot) be constructed on the empty context ε . However, in type theory the type A itself must be well-defined before we can think about the construction of inhabitants of A . Hence, we need some *initial context* Γ_{init} which ensures the well-definedness of all propositions: $\vdash A$ is translated into $\exists_M(\Gamma_{init} \vdash M : A)$ and $\not\vdash A$ into $\neg\exists_M(\Gamma_{init} \vdash M : A)$.

Of course, statements in the initial context should not be contracted in a revision process, since this initial context acts as a kind of 'axiom base' for the well-definedness of the propositions. The contraction procedure $Ctr_A(\Gamma; \Gamma')$, as described above, will not consider variables inside Γ_{init} , since the statements of Γ_{init} are at the wrong level of typing to have their subjects appear in terms inhabiting propositions (cf. section 2.1).

Note that if A is a tautology, there exists a proof object in which no free variables occur: $\exists_M(\Gamma_{init} \vdash M : A)$ where $V_A = \emptyset$. Since M cannot be blocked by removing variables in V_A from the context, we cannot contract over tautologies. On the one hand this is a good thing: one does not want to lose tautologies. On the other hand, this has as a consequence that Contraction becomes a *partial* operation, which may be unsuccessful!

Below we present the Gärdenfors postulates for belief bases as given in [7], followed by their type-theoretical translation and a discussion of their validity. The original postulates quantify over all sentences A and belief sets H , their translations over all types A and contexts Γ (where $\Gamma \supseteq \Gamma_{init}$). In addition, the postulates are stated using $Cn(H)$, the deductively closed set of

consequences of H (i.e. with infinite horizon). In the translation of e.g. Gärdenfors's ($H^\circ 3$)-postulate we take $A \notin Cn(H)$ to mean that there exists no proof object of type A (within the horizon) on the person's context, $\neg\exists_N(N : A \in Conseq_h(\Gamma))$.

($H^\circ 1$) $H^\circ A$ is a belief set.

Its translation is:

If $Ctr_A(\Gamma; \Gamma')$, then Γ' is a well-formed context.

This holds: Assume $Ctr_A(\Gamma; \Gamma')$, then there exists some set $V \subseteq V_A$, possibly empty, such that $\Gamma \setminus V \equiv \Gamma'$. By Lemma 1, Γ' is a well-formed context.

($H^\circ 2$) $H^\circ A \subseteq H$.

Its translation is:

If $Ctr_A(\Gamma; \Gamma')$, then $\Gamma' \subseteq \Gamma$.

This follows from the definition of the removal-operation (definition 2).

($H^\circ 3$) If $A \notin Cn(H)$, then $H^\circ A = H$.

Its translation is:

If $\neg\exists_N(N : A \in Conseq_h(\Gamma))$ and $Ctr_A(\Gamma; \Gamma')$, then $\Gamma \equiv \Gamma'$.

This holds: Assume $\neg\exists_N(N : A \in Conseq_h(\Gamma))$ and $Ctr_A(\Gamma; \Gamma')$ and suppose $\Gamma \not\equiv \Gamma'$. Then (see $H^\circ 2$) Γ' is a proper subcontext of Γ . Hence there is some variable z occurring in Γ as a subject, such that $z \in V$, where V is the set of variables chosen to be removed and $z \in V$ not in Γ' . Hence z must have occurred free in some term N such that $\Gamma \vdash N : A$ within the horizon, but then $\exists_N(N : A \in Conseq_h(\Gamma))$. Contradiction.

($H^\circ 4$) If $\not\vdash A$, then $A \notin Cn(H^\circ A)$.

Its translation is:

If $Ctr_A(\Gamma; \Gamma')$, then $\neg\exists_M(M : A \in Conseq_h(\Gamma'))$.

This postulate holds by our definition of contraction.

Note that the condition $\not\vdash A$ (' A ' is not a tautology) is implicitly present in our translation, because this is implied by the condition $Ctr_A(\Gamma; \Gamma')$. In fact, if A is a tautology, then A has a proof object, but this proof object has no free variables. Therefore the set V_A is empty and hence Contraction of A as described before is not possible (there is no Γ' such that $Ctr_A(\Gamma; \Gamma')$).

($H^\circ 5$) $H \subseteq (H^\circ A) + A$.

Its translation is:

If $Ctr_A(\Gamma; \Gamma')$, then $\Gamma \subseteq \Gamma', z : A$.

Note that we have to add a proof object z for A . We could not use a definition $z := M : A$, since this implies that $\Gamma' \vdash M : A$ for some M , which contradicts $Ctr_A(\Gamma; \Gamma')$.

This postulate, which has a controversial status in the literature (in fact: base contractions generally violate it), does not hold here. A simple counterexample is the following: Take $\Gamma \equiv \Gamma_{init}, x : B \rightarrow A, y : B \vdash xy : A$, then $Ctr_A(\Gamma, \Gamma')$, where $\Gamma' \equiv \Gamma_{init}$, but $\Gamma \not\subseteq \Gamma_{init}, z : A$.

($H^{\circ}6$) If $\vdash A \Leftrightarrow B$, then $H^{\circ}A = H^{\circ}B$.

Its translation is:

If $\exists_N(\Gamma_{init} \vdash N : A \Leftrightarrow B)$ and $Ctr_A(\Gamma; \Gamma')$ and $Ctr_B(\Gamma; \Gamma'')$, then $\Gamma' \equiv \Gamma''$.

This postulate does not hold in general, but there is a case in which it holds, as we explain below.

First, observe that in type theory we have to do work to transform proofs of A into proofs of B (and vice versa) by means of the proof N of the equivalence of A and B which contains subproofs N_1 for $A \rightarrow B$ and N_2 for $B \rightarrow A$. Then for example: If $\Gamma \vdash M : A$ for some M , then $\Gamma \vdash N_1 M : B$ (and vice versa).

We call M a *direct* proof of A and $N_1 M$ an *indirect* proof of B . Note that transforming a direct proof of A into an indirect proof of B involves one extra proof step. Hence, this can lead to a situation in which the direct proof is within the horizon, whereas the indirect proof is not.

Disregarding this horizon problem, the postulate still does not hold in general: in order to block all proofs of B , all proofs of A also have to be blocked. Hence, a set V will have to be chosen which is a subset of the union of the variables occurring free in all proofs of A and all proofs of B , i.e., $V \subseteq (V_A \cup V_B)$. However, it might still be possible to find different subsets V_1 and V_2 which both block all proofs of A and B .

Example: $\Gamma \equiv \Gamma_{init}, x : C \rightarrow A, y : C, z : D \rightarrow B, u : D$, and $\Gamma \vdash N : A \Leftrightarrow B$. Then $V_A = V_B = \{x, y, z, u\}$. Now take $V_1 = \{x, z\}$ and $V_2 = \{y, u\}$. It is easy to check that both V_1 and V_2 block all proofs of A and B . If we take $\Gamma' \equiv \Gamma \setminus V_1$ and $\Gamma'' \equiv \Gamma \setminus V_2$, then $Ctr_A(\Gamma; \Gamma')$ and $Ctr_B(\Gamma; \Gamma'')$, but $\Gamma' \not\equiv \Gamma''$.

However, the postulate *does* hold if we use the ‘safe contraction’ described in section 6.4, i.e. take $V_1 = V_2 = V_A = V_B$, then $\Gamma' \equiv \Gamma''$.

Here we end our discussion of the basic postulates $H^{\circ}1$ to $H^{\circ}6$ for base contraction. There exist two more (non-basic) postulates, $H^{\circ}7$ and $H^{\circ}8$, concerning conjunctive formulas $A \wedge B$. We do not discuss those here for two reasons: as remarked above, the type-theoretical notion of contraction can easily be generalized to a sequence of statements, so that there is no need to give a special status to the \wedge -connective; moreover, it would require us to go into the technical details of coding conjunction in type theory, which does not serve the purpose of this paper.

Concluding, as in most approaches to base revision in the literature, postulates $H^{\circ}1$ through $H^{\circ}4$ are satisfied in the type-theoretical translation, but $H^{\circ}5$ does not hold. In addition, the type-theoretical equivalent of ‘safe contraction’ satisfies $H^{\circ}6$. This exactly reflects Theorem 5.4.1 of [7].

6.3 Revision

In the standard account of revising a belief base K with new information A , the new information is always accepted and beliefs in K are abandoned to maintain consistency. Objections have been raised to this account, on the grounds that too much priority is given to new information ([10]): at each stage, new information is completely trusted. However, this complete trust is only temporary: once the new information is incorporated in the belief base, it is

itself susceptible to abandonment when in the next stage even newer information becomes available. This seems awkward.

This paper has no fundamentally new objections to add, but the emphasis on novelty has a number of additional undesired consequences from our point of view. Firstly, the new information always has to be accepted as a whole, whereas in our approach it is a possible outcome of revision that the person accepts only part of the new information. The standard account is also too absolute in another respect: because of the unlimited deductive power assumed in this approach, the person can detect beforehand whether a piece of new information is inconsistent with his current belief base, and hence whether revision should be carried out. Under the more realistic assumption of the deductive horizon, it is not possible to do this consistency check once and for all: inconsistencies, and hence the need for revision, may arise as proofs of falsity turn up inside the horizon. Finally, thinking of standard belief revision in the setting of communication, a person would be forced to accept every utterance by his dialogue partner(s), even if accommodating this information requires a major reconstruction of his own belief base. Therefore, new information and information in the belief base should be treated equally by the revision operation.

Revision procedures which do not necessarily accept the new information are known in literature as *semi-revision* procedures. For belief bases, semi-revision can be specified as a two-stage procedure ([10]):

1. *Expand by A*
2. *Make the belief base consistent by deleting either A or some original belief(s)*

Compared to the revision procedure formulated at the beginning of section 6, the order of the steps is reversed³ and the second step has been modified. The operation performed in the second stage is called *consolidation*, [10], and can be carried out by contracting over falsehood. In our approach, the procedure looks like this:

1. $Exp_{x:=M:A}(\Gamma; \Gamma')$
2. $Ctr_{\perp}(\Gamma'; \Gamma'')$

In other words, revision and contraction are related by the following identity:

$$Rev_{x:=M:A}(\Gamma; \Gamma') = Ctr_{\perp}(\Gamma, x := M : A; \Gamma')$$

This is exactly the revision procedure described earlier in sections 4.3 and 4.4. First the new information, one or more statements, is added to the context Γ , then a number of statements from the expanded context is removed to block the construction of inhabitants of falsity.

There is a close resemblance between our revision procedure and the one described in [10], called *kernel consolidation*. This correspondence is given in the Appendix of this report.

³Reversing the order alone yields *external revision*, [10]

6.4 Heuristics

What we have done so far does not add up to a theory of belief revision in the traditional sense. We have shown how a person can find the suspect beliefs when his belief state has become inconsistent, and how he can remove a number of the suspects to regain consistency, but our revision procedure does not tell the person *which* suspects to remove. Standard approaches have a selection mechanism which embodies some notion of “rational choice” between the various possibilities for revision in any given situation. These mechanisms usually introduce extra-logical structure in the belief state, and are computationally unwieldy. The underlying view is that of a solitary reasoner who has to solve the inconsistency in splendid isolation, using his infinite reasoning powers and looking only at the beliefs in his (infinite) belief state. Our concern is with agents who have finite belief states (including justifications), finite computational resources, and who have access to the world by means of observation and communication. Such agents have possibilities to (re)evaluate the various suspects, by performing observations/tests or by communicating with other agents, and a theory of belief revision cannot and should not prescribe how they make their choices. Strategies used by an agent to make these choices are not part of the theory, if they can be captured formally they could be used as heuristics on top of the theory. In this section, we briefly discuss how some selection mechanisms from standard approaches mentioned in [7] fit into our account as heuristic principles.

In so-called (*partial*) *meet contraction*, the idea is that the optimal contraction or revision is the one that requires the smallest number of insertions and/or deletions in the belief state. This criterion can be applied in the type theoretical approach. Given one particular proof of inconsistency, $\Gamma \vdash M : \perp$, removing any one of the statements of which the subjects occur free in M is sufficient to block this particular proof. However, these statements may have different numbers of statements depending on them in Γ , and so one could prefer to remove the statement with the least number of dependents to minimise the deletions from the belief state. In cases where more than one proof of falsity has to be blocked, a “blocking” subset has to be chosen from the set of all variables occurring free in these proofs. When there is more than one subset that does the job, one could again prefer the subset with the smallest number of statements (possibly taking the number of dependent statements into account).

As in the standard approach, this criterion will not always yield a single optimal solution. It is possible to end up with two or more minimal sets of statements whose removal will restore consistency. To overcome this indeterminism, additional structure is introduced in the belief state. The central idea in this construction is known as *epistemic entrenchment*: “not all sentences that are believed to be true are equal value for planning of problem-solving purposes, but certain pieces of knowledge and beliefs about world are more important than others when planning future actions conducting scientific investigations or reasoning in general” ([6]). In performing contraction or revision, the beliefs that are given up should be the ones with the lowest degree of epistemic entrenchment. Although in our opinion such an ordering of epistemic entrenchment of the beliefs in the belief state cannot be given once and for all independent of the current goals and activities of the agent performing the contraction or revision, such an ordering could in principle be added to the context representing the agent’s belief state. Note that the imposed entrenchment ordering has to respect the dependency relations between the beliefs in the context: if a belief $y := N : B$ depends on a belief $x := M : A$, then $y := N : B$ should not be epistemically more entrenched than $x := M : A$ since removing $x := M : A$ without removing $y := N : B$ will result in a context which is not

wellformed.

Another idea that can be applied, at least in spirit, in the type theoretical setting is that of *safe contraction*: a proposition B is safe with respect to a proposition A if it cannot be blamed for the derivability of A . To contract over A , all propositions that are not safe with respect to A have to be removed. There is an obvious way to translate this idea to our approach to revision: a belief $x := M : A$ is safe if it cannot be blamed for the fact that a proof object for \perp can be constructed on the belief state Γ . The simplest interpretation of “being to blame” for a statement in context would be “to have its subject appear as a free variable in a proof object for \perp ”. Hence the simplest form of safe contraction would be to remove all statements of which the subjects appear free in a proof object for \perp and their dependents from the context. However, this does not suffice if all statements that are removed themselves depend upon earlier statements in context, since the proof object for \perp could be rebuilt from these “ancestors”. One way around this problem, is to use the construction of a so-called kernel set described in the Appendix. For a given derivation horizon and a given context, this construction inductively builds the set of minimal falsity implying subsets of statements in Γ . This kernel set can reasonably be said to contain all statements that are “to blame” for the inconsistency of the context (within the horizon), hence we can define safe contraction as the removal of all these statements and their dependents. Although this will yield a unique solution, it will usually not be minimal in terms of the number of statements that are removed.

7 Concluding remarks

In this paper we explored the use of explicitly represented justifications in belief revision. Starting from the representation of beliefs as type theoretical statements and belief states as type theoretical contexts, we showed that the presence of justifications makes it easy to identify the beliefs that cause inconsistency of the belief state (section 3.2). Their presence also greatly simplifies the handling of dependencies between beliefs (section 4.1). With respect to literature, our initial assumptions put us in the area of foundations theory for belief bases. However, our account does not suffer from the drawbacks usually associated with foundations theory such as problems with disbelief propagation, circular justifications, and multiple justifications for the same belief (section 5.2). The operation of belief revision that naturally arises from our approach is one of semi-revision: new information is not automatically completely trusted (section 6.3).

The fact that our approach is deductive, and that we do not require that our theory of belief revision itself selects which beliefs have to be removed, makes its applicable to agents with limited computational resources (section 6.4 and Appendix). This holds independently of the strength of the logic in which the belief change operations are cast: the mechanisms that were used to represent justifications and dependency relations between beliefs are at the heart of type theory, making our approach applicable to a large family of type systems. Given the well established connections between type theory and logic, this means it is applicable in a wide range of logics. For instance, it can be applied in each of the Pure Type Systems from the well-known Logic Cube (see [2] and [5]), which corresponds to logics ranging from minimal propositional logic to higher order predicate logic.

Obviously many questions remain unanswered in this case study, and are left for future research. Its purpose was to establish that it is worthwhile to consider justifications explicitly

in the study of belief change operations.

References

- [1] Ahn, R., Borghuis, T., *Communication Modelling and Context-Dependent Interpretation: an Integrated Approach*. In: Types for Proofs and Programs: International Workshop, TYPES'98, Kloster Irsee, Germany, March 1998, Selected Papers, Altenkirch, T., Naraschewski, W., Reus, B. (eds.), LNCS 1657, Springer Verlag (1999), pp. 19 – 32.
- [2] Barendregt, H., Lambda calculi with types. In *Handbook of logic in computer science*, Abramsky, Gabbay and Maibaum (eds.), Oxford University Press, Oxford (1992), pp. 117 – 309.
- [3] Bunt, H., Ahn, R., Beun, R-J., Borghuis, T., and Van Overveld, K., *Multimodal Cooperation with the DenK System*. In: Multimodal Human-Computer Interaction, Bunt, H., Beun, R-J., Borghuis, T. (eds.), Lecture Notes in Artificial Intelligence 1374, Springer Verlag (1998), pp. 39 – 67
- [4] Nederpelt, R.P., Geuvers, J.H., De Vrijer, R.C. (Eds.), *Selected Papers on Automath*, Studies in Logic and the Foundations of Mathematics 133, North-Holland, Amsterdam (1994).
- [5] Geuvers, H., *Logics and type systems*. Ph.D. thesis, University of Nijmegen (1993).
- [6] Gärdenfors, P., *The dynamics of belief systems: Foundations versus coherence theories*, revue Internationale de Philosophie, 44 (1990), pp. 24 – 46.
- [7] Gärdenfors, P., Roth, H., *Belief Revision*. In: Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 4: Epistemic and Temporal Reasoning, Gabbay, D., Hogger, C., Robinson, J. (eds.), Clarendon Press (1995), pp. 36 – 119.
- [8] Hansson, S., *In defense of base contraction*, Synthese 91 (1992), pp. 239 – 245.
- [9] Hansson, S., *Taking belief bases seriously*. In: Logic and the Philosophy of Science in Uppsala, Prawitz, D. and Westerståhl, D. (eds.), Kluwer Academic Publishers (1994), pp. 13 – 28.
- [10] Hansson, S., *Semi-revision*, Journal of Applied Non-Classical Logics, Volume 7, nr. 1-2 (1997), pp. 151 – 175.

Appendix: kernel consolidation

Our revision procedure is particularly close to what Hansson calls *kernel consolidation*. This form of consolidation is based on the idea that a subset of sentences in the knowledge base K implies falsity if and only if this subset contains some minimal falsity-implying subset of K . Hence the consistency of K can be restored by removing at least one element of each minimal falsity-implying subset of K . Minimal falsity-implying subsets are called *kernels*, they are defined as follows.

Definition 5 *A subset X of sentences from a belief base K is a kernel if and only if*

1. $X \subseteq K$
2. $\perp \in \text{Cn}(X)$, and
3. If $Y \subset X$, then $\perp \notin \text{Cn}(Y)$

The set of all kernels of K is called the kernel set, denoted by $K \amalg \perp$.

The sentences of K that have to be discarded to restore consistency, are selected by a so-called *incision function*:

Definition 6 *An incision function σ for K is a function such that:*

1. $\sigma(K \amalg \perp) \subseteq \cup(K \amalg \perp)$
2. If $X \in (K \amalg \perp)$, then $X \cap \sigma(K \amalg \perp) \neq \emptyset$

Definition 7 *Let σ be an incision function for K . The kernel consolidation \approx_σ for K is defined as follows:*

$$K \approx_\sigma \perp = K \setminus \sigma(K \amalg \perp)$$

In the typetheoretical approach, falsity-implying subsets of the context Γ are sets of statements of which the subjects occur free in a proof object inhabiting \perp , i.e. $\{\text{stat}_\Gamma(y) \mid y \in FV(M)\}$, where M is a term such that $\Gamma \vdash M : \perp$. If we call this set of statements for a given proof object M ‘ S^M ’, (‘suspects’ in M), we can see that this set fulfils the first two criteria for kernels given in Definition 5:

1. $S^M \subseteq \Gamma$
2. $\Gamma_{init}, S^M \vdash M : \perp$, that is: \perp is a consequence of S^M (where Γ_{init} contains the well-typedness information needed for the derivation)

However, such a falsity-implying subset S^M is not necessarily minimal in the sense required for kernels (the third criterion): there may exist another proof object N such that $\Gamma \vdash N : \perp$ and $S^N \subset S^M$. This is due to the fact that proof objects code an entire proof for the proposition represented by their type, including proofs that contain ‘detours’, sequences of steps that could have been omitted in the proof. Such detours can invoke premises that are not really needed to prove the proposition, resulting in non-minimal subsets. A very simple example of such a situation is the following: take $\Gamma \equiv \Gamma_{init}, x : A, z : A \rightarrow A, y : A \rightarrow \perp$, then there are at least two proof objects inhabiting falsity, $\Gamma \vdash y(zx) : \perp$ and $\Gamma \vdash yx : \perp$. Clearly,

the falsity-implying subset for the first proof object is not minimal, the second proof object is constructed without using $z : A \rightarrow A$. Although in typed λ -calculus some detours can be eliminated by performing reductions on proof objects⁴, we cannot in general prevent a person from having a belief state on which non-minimal proofs of falsity can be derived.

Moreover, in discussing the minimality of falsity-implying subsets, the limited deductive powers of the person have to be taken into account. Since the person can only construct proofs of $\leq h$ steps, where h is the horizon distance, we can at best talk about falsity-implying subsets which are minimal with respect to these proofs. Given a subset S^M for some inhabitant M of falsity, there may exist a set S^N such that $S^N \subset S^M$ where the proof object N for falsity cannot be constructed within the horizon h . Hence, this smaller set S^N should not be considered by the selection procedure.

The assumption of horizon enables an inductive procedure for the constructing the kernel set $\Gamma \amalg^h \perp$, the set of all minimal falsity-implying subsets within the horizon. For a given context Γ , one systematically generates all derivations of length zero, then all derivations of length 1, then all derivations of length 2, \dots , up to all derivations of length h . Among each layer of derivations, one picks out all derivations of an inhabitant of falsity. By comparing the sets of free variables of these inhabitants, the minimal falsity-implying subsets for that layer can be found, i.e. for the i -th layer ($1 \leq i \leq h$) all $FV(M)$ such that $\Gamma \vdash^i M : \perp$, and there is no N such that $\Gamma \vdash^i N : \perp$ and $FV(N) \subset FV(M)$. The sets S^M that are minimal for a layer are then added to the kernel set $\Gamma \amalg^i \perp$ if there is no S^N already in $\Gamma \amalg^i \perp$ such that $S^N \subset S^M$. In other words, before adding the sets that are minimal in a layer it is checked whether they are also minimal with respect to sets from previous layers.

Given the inductively constructed kernel set $\Gamma \amalg^h \perp$, the type theoretical analogons of *incision function* and *kernel consolidation* can be defined exactly as given in Definitions 6 and 7, but for the replacement of $K \amalg \perp$ by $\Gamma \amalg^h \perp$. Note that in the newly attained definition the slash in $\Gamma \setminus \sigma(\Gamma \amalg^h \perp)$ stands for the type theoretical removal operation defined in section 4.1, rather than the standard set theoretical operation in definition 6, i.e. not only the statements selected by the incision function ($\sigma(\Gamma \amalg^h \perp)$) are removed from Γ but also all statements depending on them ($\text{dep}_\Gamma(\sigma(\Gamma \amalg^h \perp))$). Since dependencies are not considered in the setting of Hansson, we need to be able to distinguish between those two kinds of statements. The notion of 'independence' can easily be defined as follows:

Definition 8 *A statement $x := M : A$ is an independent member of the set of statements Δ iff there does not exist a statement $z := N : B \in \Delta$ such that $x \in \text{dep}_\Delta(z)$.*

In [10], kernel consolidation is characterised by a theorem which links its construction to a number of postulates. We restate this theorem here for type theoretical knowledge states:

Theorem 4 *An operation $>$ defined on type theoretical knowledge states is an operation of kernel consolidation iff for all contexts Γ :*

1. ($\Gamma >$) is consistent (consistency)

⁴Sometimes a term representing a non-minimal proof can be β -reduced to a minimal one, since β -reduction corresponds to cut elimination (see for instance [5], pp.56 -57): take $\Gamma \equiv \Gamma_{init}, x : A, y : B, z : A \rightarrow \perp$, and $M \equiv ((\lambda u : A. (\lambda v : B. u))x)y)z : \perp$, then the \perp -implying subset S^M is $\{x : A, y : B, z : A \rightarrow \perp\}$. After performing β -reduction twice, we find the normal form of M which is xz . Now $\{x : A, z : A \rightarrow \perp\}$ is a minimal \perp -implying subset.

2. $(\Gamma >) \subseteq \Gamma$ (inclusion)

3. If $x := M : A$ is an independent member of $\Gamma - (\Gamma >)$, then there exists some Γ' such that $\Gamma' \subseteq \Gamma$, Γ' is consistent and $\Gamma', x := M : A$ is inconsistent (core-retainment).

PROOF Given that x is independent, the proof is completely analogous to that of Hansson. There are two cases in the proof where the independence is needed to ensure that a statement is an element of $\sigma(\Gamma \amalg^h \perp)$ rather than merely an element of $\text{dep}_\Gamma(\sigma(\Gamma \amalg^h \perp))$: in proving core-retainment in the direction *from construction to postulates*, and in proving that σ is an incision function in the direction *from postulates to construction*.

Computing Science Reports

Department of Mathematics and Computing Science Eindhoven University of Technology

If you want to receive reports, send an email to: m.m.j.l.philips@tue.nl (we cannot guarantee the availability of the requested reports)

In this series appeared:

- | | | |
|-------|--|---|
| 97/02 | J. Hooman and O. v. Roosmalen | A Programming-Language Extension for Distributed Real-Time Systems, p. 50. |
| 97/03 | J. Blanco and A. v. Deursen | Basic Conditional Process Algebra, p. 20. |
| 97/04 | J.C.M. Baeten and J.A. Bergstra | Discrete Time Process Algebra: Absolute Time, Relative Time and Parametric Time, p. 26. |
| 97/05 | J.C.M. Baeten and J.J. Vereijken | Discrete-Time Process Algebra with Empty Process, p. 51. |
| 97/06 | M. Franssen | Tools for the Construction of Correct Programs: an Overview, p. 33. |
| 97/07 | J.C.M. Baeten and J.A. Bergstra | Bounded Stacks, Bags and Queues, p. 15. |
| 97/08 | P. Hoogendijk and R.C. Backhouse | When do datatypes commute? p. 35. |
| 97/09 | Proceedings of the Second International Workshop on Communication Modeling, Veldhoven, The Netherlands, 9-10 June, 1997. | Communication Modeling- The Language/Action Perspective, p. 147. |
| 97/10 | P.C.N. v. Gorp, E.J. Luit, D.K. Hammer E.H.L. Aarts | Distributed real-time systems: a survey of applications and a general design model, p. 31. |
| 97/11 | A. Engels, S. Mauw and M.A. Reniers | A Hierarchy of Communication Models for Message Sequence Charts, p. 30. |
| 97/12 | D. Hauschildt, E. Verbeek and W. van der Aalst | WOFLAN: A Petri-net-based Workflow Analyzer, p. 30. |
| 97/13 | W.M.P. van der Aalst | Exploring the Process Dimension of Workflow Management, p. 56. |
| 97/14 | J.F. Groote, F. Monin and J. Springintveld | A computer checked algebraic verification of a distributed summation algorithm, p. 28 |
| 97/15 | M. Franssen | λP -: A Pure Type System for First Order Logic with Automated Theorem Proving, p.35. |
| 97/16 | W.M.P. van der Aalst | On the verification of Inter-organizational workflows, p. 23 |
| 97/17 | M. Vaccari and R.C. Backhouse | Calculating a Round-Robin Scheduler, p. 23. |
| 97/18 | Werkgemeenschap Informatiewetenschap redactie: P.M.E. De Bra | Informatiewetenschap 1997 Wetenschappelijke bijdragen aan de Vijfde Interdisciplinaire Conferentie Informatiewetenschap, p. 60. |
| 98/01 | W. Van der Aalst | Formalization and Verification of Event-driven Process Chains, p. 26. |
| 98/02 | M. Voorhoeve | State / Event Net Equivalence, p. 25 |
| 98/03 | J.C.M. Baeten and J.A. Bergstra | Deadlock Behaviour in Split and ST Bisimulation Semantics, p. 15. |
| 98/04 | R.C. Backhouse | Pair Algebras and Galois Connections, p. 14 |
| 98/05 | D. Darns | Flat Fragments of CTL and CTL*: Separating the Expressive and Distinguishing Powers. P. 22. |
| 98/06 | G. v.d. Bergen, A. Kaldewaij V.J. Dielissen | Maintenance of the Union of Intervals on a Line Revisited, p. 10. |
| 98/07 | Proceedings of the workshop on Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM'98) June 22, 1998 Lisbon, Portugal | edited by W. v.d. Aalst, p. 209 |
| 98/08 | Informal proceedings of the Workshop on User Interfaces for Theorem Provers. Eindhoven University of Technology ,13-15 July 1998 | |

edited by R.C. Backhouse, p. 180

- 98/09 K.M. van Hee and H.A. Reijers An analytical method for assessing business processes, p. 29.
- 98/10 T. Basten and J. Hooman Process Algebra in PVS
- 98/11 J. Zwanenburg The Proof-assistent Yarrow, p. 15
- 98/12 Ninth ACM Conference on Hypertext and Hypermedia
Hypertext '98
Pittsburgh, USA, June 20-24, 1998
Proceedings of the second workshop on Adaptive Hypertext and Hypermedia.
Edited by P. Brusilovsky and P. De Bra, p. 95.
- 98/13 J.F. Groote, F. Monin and J. v.d. Pol Checking verifications of protocols and distributed systems by computer.
Extended version of a tutorial at CONCUR'98, p. 27.
- 98/14 T. Verhoeff (artikel volgt)
- 99/01 V. Bos and J.J.T. Kleijn Structured Operational Semantics of χ , p. 27
- 99/02 H.M.W. Verbeek, T. Basten
and W.M.P. van der Aalst Diagnosing Workflow Processes using Woflan, p. 44
- 99/03 R.C. Backhouse and P. Hoogendijk Final Dialgebras: From Categories to Allegories, p. 26
- 99/04 S. Andova Process Algebra with Interleaving Probabilistic Parallel Composition, p. 81
- 99/05 M. Franssen, R.C. Velkamp and
W. Wesselink Efficient Evaluation of Triangular B-splines, p. 13
- 99/06 T. Basten and W. v.d. Aalst Inheritance of Workflows: An Approach to tackling problems related to change, p. 66
- 99/07 P. Brusilovsky and P. De Bra Second Workshop on Adaptive Systems and User Modeling on the World Wide
Web, p. 119.
- 99/08 D. Bosnacki, S. Mauw, and T. Willemse Proceedings of the first international syposium on Visual Formal Methods - VFM'99
- 99/09 J. v.d. Pol, J. Hooman and E. de Jong Requirements Specification and Analysis of Command and Control Systems
- 99/10 T.A.C. Willemse The Analysis of a Conveyor Belt System, a case study in Hybrid Systems and timed
 μ CRL, p. 44.
- 99/11 J.C.M. Baeten and C.A. Middelburg Process Algebra with Timing: Real Time and Discrete Time, p. 50.
- 99/12 S. Andova Process Algebra with Probabilistic Choice, p. 38.
- 99/13 K.M. van Hee, R.A. van der Toorn,
J. van der Woude and P.A.C. Verkoulen A Framework for Component Based Software Architectures, p. 19
- 99/14 A. Engels and S. Mauw Why men (and octopuses) cannot juggle a four ball cascade, p. 10
- 99/15 J.F. Groote, W.H. Hesselink, S. Mauw,
R. Vermeulen An algorithm for the asynchronous *Write-All* problem based on process collision*,
p. 11.
- 99/16 G.J. Houben, P. Lemmens A Software Architecture for Generating Hypermedia Applications for Ad-Hoc
Database Output, p. 13.
- 99/17 T. Basten, W.M.P. v.d. Aalst Inheritance of Behavior, p.83
- 99/18 J.C.M. Baeten and T. Basten Partial-Order Process Algebra (and its Relation to Petri Nets), p. 79
- 99/19 J.C.M. Baeten and C.A. Middelburg Real Time Process Algebra with Time-dependent Conditions, p.33.
- 99/20 Proceedings Conferentie Informatiewetenschap 1999
Centrum voor Wiskunde en Informatica
12 november 1999, p.98 edited by P. de Bra and L. Hardman
- 00/01 J.C.M. Baeten and J.A. Bergstra Mode Transfer in process Algebra, p. 14
- 00/02 J.C.M. Baeten Process Algebra with Explicit Termination, p. 17.
- 00/03 S. Mauw and M.A. Reniers A process algebra for interworkings, p. 63.
- 00/04 R. Bloo, J. Hooman and E. de Jong Semantical Aspects of an Architecture for Distributed Embedded
Systems*, p. 47.

00/05	J.F. Groote and M.A. Reniers	Algebraic Process Verification, p. 65.
00/06	J.F. Groote and J. v. Wamel	The Parallel Composition of Uniform Processes wit Data, p. 19
00/07	C.A. Middelburg	Variable Binding Operators in Transition System Specifications, p. 27.
00/08	I.D. van den Ende	Grammars Compared: A study on determining a suitable grammar for parsing and generating natural language sentences in order to facilitate the translation of natural language and MSC use cases, p. 33.
00/09	R.R. Hoogerwoord	A Formal Development of Distributed Summation, p. 35
00/10	T. Willemse, J. Tretmans and A. Klomp	A Case Study in Formal Methods: Specification and Validation on the OM/RR Protocol, p. 14.
00/11	T. Basten and D. Bošnački	Enhancing Partial-Order Reduction via Process Clustering, p. 14
00/12	S. Mauw, M.A. Reniers and T.A.C. Willemse	Message Sequence Charts in the Software Engineering Process, p. 26
00/13	J.C.M. Baeten, M.A. Reniers	Termination in Timed Process Algebra, p. 36
00/14	M. Voorhoeve, S. Mauw	Impossible Futures and Determinism, p. 14
00/15	M. Oostdijk	An Interactive Viewer for Mathematical Content based on Type Theory, p. 24.
00/16	F. Kamareddine, R. Bloo, R. Nederpelt	Characterizing λ -terms with equal reduction behavior, p. 12

Belief Revision with Explicit Justifications:
an Exploration in Type Theory

by

Tijn Borghuis and Rob Nederpelt

00/17

ISSN 0926-4515

All rights reserved

editors: prof.dr. J.C.M. Baeten
prof.dr. P.A.J. Hilbers

Reports are available at:
<http://www.win.tue.nl/win/cs>

Computing Science Reports 00/17
Eindhoven, October 2000