BELIEF REVISION IN TYPE THEORY

TIJN BORGHUIS†AND FAIROUZ KAMAREDDINE‡AND ROB NEDERPELT†

†*Mathematics and Computing Science, Eindhoven Univ. of Technology, P.O.Box 513, 5600 MB Eindhoven, the Netherlands. E-mail: {v.a.j.borghuis,wsinrpn}@win.tue.nl*

‡*Computing and Electrical Engineering, Heriot-Watt Univ., Riccarton, Edinburgh EH14 4AS, Scotland. E-mail: fairouz@cee.hw.ac.uk*

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 in the context of type theory. This allows for a deductive perspective on belief revision which can be implemented using existing machinery for deductive reasoning.

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 usually considered as first-class citizens. The two main approaches for dealing with belief revision (foundation and coherence theories[5]) represent justifications of beliefs implicitly (e.g. as relations between beliefs in foundations theory) 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[2] in the DenK-project[4]. In this project a formal model was made of a specific communication situation, and based on this model, a human-computer interface was implemented. 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 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 operate on both), every inconsistency that surfaces in the agents belief state has its own justification containing the justifications of the beliefs that cause the inconsistency.

2 Type theory for knowledge representation

Judgements: The basic relation in type theory is the *judgement* $\Gamma \vdash a : T$ (read as 'term a has type T in context Γ'). Here 'a' and 'T' are both formulas written according to a well-defined syntax. $a : T$ is called a *statement*, whose *subject* is the term a. 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. $x_1 : T_1, \ldots, x_n : T_n$. The judgement $\Gamma \vdash a : T$ 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* $x_1, \ldots, x_n$, so a *depends on* x_1 to x_n. The set of subject variables $\{x_1, \ldots, x_n\}$ is called the *domain* of Γ.

Statements: The intuitive notion 'has type' has a direct counterpart in naive set theory, viz. 'is element of'. For example, the statement '$a : \mathbf{N}$' ('term a *has type* $\mathbf{N}$'), assuming that $\mathbf{N}$ is a symbol representing the set of natural numbers, can 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' (abbreviated as *PAT*) in order to emphasize this usage of types.

Contexts: The context Γ in a judgement $\Gamma \vdash a : T$ contains the 'prerequisites' necessary for establishing the statement $a : T$. In $\Gamma = x_1 : T_1, \ldots, x_n : T_n$, a statement $x_i : T_i$ expresses many 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 deal with the well-formedness of these types. For the T_i in 1. and 2. above, we can have:

1. T_i : `set`, to express that T_i is a well-formed formula representing a set,
2. T_i : `prop`, to express that T_i is well-formed representing a proposition.

The last-mentioned statements can also be part of a context. So a context could look like: $T_1 : \texttt{prop}, T_2 : \texttt{set}, x_1 : T_1, x_2 : T_2$. 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 $\mathcal{S}$.

We identify three characteristics of knowledge which, according to us, should be taken into account in any attempt to formalize knowledge:

- *Subjectivity:* Knowledge of an agent is *partial*: no one knows everything, and agents differ in what they know and don't know. Also, knowledge is formulated in terms of *concepts* which are subjective in the sense that one agent may judge something to be an instance of a certain concept, while another agent would not recognize this as such.
- *Justification:* Knowledge is justified: agents not only *know* things, but they have *reasons* for knowing them. Generally, parts of knowledge are justified in terms of more basic parts; an agent'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.).
- *Incrementality:* The knowledge of an agent can be *extended* as new information becomes available. Whether this information can be incorporated by the agent 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.

With these requirements, the traditional 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. An agent'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 agent can have a concept, whereas another agent has no comparable concept. And in case agents *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 allow 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:* An agent'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' an agent's knowledge state. In type theory, a statement can only extend a context if it obeys certain well-formedness restrictions.

The knowledge *state* of an agent consists of 'everything he knows' at some instant. Given our characterization of knowledge, this means that everything in a knowledge state is formulated in terms of the agent's concepts. Hence:

- *Meaningfulness:* An agent has formed his own, private concepts, and only things 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. (We assume that the entirety of concepts of an agent is fixed.)

- *Inhabitation:* Whatever an agent knows about the world is recorded in a knowledge state in the form of meaningful expressions that he accepts. This includes expressions about which objects 'inhabit' the concepts, and which propositions hold, according to the agent.

If we take the following (very simple) context as representing an agent's knowledge state: $T_1 : \mathtt{prop}, T_2 : \mathtt{set}, x_1 : T_1, x_2 : T_2$, we can see:

- *Meaningfulness is captured by statements of the form* $T : \mathtt{prop}$ *or* $T : \mathtt{set}$. That is to say, in this example the agent has two concepts, viz. T_1, which is a proposition to him, and T_2, which is a set. 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 agent'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 agent.

'Everything an agent knows' at a certain instant can be divided into:

- *Explicit knowledge* expressed by the statements in context Γ. These are explicitly represented pieces of knowledge directly available to the agent.

- *Implicit knowledge* expressed by statements *derivable* on context Γ. These are consequences (obtained by inference) of an agent's explicit knowledge.

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

3 Concluding remarks

We explored the use of explicitly represented justifications in belief revision where beliefs and belief states were represented respectively as type theoretical statements and contexts (for details see [3]). Justifications make it easy to identify the beliefs that cause inconsistency of the belief state and greatly simplify the handling of dependencies between beliefs. Our approach is applicable to agents with limited computational resources because it is deductive and we do not require that our theory of belief revision itself selects which beliefs have to be removed. 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) a large family of type systems, and hence b) given the connections between type theory and logic, in a wide range of logics[2]. Our work has been implemented on the basis of a standard type theoretic theorem prover where the agents belief state is represented as type theoretical contexts as described in this paper [4].

Although we know of no work in the literature where justifications are explicitly represented, we show in [3] that our framework is related to: a) revision for belief bases and to Foundations Theory, but 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; and b) the work of Hansson on semi-revision, whose notion of consolidation can be simulated in our framework and where new information is not automatically completely trusted.

References

1. Ahn, R., Borghuis, T., Communication Modelling and Context-Dependent Interpretation: an Integrated Approach. In: TYPES'98. 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. Borghuis, T., and Nederpelt, R., Belief Revision with Explicit Justifications, an Exploration in Type Theory. CS-report 00-17, Eindhoven University of Technology, Dept. of Math. and Comp. Sc., NL (2000).
4. 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.
5. Gärdenfors, P., *The dynamics of belief systems: Foundations versus coherence theories*, Revue Int. de Philosophie, 44 (1990), pp. 24 – 46.