Automated Support for the Development of Non-Classical Logics

Ullrich Hustadt
Max-Planck-Institut für Informatik,
Im Stadtwald, W-6600 Saarbrücken, Germany
Phone: (+49 681) 302-5431, Fax: (+49 681) 302-5430,
E-mail: `Ullrich.Hustadt@mpi-sb.mpg.de`

Abstract

The most natural means for specifying a non-classical logic is by means of a Hilbert calculus. Usually, the semantics of a non-classical logic is given in terms of possible worlds. Given an axiomatization of a non-classical logics, the *correspondence problem* in these logics is to find for every given Hilbert axiom an equivalent property of the accessibility relation (van Benthem (1984)). For mechanizing deduction in non-classical logics it is very important to find these correspondences Ohlbach (1991). So far the method for finding the correspondences was mostly by intuition and the verification required complex proofs van Benthem (1984).
SCAN is an algorithm which offers a method for computing the correspondences fully automatically. Moreover, since SCAN preserves equivalences, the computed correspondence axioms are *guaranteed to be complete* in the sense that a formula is derivable in the Hilbert calculus if and only if it is valid in the frames which are models of the computed correspondence axiom.
In this paper we present the SCAN algorithm and an application of it to the problem of collapsing modal operators in multi-modal logics: Given a Hilbert calculus for modal operators $\Box_{m_1}$ and $\Box_{m_2}$ we have to ensure that

$$\Box_{m_1} P \Leftrightarrow \Box_{m_2} P$$

doesn't hold for all formulae P, because this is in general an unwanted consequence of the given axiomatization.

1 Introduction

Except for time where we have a precise mathematical model, all the other notions used to describe mental attitudes of agents are fuzzy. What is 'belief', 'knowledge', 'intention' etc. in mathematical terms? Nobody knows, therefore the only way to use these notions in a computer system is by means of approximations – formulations which are mathematically precise, but which do not necessarily reflect all aspects of the reality. Since the application defines the necessary degree of the approximation, a method is needed for specifying the approximation on a very abstract level and compiling it as automatically as possible into a mechanizable deduction system. In Gabbay and Ohlbach (1992), Szałas (1992), and Simmons (1992) a method for transformation a Hilbert calculus into predicate logic was proposed, which can be applied to this problem.

As an example, suppose, *Want*, *Believe* and *because*–operators are to be defined. Nobody has an idea what these notions mean in precise mathematical terms. But everybody would agree to some basic correlations between them, for example if you want R because you believe that R implies Q then in reality you want Q, or formally

$$\mathit{Want}\ R\ \mathit{because}\ \mathit{Believe}(R \rightarrow Q) \rightarrow \mathit{Want}\ Q$$

This is an Hilbert axiom correlating the *Want*–Operator with *Believe*, *because* and $\rightarrow$. With a growing number of modal operators like *Want* or *Believe*, it will be hard to tell wether the correlations are still consistent with each other. Furthermore, if we have chosen multiple modal operators to describe different mental attitudes, these modal operators shouldn't become equivalent. An equivalence like

$$\forall R\colon \mathit{Want}\ R \leftrightarrow \mathit{Believe}\ R$$

is usually an unwanted correlation, although the set of axioms can still be consistent. In such a situation, we say that *Want* and *Believe collapse.*

In the sequel we will introduce a method for second-order quantifier elimination. Gabbay and Ohlbach (1992) and Allgayer, Ohlbach, and Reddig (1992) have shown the usefulness of second-order quantifier elimination for the mechanization of non-classical logics. In this paper we will focus on the application of second-order quantifier elimination to the problem of axiomatizing consistent modal logics with non-collapsing modal operators.

2 Hilbert Calculi for Non-Classical Logics

The most natural means for specifying a logic for fuzzy notions like *Want* or *Believe* is by means of a Hilbert calculus. A Hilbert calculus consists of some axioms, i.e. formulae or formulae schemata respectively which are apriori considered true, together with some production rules for generating from the axioms other true formulae (tautologies). The axiomatization of normal modal logic (Chellas 1980) which may be used as a basis for axiomatizing 'belief' etc. is an example for a Hilbert calculus:

Example 1 (Hilbert Calculus for Normal Modal Logic)

Axioms:	all axioms of propositional logic	
	$\Box(P \to Q) \to (\Box P \to \Box Q)$	(K–Axiom)
Rules:	$\dfrac{P, \quad P \to Q}{Q}$	(Modus Ponens)
	$\dfrac{P}{\Box P}$	(Necessitation Rule)

Implicitly there is always a substitution rule allowing to consider the propositional variables $(P, Q, \ldots)$ as placeholders for arbitrary formulae. Different variants of the logic manifest themselves by additional Hilbert axioms.

Usually, the semantics of a modal logic is given in terms of possible worlds. Hilbert axioms like the K-Axiom above correspond to particular properties of the possible worlds structure. For example, the Hilbert axiom

$$\forall P \ \Box P \to P \tag{1}$$

corresponds to the reflexivity of the accessibility relation. The *correspondence problem* in these logics is to find for a given Hilbert axiom an equivalent semantic property of the accessibility relation (van Benthem 1984). For mechanizing deduction in non-classical logics it is very important to find these correspondences (Ohlbach 1991). So far the method for finding the semantic properties of the accessibility relation was mostly by intuition and the verification required complex proofs (van Benthem 1984).

In the following section, we will introduce an algorithm for second-order quantifier elimination, which can be used to mechanize the finding of semantic properties.

3 The SCAN Algorithm

SCAN[1] is the first algorithm which offers a method for computing the correspondences fully automatically. Moreover, since SCAN preserves equivalences, the computed semantic properties are *guaranteed to be complete* in the sense that a formula is derivable in the Hilbert calculus if and only if it is valid in the frames which are models of the computed corresponding semantic properties.

Definition 2 (The SCAN Algorithm)
Input to SCAN is a formula $\alpha = \exists P_1, \ldots, P_n \ \psi$ with predicate variables $P_1, \ldots, P_n$ and an arbitrary first–order formula ψ.
Output of SCAN — if it terminates — is a formula φ_α which is logically equivalent to α, but not containing the predicate variables $P_1, \ldots, P_n$.

SCAN performs the following three steps:

[1]SCAN means 'Synthesizing Correspondence Axioms for Normal logics.' The name has been chosen before we realized that is applicable in a general context.

1. ψ is transformed into clause form using second order skolemization. That means the resulting formula has the form: $\exists P_1, \ldots, P_n \exists f_1, \ldots, f_n \psi'$ where the f_i are the Skolem functions and ψ' is a set of clauses. From the algorithm's point of view, the quantifier prefix can be ignored. Therefore ψ' is treated as an ordinary clause set with the usual Skolem constants and functions.

2. All C–resolvents and C–factors with the predicate variables $P_1, \ldots, P_n$ have to be generated. C–resolution ('C' for constraint) is defined as follows:

$$\begin{array}{ll} P(s_1, \ldots, s_n) \vee C & P(\ldots) \text{ and } \neg P(\ldots) \\ \neg P(t_1, \ldots, t_n) \vee D & \text{are the } \textit{resolution literals} \\ \hline C \vee D \vee s_1 \neq t_1 \vee \ldots \vee s_n \neq t_n & \end{array}$$

   and the C-factorization rule is defined analogously:

$$\frac{P(s_1, \ldots, s_n) \vee P(t_1, \ldots, t_n) \vee C}{P(s_1, \ldots, s_n) \vee C \vee s_1 \neq t_1 \vee \ldots \vee s_n \neq t_n}.$$

   Notice that only C-resolutions between different clauses are allowed (no self resolution). A C-resolution or C-factorization can be optimized by destructively resolving literals $x \neq t$ where the variable x does not occur in t with the reflexivity equation. C–resolution and C–factorization takes into account that second order quantifiers may well impose conditions on the interpretations which must be formulated in terms of equations and inequations.

   As soon as *all* resolvents and factors between a particular literal and the rest of the clause set have been generated (the literal is 'resolved away'), the clause containing this literal must be deleted (purity deletion). If all clauses are deleted this way, this means that α is a tautology.

   All equivalence preserving simplifications may be applied freely. These are for example:

   - Tautologous resolvents can be deleted.
   - Subsumed clauses can be deleted.
   - Subsumption factoring can be performed. Subsumption factoring means that a factor subsumes its parent clause. This may be realized by just deleting some literals. For example $Q(x), Q(a)$, where x is a variable, can be simplified to $Q(a)$.
   - Subsumption resolution can also be performed. Subsumption resolution means that a resolvent subsumes its parent clause, and this again may be realized by deleting some literals Ohlbach and Siekmann (1991). For example the resolvent between $P \vee Q$ and $\neg P \vee Q \vee R$ is just $Q \vee R$ such that $\neg P$ can be deleted from the clause. (An instance of this operation is realized as so called 'unit_deletion' in the OTTER theorem prover.)

If an empty clause is generated, this means that α is contradictory.

3. If the previous step terminates and there are still clauses left then reverse the skolemization. A method for reversing the skolemization in a set F of clauses is (1) to abstract all arguments of all occurrences of Skolem functions by variables, i.e. $f(s_1, \ldots, s_n)$ is replaced with $f(x_1, \ldots, x_n)$ and additional literals $x_i \neq s_i$ are added to the clause where the x_i are fresh variables and (2) to consistently rename all variables such that the arguments of all occurrences of the Skolem function are the same. If this is possible and $F[f(x_1, \ldots, x_n)]$ is the result then $\forall x_1, \ldots, x_n \exists y\ F[y]$ is the solution. This process is repeated for all Skolem functions.

   If it is not possible to rename the variables consistently, the only chance is to take parallel Henkin quantifiers (Henkin 1961) or leave the second–order quantification.

$\triangle$

The step from the Hilbert axioms to the second-order formulae is explained in detail in Gabbay and Ohlbach (1992). For our purposes it is enough to assume that we have semantic definitions for the modal operators and other connectives in the Hilbert axioms, which can be used to rewrite the axioms to second-order formulae.

We illustrate how the SCAN algorithm can be applied by computing the reflexivity of the accessibility relation from the axiom $\Box P \rightarrow P$.

Example 3 We start with the formula

$$\alpha = \forall\, P \colon \Box P \rightarrow P$$

and semantic definitions

$$\forall\, P \colon w \models \Box P \Leftrightarrow \forall\, w' \colon \mathcal{R}(w, w') \Rightarrow w' \models P$$

and

$$\forall\, P \colon w \models P \rightarrow R \Leftrightarrow w \models \neg P \vee R,$$

where $\mathcal{R}$ is the accessibility relation on worlds.

Since SCAN accepts only existentially quantified predicate variables, α is negated and then the semantic definitions are applied as rewrite rules until $\Box$ and $\rightarrow$ disappear. The result is put into clause form.

We obtain the two clauses:

$$\begin{array}{r} \neg\mathcal{R}(w, w') \vee P(w'), \\ \neg P(w) \end{array}$$

where w is a Skolem constant. Resolution with P-literals — only these resolutions are allowed — yields

$$\begin{array}{r} \neg\mathcal{R}(w, w') \vee P(w'), \\ \neg P(w) \\ \hline \neg\mathcal{R}(w, w) \hfill \end{array}$$

The two parent clauses are pure (no further resolutions are possible) and can be deleted. The single remaining clause $\neg\mathcal{R}(w,w)$ is unskolemized to $\exists\, w\colon \neg\mathcal{R}(w,w)$ and negated yielding $\forall\, w\colon \mathcal{R}(w,w)$ which is the desired result[2].

4 Detecting Collapsing Modal Operators

The need for a tool like SCAN becomes apparent if we consider a set of Hilbert axioms specifying not only a single modal operator, but multiple, interacting modal connectives. For the standard Hilbert axioms describing a single modal operator the corresponding properties of the accessibility relation are well-known. For multiple, interacting modal operators the corresponding results are less accessible. With SCAN we can compute the corresponding properties of the accessibility relation automatically.

But we can do even more. The first step after specifying a set of Hilbert axioms for multiple, interacting modal operators is to exhibit their consistency. In a second step, we have to show that the modal operators do not collapse, i.e.

$$\Box_{m_1} P \leftrightarrow \Box_{m_2} P$$

shouldn't hold for distinct modal operators $\Box_{m_1}$ and $\Box_{m_2}$.

The first step can be done with the aid of SCAN and any first-order refutational theorem prover. We use SCAN to find the correspondence axioms of the given Hilbert axioms. If SCAN terminates and produces a set of first-order formulae for every axiom, we feed the first-order refutational theorem prover with all these formulae to find an inconsistency. Of course, the problem of proving the inconsistency of a set of first-order formulae is only semi-decidable. In case our set of formulae is consistent, the theorem prover may not terminate.

In the second step, we want to show that

$$\neg\forall\, P\colon \Box_{m_1} P \leftrightarrow \Box m_2\, P.$$

holds for modal operators $\Box_{m_1}$ and $\Box_{m_2}$. We use SCAN to eliminate the second order quantifier $\forall\, P$. We will get a corresponding first-order semantic property (if the process terminates). This first-order semantic property can be proven using a first-order refutational theorem prover.

Example 4 van der Hoek (1992) considers the following Hilbert axioms for a multi-modal logic of knowledge and belief:

$$\neg\Box_{believe}\, \mathit{false} \tag{2}$$

$$\neg\Box_{know} P \rightarrow \Box_{know} \neg\Box_{know} P \tag{3}$$

$$\Box_{know} P \rightarrow \Box_{believe} P \tag{4}$$

$$\Box_{believe} P \rightarrow \Box_{believe} \Box_{know} P \tag{5}$$

[2] In this particular example we get the "relational" translation of the $\Box$-Operator (Moore 1980, Ohlbach 1991). A slight modification of the semantics of $\Box$ yields the more compact and computationally more efficient "functional" translation (Ohlbach 1991).

Axiom (2) says that we don't believe anything that is false. Axiom (3) is the axiom of negative knowledge introspection, i.e. if we don't know P, then we know that we don't know P. Axiom (4) defines that knowledge is 'stronger' than believe. Axiom (5) is intended to model the attitude of an agent who thinks that he is very critical in adopting believes: he only believes P if he believes that he knows P.

SCAN produces the following corresponding semantic properties in clause form

$$\mathcal{R}_{believe}(w_1, f(w_1)) \tag{6}$$
$$\neg\mathcal{R}_{know}(w_1, w_2) \vee \neg\mathcal{R}_{know}(w_1, w_3) \vee \mathcal{R}_{know}(w_2, w_3) \tag{7}$$
$$\neg\mathcal{R}_{believe}(w_1, w_2) \vee \neg\mathcal{R}_{know}(w_2, w_3) \vee \mathcal{R}_{believe}(w_1, w_3) \tag{8}$$
$$\neg\mathcal{R}_{believe}(w_1, w_2) \vee \mathcal{R}_{know}(w_1, w_2), \tag{9}$$

where w_1, w_2, and w_3 denote (universally quantified) variables and f is a Skolem function. (6)–(9) say that $\mathcal{R}_{believe}$ is serial, contained in $\mathcal{R}_{know}$, transitive over $(\mathcal{R}_{believe}, \mathcal{R}_{kow})$, and $\mathcal{R}_{know}$ is euclidean.

Now is there actually a difference between knowledge and believe provided the given Hilbert axioms hold? We already know that for all P

$$\Box_{know}P \to \Box_{believe}P$$

holds. Now we want to check if the converse is also true. That is, we check if

$$\Box_{believe}P \to \Box_{know}P \tag{10}$$

is an additional consequence of the Hilbert axioms. The corresponding semantic property of (10) is

$$\forall\, w_1, w_2 {:}\, \mathcal{R}_{know}(w_1, w_2) \to \mathcal{R}_{believe}(w_1, w_2).$$

Negating this property yields the clauses

$$\mathcal{R}_{know}(v, w) \tag{11}$$
$$\neg\mathcal{R}_{believe}(v, w), \tag{12}$$

where v and w are Skolem constants. Resolution of clause (8) with (12) and (6) yields

$$\neg\mathcal{R}_{know}(f(v), v) \tag{13}$$

Now clause (7) can be resolved with clause (13) and (11) giving

$$\neg\mathcal{R}_{know}(v, f(v)). \tag{14}$$

The resolvent of clause (14) and clause (9) is

$$\neg\mathcal{R}_{believe}(v, f(v)) \tag{15}$$

Resolution of clause (15) and (6) yields the empty clause, i.e. we have proven the inconsistency of this set $\{(6), \ldots, (12)\}$ of clauses.

So,

$$\Box_{know} P \leftrightarrow \Box_{believe} P$$

actually holds for all P. Every clause of the semantic properties corresponding to the considered Hilbert axioms has been used in the proof above. Therefore, all four Hilbert axioms together force the collapse of knowledge and belief. We have to abandon at least one of them to get back to a set of axioms where knowledge and belief are distinct operators.[3]

5 Limitations of SCAN

The SCAN algorithm can produce first-order semantic properties only for those second-order Hilbert axioms where such corresponding properties actually exist. From modal logic we know cases where a Hilbert axiom has a semantic property which is only second-order axiomatizable. The most well-known example is the McKinsey axiom

$$\forall P\colon \Box\Diamond P \rightarrow \Diamond\Box P.$$

SCAN produces the following property of the accessibility relation for the McKinsey axiom:

$$\forall a \begin{pmatrix} \forall f\ \exists x \\ \forall g\ \exists y \end{pmatrix} \begin{array}{l} ((\mathcal{R}(a,x) \rightarrow \mathcal{R}(x,f(x)))\wedge \\ (\mathcal{R}(a,y) \rightarrow \mathcal{R}(y,g(y)))) \rightarrow \\ (\mathcal{R}(a,x) \wedge \mathcal{R}(a,y) \wedge f(x) = g(y)). \end{array}$$

Because of the quantification over functions, this is still second-order. In such a case, we will not be able to make further investigations.

Furthermore, SCAN cannot invent a good axiomatization of a non-classical logic. For example, the McKinsey axiom combined with the transitivity axiom

$$\Box P \rightarrow \Box\Box P,$$

corresponds to a first-order axiomatizable property of the accessibility relation, i.e. atomicity of the accessibility relation

$$\forall x\colon \exists y\colon (\mathcal{R}(x,y) \wedge \forall z\colon \mathcal{R}(y,z) \rightarrow z = y))$$

(van Benthem 1984, page 203). Therefore, the combination of these two axioms is suitable for a mechanization using a first-order theorem prover. Of course, there is no way for SCAN to give a hint that the McKinsey axiom should be combined with the transitivity axiom to get a usable set of semantic properties.

[3]van der Hoek (1992) also proved the collapse of knowledge and belief. The advantage we get using SCAN is the ability to perform such checks automatically.

6 Conclusion

Cohen (1991) claims that AI is suffering a methodological malaise, because AI is dominated by two largely distinct camps: model-oriented researchers and system-oriented researchers. One of the problems is that the results of model-oriented researchers migrate only slowly to the system-oriented researchers and on the other hand, the real problems which system-oriented researchers are faced with, arouse only slowly the interest of model-oriented researchers.

SCAN can be seen as a tool developed by model-oriented researchers to help system-oriented researchers to solve their problems in their own way. SCAN is a tool for playing with different axiomatizations of non-classical logics. It can be used, as we have done in this note, to examine the consistency of an axiomatization or detect the collapse of modal operators in a modal logic. Another important application of SCAN is the development of deductive systems for non-classical logics. See Ohlbach (1991).

This illustrates, how SCAN reduces the routine work of researchers, yielding more room for the art of inventing axiomatizations of non-classical logics.

References

Allgayer, J., Ohlbach, H., and Reddig, C., 1992. Modelling Agents with Logic. In Andre, E., Cohen, R., Graf, W., Kass, B., Paris, C., and Wahlster, W., editors, *UM92 — Proceedings of the Third International Workshop on User Modeling, DFKI Document D-92-17*.

Chellas, B. F., 1980. *Modal logic: an introduction*. Cambridge University Press, New York.

Cohen, P., 1991. A Suvey of the Eight National Conference on AI: Pulling Together or Pulling Apart. *AI Magazine* **12**(1):16–41.

Gabbay, D. M. and Ohlbach, H. J., 1992. Quantifier elimination in second–order predicate logic. *South African Computer Journal* **7**:35–43. appeared also in Proc. of KR92, Morgan Kaufmann, 1992, pp 425–436.

Henkin, L., 1961. Some remarks on infinitely long formulas. In *Infinitistic Methods*, pp. 167–183. Pergamon Press, Oxford.

Moore, R., 1980. Reasoning about Knowledge and Action. Technical Note 191, SRI International, Menlo Park, CA.

Ohlbach, H. J., 1991. Semantics Based Translation Methods for Modal Logics. *Journal of Logic and Computation* **1**(5):691–746.

Ohlbach, H. J. and Siekmann, J. H., 1991. The Markgraf Karl Refutation Procedure. In Lassez, J. L. and Plotkin, G., editors, *Computational Logic, Essays in Honor of Alan Robinson*, pp. 41–112. MIT Press.

Simmons, H., 1992. An Algorithm for Eliminating Predicate Variables from Π^1_1 Sentences (with special reference to modal correspondence theory). Department of Computer Science, The University of Manchester, Draft.

Szałas, A., 1992. On Correspondence Between Modal and Classical Logic: Automated Approach. Technical Report MPI–I–92–209, Max-Planck-Institute for Computer Science.

van Benthem, J., 1984. Correspondence Theory. In Gabbay, D. M. and Guenthner, F., editors, *Handbook of Philosophical Logic, Vol. II, Extensions of Classical Logic, Synthese Library Vo. 165*, pp. 167–248. D. Reidel Publishing Company, Dordrecht.

van der Hoek, W., 1992. *Modalities for Reasoning about Knowledge and Quantities.* PhD thesis, Vrije Universiteit van Amsterdam, Elinkwijk, Utrecht, Netherlands.