Reasoning about agents in the KARO framework

U. Hustadt[1], C. Dixon[1], R. A. Schmidt[2], M. Fisher[1], J.-J. Meyer[3], and W. van der Hoek[3]

[1]*Department of Computer Science, University of Liverpool, UK.* {U.Hustadt,C.Dixon,M.Fisher}@csc.liv.ac.uk.
[2]*Department of Computer Science, University of Manchester, UK.* schmidt@cs.man.ac.uk.
[3]*Department of Computer Science, University of Utrecht, The Netherlands.* {jj,wiebe}@cs.uu.nl.

Abstract

This paper proposes two methods for realising automated reasoning about agent-based systems. The framework for modelling intelligent agent behaviour that we focus on is a core of KARO logic, an expressive combination of various modal logics including propositional dynamic logic, a modal logic of knowledge, a modal logic of wishes, and additional non-standard operators. The first method we present is based on a translation of core KARO logic to first-order logic combined with first-order resolution. The second method uses an embedding of core KARO logic into a combination of branching-time temporal logic CTL and multi-modal S5 plus a clausal resolution calculus for these combined logics. We discuss the advantages and shortcomings of each approach and suggest ways to extend each variant to cover more of the KARO framework.

1. Introduction

The use of *agents* is now seen as an essential tool in representing, understanding and implementing complex software systems. In particular, the characterisation of complex components as *intelligent* or *rational* agents allows the system designer to analyse applications at a much higher level of abstraction [1, 2]. In order to describe and reason about such agents, a number of theories of rational agency have been developed, for example the BDI [3] and KARO [4] frameworks. Usually, these frameworks are represented as complex multi-modal logics. These logics, in addition to their use in agent theories, where the basic representation of agency and rationality is explored, form the basis for agent-based formal methods. In both these uses, the notion of proof is important. In agent theories, theorem proving allow us to examine properties of the overall theory and, in some cases, to characterise computation within that theory. In agent-based formal methods, theorem proving is clearly important in developing verification techniques.

The leading agent theories and formal methods in this area all share similar logical properties. Usually, the agent theories used have:

- an *informational* component, being able to represent an agent's beliefs (by the modal logic KD45) or knowledge (by the modal logic S5),
- a *dynamic* component, allowing the representation of dynamic activity (by temporal or dynamic logic), and,
- a *motivational* component, often representing the agents desires, intentions or goals (by the modal logic KD).

Thus, the predominant approaches use particular combinations of modal logics. The particular agent theory that we consider here, the KARO framework (for Knowledge, Abilities, Results and Opportunities) [4], combines actions (via propositional dynamic logic PDL), knowledge, and wishes.

While proof methods have been developed for other agent theories like the BDI framework [5], no such methods exist for the KARO framework. Thus, our aim in this paper is to examine possible approaches to the development of automated proof methods for the KARO framework. We study two approaches to the problem of proof in this complex system:

- proof methods for the fusion of PDL and $\mathsf{S5}_{(\mathsf{m})}$ based upon translation to classical logic and first-order resolution; and
- representation of KARO in terms of the fusion of CTL and $\mathsf{S5}_{(\mathsf{m})}$ and proof methods by direct clausal resolution on this combined logic.

These approaches both show how we can verify properties of agent-based systems represented in the KARO theory of rational agents, but there are fundamental differences in the techniques used. The first approach involves translating all modal and dynamic logic aspects into classical logic and then carrying out proof by defining specific orderings on classical resolution. The second approach retains the non-classical structure and develops appropriate resolution rules for the combined logic. In addition, branching-time temporal logic, rather than propositional dynamic logic, is used to represent the agent's dynamic behaviour.

2. Basic KARO Elements

The KARO logic [6, 7] is a formal system that may be used to *specify*, *analyse* and *reason about* the behaviour of rational agents. Concerning the informational attitudes of agents, in the basic framework [6], it can be expressed that agent i *knows* a fact φ (written as $\mathbf{K}_i\varphi$). The modality $\mathbf{K}_i$ is a standard S5 modality. Consequently, the informational component of KARO is a multi-modal $\mathsf{S5}_{(\mathsf{m})}$ logic. In the full system we also consider beliefs; these epistemic and doxastic attitudes were extensively studied in [8]. On an equal footing with these informational attitudes, the language encompasses a *dynamic* component. Starting with some atomic actions $\mathsf{Ac}_{\mathsf{at}}$, KARO allows for composite actions such as sequential composition $\alpha\,;\beta$, testing $\varphi!$, conditionals (`if` φ `then` α `else` β), repetition (`while` φ `do` α) and we also investigated (cf. [9]) several notions of choice ($\alpha + \beta$). The framework is especially fit to reason about the preconditions for such actions: one can express whether agent i is *able* to perform action α ($\mathbf{A}_i\alpha$) or has the *opportunity* to do α ($\mathbf{O}_i\alpha$), and also that φ is a *result* of doing α ($[\mathsf{do}_i(\alpha)]\varphi$). In this paper we concentrate on one particular variant of the KARO framework and define a core subsystem for which we are able to provide sound, complete, and terminating inference systems.

Formally, the logic we consider is an extended modal logic given by the fusion of a PDL-like logic and multi-modal S5 and KD. The language of the KARO framework is defined over three primitive types: (i) a countably infinite set of propositional variables, (ii) a set Ag of agent names (a finite subset of the positive integers), and (iii) a countably infinite set $\mathsf{Ac}_{\mathsf{at}}$ of atomic actions.

Formulae are defined inductively as follows.

- $\top$ is an atomic propositional formula;
- $\varphi \vee \psi$ and $\neg\varphi$ are propositional formula provided φ and ψ are propositional formulae;
- $\mathbf{K}_i\varphi$ (knowledge), $[\mathsf{do}_i(\alpha)]\varphi$ (achievement of results by actions), $\mathbf{A}_i\alpha$ (ability), $\mathbf{O}_i\alpha$ (opportunity), $\mathbf{W}^s_i\varphi$ (selected wish), and $\Diamond_i\varphi$ (implementability) are propositional formulae, provided i is an agent name, α is an action formula and φ is a propositional formula;
- `id` (skip) is an atomic action formula;
- $\alpha \vee \beta$ (non-deterministic choice), $\alpha\,;\beta$ (sequencing), $\varphi!$ (confirmation or test), $\alpha^{(n)}$ (bounded repetition), and $\alpha^\star$ (unbounded repetition) are action formulae, provided α and β are action formulae, φ is a propositional formula, and n is a natural number.

Implicit connectives include the usual connectives such as $\bot, \wedge, \rightarrow, \ldots$ for propositional formulae, the duals of $\mathbf{K}_i$, $\mathbf{O}_i$ and $[\mathsf{do}_i(\alpha)]$ (denoted by $\langle\mathsf{do}_i(\alpha)\rangle$), as well as

- $\mathbf{PracPoss}_i(\alpha,\varphi) = \langle\mathsf{do}_i(\alpha)\rangle\varphi \wedge \mathbf{A}_i\alpha$,
- $\mathbf{Can}_i(\alpha,\varphi) = \mathbf{K}_i\mathbf{PracPoss}_i(\alpha,\varphi)$,
- $\mathbf{Cannot}_i(\alpha,\varphi) = \mathbf{K}_i\neg\mathbf{PracPoss}_i(\alpha,\varphi)$,
- $\mathbf{Goal}_i\varphi = \neg\varphi \wedge \mathbf{W}^s_i\varphi \wedge \Diamond_i\varphi$, and
- $\mathbf{Intend}_i(\alpha,\varphi) = \mathbf{Can}_i(\alpha,\varphi) \wedge \mathbf{K}_i\mathbf{Goal}_i\varphi$.

The semantics of KARO logic is based on *interpretations* which are tuples $\mathcal{M} = (W, V, D, I, M)$, where: (i) W is a non-empty set of worlds, (ii) V maps propositional variables to subsets of W, (iii) for every $i \in \mathsf{Ag}$ and every $a \in \mathsf{Ac}_{\mathsf{at}}$, D contains a binary relation $r_{(i,a)}$ on W and a subset $c_{(i,a)}$ of W, (iv) I contains an equivalence relation K_i on W for each agent $i \in \mathsf{Ag}$, and (v) M contains a serial relation relation W_i on W for each agent $i \in \mathsf{Ag}$. Following the characterisation of agent theories in the introduction, D, I, and M comprise the dynamic, informational, and motivational components in the semantics of KARO logic.

The relations $r_{(i,a)}$ and sets $c_{(i,a)}$ are extended to $\mathsf{Ag} \times \mathsf{Ac}$-sorted relations $r^*_{(i,a)}$ and sets $c^*_{(i,a)}$ in a way standard for dynamic logic [6]. We state only the cases for the composition of actions and confirmation.

$$
\begin{aligned}
r^*_{(i,\alpha\,;\,\beta)} &= \{(u,w) \mid \exists v \in W.\,(u,v) \in r^*_{(i,\alpha)} \wedge \\
&\qquad\qquad (v,w) \in r^*_{(i,\beta)}\} \\
r^*_{(i,\varphi!)} &= \{(u,w) \mid u = w \wedge \mathcal{M}, w \models \varphi\}
\end{aligned}
$$

and

$$
\begin{aligned}
c^*_{(i,\alpha\,;\,\beta)} &= \{w \mid c^*_{(i,\alpha)}(w) \wedge \\
&\qquad \forall v \in W.(w,v) \in r^*_{(i,\alpha)} \wedge v \in c^*_{(i,\beta)}\} \\
c^*_{(i,\varphi!)} &= \{w \mid \mathcal{M}, w \models \varphi\}.
\end{aligned}
$$

The semantics of well-formed formulae of the KARO logic is defined as follows.

$$
\begin{array}{ll}
\mathcal{M}, w \models \top & \\
\mathcal{M}, w \models p & \text{iff } w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \text{iff } \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \vee \psi & \text{iff } \mathcal{M}, w \models \varphi \text{ or } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models [\mathsf{do}_i(\alpha)]\varphi & \text{iff } \forall v \in W.(w,v) \in r^*_{(i,\alpha)} \rightarrow \\
& \qquad \mathcal{M}, v \models \varphi \\
\mathcal{M}, w \models \mathbf{A}_i\alpha & \text{iff } w \in c^*_{(i,\alpha)} \\
\mathcal{M}, w \models \mathbf{O}_i\alpha & \text{iff } \mathcal{M}, w \models \langle\mathsf{do}_i(\alpha)\rangle\top \\
\mathcal{M}, w \models \mathbf{W}^s_i\varphi & \text{iff } \forall v \in W.(w,v) \in W_i \rightarrow \mathcal{M}, v \models \varphi \\
\mathcal{M}, w \models \mathbf{K}_i\varphi & \text{iff } \forall v \in W.(w,v) \in K_i \rightarrow \mathcal{M}, v \models \varphi \\
\mathcal{M}, w \models \Diamond_i\varphi & \text{iff } \exists k \in \mathbb{N}\, \exists a_1, \ldots, a_k \in \mathsf{Ac}_{\mathsf{at}}. \\
& \qquad \mathcal{M}, w \models \mathbf{PracPoss}_i(a_1; \ldots ; a_k, \varphi)
\end{array}
$$

If $\mathcal{M}, w \models \varphi$ we say φ *holds at* w (in $\mathcal{M}$) or φ is *true in* w. A formula φ is *satisfiable* iff there is an interpretation $\mathcal{M}$ and a world w such that $\mathcal{M}, w \models \varphi$.

Even though the logic defined above does not include all the features of the KARO framework, we refer to it as the *KARO logic*. In this paper we make the following simplifying assumptions: (i) we assume $\mathbf{A}_i\alpha = \mathbf{O}_i\alpha$, (ii) we exclude the unbounded repetition operator $\alpha^\star$, wishes $\mathbf{W}^s_i\varphi$, and implementability $\Diamond_i\varphi$, and (iii) there is no interaction between the dynamic and informational component. This fragment of the KARO logic is called the *core KARO logic*. In Section 5 we will discuss in how far these simplifying assumptions can be relaxed.

3. Proof by Translation

The translation approach to modal reasoning is based on the idea that inference in (combinations of) modal logics can be carried out by translating modal formulae into

$$
\begin{aligned}
\langle \mathtt{do}_i(\alpha)\rangle\psi &\Rightarrow \neg[\mathtt{do}_i(\alpha)]\neg\psi \\
[\mathtt{do}_i(\alpha \vee \beta)]\psi &\Rightarrow [\mathtt{do}_i(\alpha)]\psi \wedge [\mathtt{do}_i(\beta)]\psi \\
[\mathtt{do}_i(\alpha\,;\beta)]\psi &\Rightarrow [\mathtt{do}_i(\alpha)][\mathtt{do}_i(\beta)]\psi \\
[\mathtt{do}_i(\mathtt{id})]\psi &\Rightarrow \psi \\
[\mathtt{do}_i(\phi!)]\psi &\Rightarrow \neg\phi \vee \psi \\
[\mathtt{do}_i(\alpha^{(1)})]\psi &\Rightarrow [\mathtt{do}_i(\alpha)]\psi \\
[\mathtt{do}_i(\alpha^{(n+1)})]\psi &\Rightarrow [\mathtt{do}_i(\alpha)][\mathtt{do}_i(\alpha^{(n)})]\psi
\end{aligned}
$$

Fig. 1. Transformation rules for the core KARO logic

first-order logic and using conventional first-order theorem proving techniques. There are various different translation morphisms for modal logics whose properties vary with regards the extent to which they are able to map modal logics into first-order logic, the decidability of the fragments of first-order logic into which modal formulae are translated, and the computational behaviour of first-order theorem provers on these fragments, see e.g. [10–13].

In the following we present a decision procedure for the satisfiability problem in the core KARO logic consisting of three components: (i) a normalisation function which reduces complex action formulae to atomic action subformulae, (ii) a particular translation of normalised formulae into a fragment of first-order logic, and (iii) a resolution-based decision procedure for this fragment.

Using the rewrite rules given in Figure 1 and similar rules for $\mathbf{O}_i\alpha$ and $\mathbf{A}_i\alpha$, the normalisation function maps any formula φ of the core KARO logic to a normal form $\varphi{\downarrow}$. It is straightforward to see that the rewrite relation defined by these rules is confluent and terminating. The normal form $\varphi{\downarrow}$ of φ is logically equivalent to φ, it is unique, and in the absence of the unbounded repetition operator, $\varphi{\downarrow}$ contains no non-atomic action formulae. We obtain the following result.

Lemma 1 *Let φ be a formula in the core KARO logic without occurrences of the unbounded repetition operator. Then $\varphi{\downarrow}$ is satisfiable iff φ is satisfiable, and $\varphi{\downarrow}$ does not contain any non-atomic action formulae.*

The particular translation we use has only recently been proposed by de Nivelle [14] and can be seen as a special case of the T-encoding introduced by Ohlbach [15]. It allows for conceptually simple decision procedures for extensions of $\mathsf{K4}$ by ordered resolution. As compared to tableaux-based procedures a feature of this approach is the absence of loop checking mechanisms for transitive modal logics.

Without loss of generality we assume that the modal formulae under consideration are normalised and in negation normal form. We define the translation function π as follows.

$$
\begin{aligned}
\pi(\top, x) &= \top \\
\pi(p, x) &= q_p(x) \\
\pi(\neg\varphi, x) &= \neg\pi(\varphi, x) \\
\pi(\varphi \vee \psi, x) &= \pi(\varphi, x) \vee \pi(\psi, x) \\
\pi([\mathtt{do}_i(a)]\varphi, x) &= \forall y\, (\mathtt{do}_i^a(x, y) \rightarrow \pi(\varphi, y)) \\
\pi(\mathbf{O}_i a, x) &= \pi(\langle \mathtt{do}_i(a)\rangle\top, x) \\
\pi(\mathbf{A}_i a, x) &= \pi(\langle \mathtt{do}_i(a)\rangle\top, x) \\
\pi(\mathbf{K}_i\varphi, x) &= q_{\mathbf{K}_i\varphi}(x)
\end{aligned}
$$

a is an atomic action, p is a propositional variable, q_p is a unary predicate symbol uniquely associated with p, $q_{\mathbf{K}_i\varphi}$ is a predicate symbol uniquely associated with $\mathbf{K}_i\varphi$, and $\mathtt{do}_i^a$ is a binary predicate symbol which represents the relation $r_{(i,a)}$ in the semantics.

Finally, let $\Pi(\psi)$ be the formula

$$\exists x\, \pi(\psi, x) \wedge \textstyle\bigwedge_{\mathbf{K}_i\varphi \in \Gamma_{\mathbf{K}}(\psi)} \mathrm{Ax}(\mathbf{K}_i\varphi),$$

where $\Gamma_{\mathbf{K}}(\psi)$ is the set of subformulae of the form $\mathbf{K}_i\varphi$ in ψ, and $\mathrm{Ax}(\mathbf{K}_i\varphi)$ is the formula

$$
\begin{aligned}
&\forall x\, (q_{\mathbf{K}_i\varphi}(x) \leftrightarrow \forall y\, (K_i(x, y) \rightarrow \pi(\varphi, y))) \\
&\wedge \forall x, y\, ((q_{\mathbf{K}_i\varphi}(x) \wedge K_i(x, y)) \rightarrow q_{\mathbf{K}_i\varphi}(y)) \\
&\wedge \forall x, y\, ((q_{\mathbf{K}_i\varphi}(y) \wedge K_i(x, y)) \rightarrow q_{\mathbf{K}_i\varphi}(x)) \\
&\wedge \forall x\, K_i(x, x).
\end{aligned}
$$

Note that no additional definition of K_i is required. Based on the close correspondence between the translation morphism Π and the semantics of the core KARO logic it is possible to prove the following.

Theorem 1 *A formula φ of the core KARO logic is satisfiable iff $\Pi(\varphi)$ is first-order satisfiable.*

One of the advantages of using the translation morphism proposed by de Nivelle is the fact that for any formula φ of the core KARO logic $\Pi(\varphi)$ can easily be seen to belong to a number of well-known solvable first-order classes, including the two-variable fragment of first-order logic or the guarded fragment. Furthermore, using certain structural transformation techniques, described for example in [10], $\Pi(\varphi)$ can be embedded into a number of solvable clausal classes, for example, the classes $\mathcal{S}^+$ [16] and DL^* [10]. In the following, by $\mathrm{CL}_{\mathrm{DL}^*}(\Pi(\varphi))$ we denote an embedding of $\Pi(\varphi)$ into the class DL^*.

For all the solvable classes mentioned, resolution-based decision procedures exist and can be formulated in the resolution framework of Bachmair and Ganzinger [17]. In this framework, the resolution calculus is parameterised by two parameters: an admissible ordering $\succ$ and a selection function S. Essentially, an admissible ordering is a total (well-founded) strict ordering on the ground level such that for literals $\ldots \succ \neg A_n \succ A_n \succ \ldots \succ \neg A_1 \succ A_1$ holds. This is extended to the non-ground level in a canonical manner. A *selection function* assigns to each clause a possibly empty set of occurrences of negative literals and no restrictions are imposed on the selection function.

The calculus itself consists of the following inference rules (A_1, A_2 denote atoms, C, D denote clauses):

Resolution: $$\frac{C \vee A_1 \quad \neg A_2 \vee D}{(C \vee D)\sigma}$$

where (i) σ is a most general unifier of A_1 and A_2, (ii) no literal in C is selected, and $A_1\sigma$ is strictly $\succ$-maximal with respect to $C\sigma$, and (iii) $\neg A_2$ is either selected, or $\neg A_2\sigma$ is maximal with respect to $D\sigma$ and no literal in D is selected.

We implicitly assume that the premises have no common variables.

Factoring: $$\frac{C \vee A_1 \vee A_2}{(C \vee A_1)\sigma}$$

where (i) σ is a most general unifier of A_1 and A_2, and (ii) no literal in C is selected and $A_1\sigma$ is $\succ$-maximal with respect to $C\sigma$.

Besides resolution and factoring, the calculus contains a *splitting rule* akin to disjunction elimination in semantic tableaux. Let N be a set of clauses containing a clause $C = C_1 \vee C_2$ that has two variable-disjoint subclauses C_1 and C_2. Then, instead of refuting N we can try to refute $N \cup \{C_1\}$ and $N \cup \{C_2\}$. If both sets are refutable, then also N is refutable, but if one of the sets is satisfiable, then also N is satisfiable.

The calculus is refutationally complete and compatible with a general notion of *redundancy* for clauses and inferences, with which additional don't-care non-deterministic simplification and deletion rules can be applied [17]. For our purposes it is sufficient that tautological clauses and variant clauses are eliminated from the clause set during a derivation.

A decision procedure for DL* can be obtained using an ordering $\succ$ defined as follows. Let $>_d$ be an ordering on terms which is defined by $s >_d t$ if s is deeper than t, and every variable that occurs in t, occurs deeper in s. Then define $P(s_1, \ldots, s_n) \succ Q(t_1, \ldots, t_n)$ by $\{s_1, \ldots, s_n\} >_d^{mul} \{t_1, \ldots, t_n\}$, where $>_d^{mul}$ is the multiset extension of $>_d$.

Theorem 2 (Soundness, completeness, and termination) *Let φ be a formula of the core KARO logic and let $N = \mathrm{CL}_{\mathrm{DL}^*}(\Pi(\varphi))$. Then:*

1. Any derivation from N based on $\succ$ terminates in exponential time (in the size of the signature of N).

2. φ is unsatisfiable iff the empty clause can be derived from N.

Proof: Termination, soundness, and completeness is a consequence of Theorem 5.4 in [10]. In [10] it is stated that any derivation from a set N' in the class DL* terminates in double exponential time in the size of the signature of N'. This is basically due to the fact that there is a double exponential upper bound on the number of clauses derivable from N' and the fact that applications of the inference rules as well as redundancy elimination steps require only polynomial time in the size of the derived clause set. However, for clause sets obtained from the translation of formulae of the core KARO logic it is possible to obtain a single exponential upper bound on the number of derivable clauses. The complexity of the inference steps and redundancy elimination steps remains unchanged, thus providing us with the upper bound stated in our theorem. ■

It is straightforward to see that the satisfiability problem for formulae of the core KARO logic is PSPACE-complete. As pointed out in the proof of Theorem 2, the approach described above will require exponential space in the worst-case. A computationally space optimal decision procedure, based on translation and a refinement of the resolution calculus using a particular selection function instead of an ordering refinement, can be developed along the lines of [18]. This alternative decision procedure would require only polynomial space in the worst case.

4. Proof by Clausal Temporal Resolution

Here, we use the simple observation that the use of PDL in the KARO framework is very similar to the use of branching-time temporal logic. Thus, we attempt to use a simple CTL branching-time temporal logic to represent the dynamic component of the core KARO logic. We assume that formulae of the core KARO logic are normalised using the rewrite rules of Figure 1. Then dynamic operators are replaced by CTL formulae by the following rules: for example,

$\langle \mathtt{do}_i(a) \rangle \varphi$ is replaced by $\mathbf{E}\bigcirc(done_i^a \wedge \varphi)$ and
$[\mathtt{do}_i(a)]\varphi$ is replaced by $\mathbf{A}\bigcirc(done_i^a \Rightarrow \varphi)$

where $done_i^a$ is a propositional variable uniquely associated with agent i and atomic action a. Initially we work in the combination of CTL with multi-modal S5 with no interaction between these logics. The semantics of this logic is given for example in [2].

Formulae in the fusion of CTL and $\mathsf{S5}_{(\mathsf{m})}$ can rewritten into a normal form, called SNF$_{karo}$, that separates temporal and modal aspects (as is done in [19]). Formulae in SNF$_{karo}$ are of the general form $\mathbf{A}\square^* \bigwedge_i T_i$ where $\mathbf{A}\square^*$ is the universal relation (which can be defined in terms of the operators "everyone knows" and "always"; see for example [19]) and each T_i is a *clause* and must be of one of the following forms.

$$\begin{array}{rlr}
\mathbf{start} \Rightarrow \bigvee_{k=1}^{n} L_k & & (\textit{initial} \text{ clauses}) \\
\bigwedge_{j=1}^{m} L'_j \Rightarrow \mathbf{A}\bigcirc \bigvee_{k=1}^{n} L_k & & (\textit{step} \text{ clause}) \\
\bigwedge_{j=1}^{m} L'_j \Rightarrow \mathbf{E}\bigcirc (\bigvee_{k=1}^{n} L_k)_{\langle \mathsf{c_i} \rangle} & & (\textit{step} \text{ clauses}) \\
\bigwedge_{j=1}^{m} L'_j \Rightarrow \mathbf{A}\Diamond L & & (\textit{sometime} \text{ clauses}) \\
\bigwedge_{j=1}^{m} L'_j \Rightarrow \mathbf{E}\Diamond L_{\langle \mathsf{c_i} \rangle} & & (\textit{sometime} \text{ clauses}) \\
\mathbf{true} \Rightarrow \bigvee_{k=1}^{n} M_k^i & & (\mathbf{K}_i \text{ clauses}) \\
\mathbf{true} \Rightarrow \bigvee_{k=1}^{n} L_k & & (\text{literal clauses})
\end{array}$$

where L'_j, L_k, and L are literals and M_k^i are either literals, or modal literals involving the modal operator $\mathbf{K}_i$. Further, each $\mathbf{K}_i$ clause has at least one disjunct that is a modal literal. $\mathbf{K}_i$ clauses are sometimes known as *knowledge clauses*. Each step and sometime clause that involves the $\mathbf{E}$-operator is labelled by an index of the form $\langle \mathsf{c_i} \rangle$ similar to the use of Skolem constants in first-order logic. This index indicates a particular path and arises from the translation of formulae such as $\mathbf{E}(L\mathcal{U}L')$. During the translation to the normal form such formulae are translated into several $\mathbf{E}$ step clauses and an $\mathbf{E}$ sometime clause (which ensures that L' must actually hold). To indicate that all these clauses refer to the same path they are annotated with an index. The outer '$\mathbf{A}\square^*$' operator that surrounds the conjunction of clauses is usually omitted. Similarly, for convenience the conjunction is dropped and we consider just the set of clauses T_i. We denote the normalisation function into SNF$_{karo}$ by τ.

In the following we present a resolution based calculus for SNF$_{karo}$. In contrast to the translation approach described in the previous section, this calculus works directly on SNF$_{karo}$ formulae. The inference rules are divided into initial resolution rules, knowledge resolution rules, step resolution rules, and temporal resolution rules, which will be described in the following.

Initial Resolution. In the following, if L is a literal, then $\sim L$ denotes A if $L = \neg A$ and it denotes $\neg L$, otherwise. A literal clause may be resolved with an initial clause (IRES1) or two initial clauses may be resolved together (IRES2) as follows where C and D are disjunctions of literals.

$$[\text{IRES1}] \quad \begin{array}{l} \mathbf{true} \Rightarrow C \vee L \\ \mathbf{start} \Rightarrow D \vee \sim L \\ \hline \mathbf{start} \Rightarrow C \vee D \end{array}$$

$$[\text{IRES2}] \quad \begin{array}{l} \mathbf{start} \Rightarrow C \vee L \\ \mathbf{start} \Rightarrow D \vee \sim L \\ \hline \mathbf{start} \Rightarrow C \vee D \end{array}$$

Knowledge Resolution. During knowledge resolution we apply the following rules which are based on the modal resolution system introduced by Mints [20]. In general we may only apply a (knowledge) resolution rule between two literal clauses, a knowledge and a literal clause, or between two knowledge clauses relating to the same modal operator e.g. two $\mathbf{K}_1$ clauses.

$$[\text{KRES1}] \quad \begin{array}{l} \mathbf{true} \Rightarrow C \vee M \\ \mathbf{true} \Rightarrow D \vee \sim M \\ \hline \mathbf{true} \Rightarrow C \vee D \end{array}$$

$$[\text{KRES2}] \quad \begin{array}{l} \mathbf{true} \Rightarrow C \vee \mathbf{K}_i L \\ \mathbf{true} \Rightarrow D \vee \mathbf{K}_i \sim L \\ \hline \mathbf{true} \Rightarrow C \vee D \end{array}$$

$$[\text{KRES3}] \quad \begin{array}{l} \mathbf{true} \Rightarrow C \vee \mathbf{K}_i L \\ \mathbf{true} \Rightarrow D \vee \sim L \\ \hline \mathbf{true} \Rightarrow C \vee D \end{array}$$

$$[\text{KRES4}] \quad \begin{array}{l} \mathbf{true} \Rightarrow C \vee \neg \mathbf{K}_i L \\ \mathbf{true} \Rightarrow D \vee L \\ \hline \mathbf{true} \Rightarrow C \vee \mathrm{mod(D)} \end{array}$$

The function $\mathrm{mod(D)}$ used in KRES4 is defined on disjunctions D of literals or modal literals, as follows.

$$\begin{array}{rcl}
\mathrm{mod(A \vee B)} & = & \mathrm{mod(A)} \vee \mathrm{mod(B)} \\
\mathrm{mod}(\mathbf{K}_\mathrm{i}\mathrm{L}) & = & \mathbf{K}_i L \\
\mathrm{mod}(\neg\mathbf{K}_\mathrm{i}\mathrm{L}) & = & \neg\mathbf{K}_i L \\
\mathrm{mod(L)} & = & \neg\mathbf{K}_i \sim L
\end{array}$$

The resolution rule KRES4 requires explanation. Take KRES4 and push in the external $\mathbf{K}_i$ operator from the surrounding $\mathbf{A}\square^*$ operator into the second premise obtaining $\mathbf{true} \Rightarrow \neg\mathbf{K_i}\neg\mathbf{D} \vee \mathbf{K_i}\mathbf{L}$ where D is a disjunction of literals or modal literals. Since, in $\mathsf{S5}$, from axioms $\mathsf{4}$, $\mathsf{5}$ and D we have

$$\begin{array}{lrcl}
\vdash & \neg\mathbf{K}_i\mathbf{K}_i\varphi & \iff & \neg\mathbf{K}_i\varphi \quad \text{and} \\
\vdash & \neg\mathbf{K}_i\neg\mathbf{K}_i\neg\varphi & \iff & \mathbf{K}_i\neg\varphi,
\end{array}$$

so we can delete $\neg\mathbf{K}_i\neg$ from any of the disjuncts in D that are modal literals and obtain the required resolvent.

Finally we require the following rewrite rule which allows us to obtain the most comprehensive set of literal clauses for use during step and temporal resolution

$$[\text{KRES5}] \quad \begin{array}{l} \mathbf{true} \Rightarrow D \vee \mathbf{K}_i L_1 \vee \mathbf{K}_i L_2 \vee \ldots \vee \mathbf{K}_i L_n \\ \hline \mathbf{true} \Rightarrow D \vee L_1 \vee L_2 \vee \ldots \vee L_n \end{array}$$

where D is a disjunction of literals.

Step Resolution. 'Step' resolution consists of the application of standard classical resolution to formulae representing constraints at a particular moment in time, together with simplification rules for transferring contradictions within states to constraints on previous states.

Pairs of step clauses may be resolved using the (step resolution) rules SRES1, SRES2, and SRES3, while SRES4 allows to generate an extra global constraint once a contradiction within a state is found. The symbol $\mathbf{P}$ in SRES4 denotes either path operator.

$$\text{[SRES1]}\quad \frac{\begin{array}{c}P \Rightarrow \mathbf{A}\bigcirc(F \vee L)\\ Q \Rightarrow \mathbf{A}\bigcirc(G \vee {\sim}L)\end{array}}{P \wedge Q \Rightarrow \mathbf{A}\bigcirc(F \vee G)}$$

$$\text{[SRES2]}\quad \frac{\begin{array}{c}P \Rightarrow \mathbf{E}\bigcirc(F \vee L)_{\langle \mathsf{c_i}\rangle}\\ Q \Rightarrow \mathbf{A}\bigcirc(G \vee {\sim}L)\end{array}}{P \wedge Q \Rightarrow \mathbf{E}\bigcirc(F \vee G)_{\langle \mathsf{c_i}\rangle}}$$

$$\text{[SRES3]}\quad \frac{\begin{array}{c}P \Rightarrow \mathbf{E}\bigcirc(F \vee L)_{\langle \mathsf{c_i}\rangle}\\ Q \Rightarrow \mathbf{E}\bigcirc(G \vee {\sim}L)_{\langle \mathsf{c_i}\rangle}\end{array}}{P \wedge Q \Rightarrow \mathbf{E}\bigcirc(F \vee G)_{\langle \mathsf{c_i}\rangle}}$$

$$\text{[SRES4]}\quad \frac{Q \Rightarrow \mathbf{P}\bigcirc\mathbf{false}}{\mathbf{true} \Rightarrow {\sim}Q}$$

A step clause may be resolved with a literal clause (where G is a disjunction of literals) and any index is carried to the resolvent to give resolution rules SRES5 and SRES6.

$$\text{[SRES5]}\quad \frac{\begin{array}{c}P \Rightarrow \mathbf{A}\bigcirc(F \vee L)\\ \mathbf{true} \Rightarrow (G \vee {\sim}L)\end{array}}{P \Rightarrow \mathbf{A}\bigcirc(F \vee G)}$$

$$\text{[SRES6]}\quad \frac{\begin{array}{c}P \Rightarrow \mathbf{E}\bigcirc(F \vee L)_{\langle \mathsf{c_i}\rangle}\\ \mathbf{true} \Rightarrow (G \vee {\sim}L)\end{array}}{P \Rightarrow \mathbf{E}\bigcirc(F \vee G)_{\langle \mathsf{c_i}\rangle}}$$

This rule states that if, by satisfying Q in the last moment in time a contradiction is produced, then P must never be satisfied in *any* moment in time. The new constraint therefore represents $\mathbf{A}\Box^*{\sim}Q$.

Temporal Resolution. During temporal resolution the aim is to resolve one of the sometime clauses, $Q \Rightarrow \mathbf{P}\Diamond L$, with a set of clauses that together imply $\Box{\sim}L$ along the same path, for example a set of clauses that together have the effect of $F \Rightarrow \bigcirc\Box{\sim}L$. However the interaction between the '$\bigcirc$' and '$\Box$' operators makes the definition of such a rule non-trivial and further the translation to SNF$_{karo}$ will have removed all but the outer level of $\Box$-operators. So, resolution will be between a sometime clause and a *set* of clauses that together imply an $\Box$-formula that occurs on the same path, which will contradict the $\Diamond$-clause.

$$\text{[TRES1]}\quad \frac{\begin{array}{c}P \Rightarrow \mathbf{A}\bigcirc\mathbf{A}\Box L\\ Q \Rightarrow \mathbf{A}\Diamond{\sim}L\end{array}}{P \wedge Q \Rightarrow \mathbf{A}({\sim}P\mathcal{W}{\sim}L)}$$

$$\text{[TRES2]}\quad \frac{\begin{array}{c}P \Rightarrow \mathbf{A}\bigcirc\mathbf{A}\Box L\\ Q \Rightarrow \mathbf{E}\Diamond{\sim}L_{\langle \mathsf{c_i}\rangle}\end{array}}{P \wedge Q \Rightarrow \mathbf{E}({\sim}P\mathcal{W}{\sim}L)_{\langle \mathsf{c_i}\rangle}}$$

$$\text{[TRES3]}\quad \frac{\begin{array}{c}P \Rightarrow \mathbf{E}\bigcirc\mathbf{E}\Box L_{\langle \mathsf{c_i}\rangle}\\ Q \Rightarrow \mathbf{A}\Diamond{\sim}L\end{array}}{P \wedge Q \Rightarrow \mathbf{A}({\sim}P\mathcal{W}{\sim}L)}$$

$$\text{[TRES4]}\quad \frac{\begin{array}{c}P \Rightarrow \mathbf{E}\bigcirc\mathbf{E}\Box L_{\langle \mathsf{c_i}\rangle}\\ Q \Rightarrow \mathbf{E}\Diamond{\sim}L_{\langle \mathsf{c_i}\rangle}\end{array}}{P \wedge Q \Rightarrow \mathbf{E}({\sim}P\mathcal{W}{\sim}L)_{\langle \mathsf{c_i}\rangle}}$$

In each case the resolvent ensures that once Q has been satisfied, meaning that the eventuality $\Diamond{\sim}L$ must be satisfied on some or all paths, the conditions for triggering a $\Box$-formula are not allowed to occur, that is, P must be false, until the eventuality (${\sim}L$) has been satisfied. It may be surprising that resolving a $\mathbf{A}$-formula with a $\mathbf{E}$-formula in TRES3 results in a $\mathbf{A}$-formula. This is because the eventuality ${\sim}L$ must appear on *all* paths so similarly the resolvent will also hold on all paths.

Simplification and subsumption are also carried out during a derivation.

Theorem 3 (Soundness, completeness, and termination) *Let φ be a formula of the core KARO logic and let $N = \tau(\varphi)$. Then:*

1. *Any derivation from N terminates.*
2. *φ is unsatisfiable iff N has a refutation by the temporal resolution procedure described above.*

The proofs are analogous to those in [21, 19, 22].

5. Beyond the Core KARO Logic

In Sections 3 and 4 we have presented two methods for modal reasoning in a restricted core of the KARO logic. We will now consider whether and how each method can be extended to cover a larger fragment of the KARO logic.

In the full framework $\mathbf{O}_i\alpha$ and $\mathbf{A}_i\alpha$ are not the same. There we have $\mathbf{O}_i\alpha{=}\langle\mathsf{do}_i(\alpha)\rangle\top$, and $\mathbf{A}_i\alpha$ is as defined in Section 2. Consequently, we can extend the normalisation function defined by the rewrite rules in Figure 1 to reduce any formula φ with occurrences of $\mathbf{O}_i\alpha$, $\mathbf{A}_i\alpha$, or $[\mathsf{do}_i(\alpha)]\psi$ where α is a non-atomic action formula to a formula $\varphi{\downarrow}$ which is logically equivalent to φ and in the absence of the unbounded repetition operator $\varphi{\downarrow}$ contains no non-atomic action formulae.

In the translation approach the translation function π has to be modified such that $\pi(\mathbf{A}_i a, x) = c_i^a(x)$ where a is an atomic action, and c_i^a represents the relation $c_{(i,a)}$ in our semantics. In the clausal temporal resolution approach $\mathbf{A}_i\alpha$ is simply represented by propositional variables c_i^α uniquely associated with i and α. It seems an alternative for both approaches that would incorporate also a commitment operator could exploit the ideas of [23].

We have also excluded wishes in our presentation. In the full KARO framework, $\mathbf{W}_i^s$ is a KD modality. The incorporation of wishes into the translation approach presents no difficulties. The translation function π is extended by $\pi(\mathbf{W}_i^s\varphi, x) = \forall y\,(W_i(x,y) \rightarrow \pi(\varphi, y))$ and $\Pi(\psi)$ contains additional conjuncts $\forall x\,\exists y\, W_i(x,y)$ for every agent i, ensuring that the binary relations W_i are serial. For the clausal temporal resolution approach the addition of wishes to the core of KARO requires (i) an extension of the normal form which allows for clauses for the wishes of each agent, and (ii) additional sound and complete resolution rules for the KD modalities $\mathbf{W}_i^s$.

The implementability operator $\Diamond_i$ excluded from core KARO logic is one of the most interesting operators of KARO logic. Recall that the semantics of $\Diamond_i$ is defined by

$$\mathcal{M}, w \models \Diamond_i\varphi \text{ iff } \exists k \in \mathbb{N} \exists a_1, \dots, a_k \in \mathsf{Ac}_{\mathsf{at}}. \quad \mathcal{M}, w \models \mathbf{PracPoss}_i(a_1; \dots; a_k, \varphi)$$

where $\mathbf{PracPoss}_i(\alpha, \varphi)$ is an abbreviation for $\langle \mathsf{do}_i(\alpha)\rangle\varphi \wedge \mathbf{A}_i\alpha$. So, $\Diamond_i\varphi$ holds if we can find atomic actions $a_1, \dots, a_k$ such that agent i is able to perform the sequence $a_1; \dots; a_k$ and performing this sequence possibly leads to a situation in which φ is true. Intuitively, proving $\Diamond_i\varphi$ requires that we find a *plan* which might bring about a situation in which φ is true. In other words, the intention for including the implementability operator into KARO logic is to internalise the *planning problem* in the logic.

However, it turns out that this intuition is misleading. To give a precise analysis of the implementability operator, let us add a modal operator $\exists$ to our language with the following semantics.

$$\mathcal{M}, w \models \exists\varphi \text{ iff } \exists v \in W. \mathcal{M}, v \models \varphi$$

Furthermore, if $\varphi[\psi_1, \psi_2]$ is a formula containing subformula occurrences of ψ_1 and ψ_2, then by $\varphi[\psi_1', \psi_2']$ we denote the formula obtained by replacing in φ the subformula occurrences of ψ_1 and ψ_2 by the formulae ψ_1' and ψ_2', respectively.

Lemma 2 *Let $\varphi[\Diamond_i\psi, \Diamond_i\vartheta]$ be a formula of KARO logic with a positive subformula occurrence of $\Diamond_i\psi$ and a negative subformula occurrence of $\Diamond_i\vartheta$. Then*

1. If $\boldsymbol{A}_i\alpha$ is, by definition, equivalent to $\boldsymbol{O}_i\alpha$ (for every action α), then $\varphi[\Diamond_i\psi, \Diamond_i\vartheta]$ is satisfiable iff $\varphi[\exists\psi, \exists\vartheta]$ is satisfiable.

2. If $\boldsymbol{A}_i\alpha$ is not, by definition, equivalent to $\boldsymbol{O}_i\alpha$ (for every action α), then (i) $\varphi[\Diamond_i\psi, \Diamond_i\vartheta]$ is satisfiable iff $\varphi[\exists\psi, \Diamond_i\vartheta]$ is satisfiable, and (ii) the satisfiability of $\varphi[\exists\psi, \exists\vartheta]$ implies the satisfiability of $\varphi[\Diamond_i\psi, \Diamond_i\vartheta]$, but not vice versa.

Proof: This lemma follows from the fact that the existential quantification over atomic actions in the semantical definition of $\Diamond_i$ is essentially an existential quantification over all binary relations on the set W in $\mathcal{M}$. ■

Thus, for the core KARO logic, the implementability operator is simply the dual master modality and has little to do with the planning problem.

Consequently, in the translation approach we can simply extend the translation morphism π by

$$\pi(\Diamond_i\varphi, x) = \exists y\, \pi(\varphi, y).$$

For the SNF approach, no such simple solution exists. Although the language of SNF_{karo} contains with $\mathbf{A}\Box^*$ a combination of operators corresponding to the master modality, $\mathbf{A}\Box^*$ can only occur at one particular position, that is, surrounding a conjunction of clauses. The operators $\mathbf{A}\Box$ and the dual $\mathbf{E}\Diamond$, which can occur inside clauses, only quantify over states reachable via a temporal path, but not over states which are only reachable via an accessibility relation associated with one of the knowledge operators $\mathbf{K}_i$. The solution is (i) an embedding of the $\mathsf{S5}_{(\mathsf{m})}$ sublogic of KARO logic into PDL and (ii) an embedding of this logic into CTL (instead of an embedding into the fusion of CTL and $\mathsf{S5}_{(\mathsf{m})}$). For the first step we have to introduce additional 'knowledge actions' $a_{\mathbf{K}_i}$ replacing the modal operators $\mathbf{K}_i$ by $[\mathsf{do}_i(a_{\mathbf{K}_i})]$. Although the accessibility relations associated with these knowledge actions are not necessarily equivalence relation, we can enforce satisfiability equivalence by adding axioms similar to $\mathrm{Ax}(\mathbf{K}_i\varphi)$ to the translated formula. Since after the second step of this transformation we obtain a pure CTL formula, we can use the operator $\mathbf{E}\Diamond$ to translate $\Diamond_i$.

However, it is clear that the current semantical definition of $\Diamond_i$ fails to correspond to our intuitive understanding of implementability. A more accurate semantical definition restricts the choice of atomic actions $a_1, \dots, a_k$, which an agent i performs to bring about a situation where φ holds, to a particular finite set of actions, for example, the set of atomic actions occurring in the formula under consideration. So, if $\mathsf{Ac}_{\mathsf{at}}^{\psi}$ denotes the finite set of atomic actions occurring in a formula ψ, then the modified semantical definition could be as follows,

$$\mathcal{M}, w \models \Diamond_i\varphi \text{ iff } \exists k \in \mathbb{N}\, \exists a_1, \dots, a_k \in \mathsf{Ac}_{\mathsf{at}}^{\psi}. \quad \mathcal{M}, w \models \mathbf{PracPoss}_i(a_1; \dots; a_k, \varphi)$$

where ψ is a specific KARO formula under consideration. In this case the existential quantifier in the definition of $\Diamond_i\varphi$ can be replaced by a disjunction over all actions in $\mathsf{Ac}_{\mathsf{at}}^{\psi}$. Then $\Diamond_i\varphi$ can be embedded into CTL^* as $\varphi \vee \mathbf{E}(\bigvee_{a \in \mathsf{Ac}_{\mathsf{at}}^{\psi}}(c_i^a \wedge \bigcirc done_i^a)))\mathcal{U}\varphi)$. Although this formula is not in CTL, it can be rewritten into a satisfiability equivalent set of SNF_{karo} clauses making use of the additional expressiveness of SNF_{karo} clauses due to the index labels we can attach to step clauses.

Also the use of the unbounded repetition operation on actions is excluded from the core KARO logic we have considered. This operation is not first-order definable and thus cannot be dealt with by the translation approach. Unbounded repetition also presents problems for the clausal temporal resolution approach as we require that only atomic actions a occur in $[\mathsf{do}_i(a)]\varphi$ and $\mathbf{A}_i a$. In the presence of unbounded repetition we are not able to remove occurrences of $\alpha^\star$ or non-atomic action below unbounded repetition using the rules of Figure 1 or similar rewrite rules. However, one possibility which may be fruitful is to translate formulae such as $\langle \mathsf{do}_i(a^\star)\rangle\varphi$, where a is an atomic action, directly into CTL as $\varphi \vee \mathbf{E}\bigcirc(\mathbf{E}(done_i^a \mathcal{U}\varphi))$.

This would be further rewritten into the normal form SNF_{karo}.

In the full KARO framework interaction between the dynamic logic and knowledge, for example $\mathbf{K}_i[\mathsf{do}_i(\alpha)]\varphi \rightarrow [\mathsf{do}_i(\alpha)]\mathbf{K}_i\varphi$ is allowed. This is similar to the interaction axiom between linear-time temporal logic and $\mathsf{S5}$, $\mathbf{K}_i\bigcirc\varphi \rightarrow \bigcirc\mathbf{K}_i\varphi$, given in [2], known as synchrony and perfect recall and is known to make the proof of validity much more complex. For example in the single agent case allowing this interaction between propositional linear time temporal logic and $\mathsf{S5}$ turns the satisfiability problem from a PSPACE complete problem into a double exponential time complete problem [2]. A clausal temporal resolution calculus for linear-time temporal logic and $\mathsf{S5}$ with synchrony and perfect recall is presented in [24]. However, in many cases the addition of such interactions leads to undecidability [2], so care is needed in this area.

6. Conclusion

Although there exist a number of theories of rational agency which are formulated in the framework of combinations of modal logics, the work on practical proof methods for the expressive logics involved in these theories has been sparse. Examples are the tableaux based proof methods developed by Rao and Georgeff for propositional BDI logics [5], and the resolution based proof methods developed by Dixon, Fisher, and Wooldridge for temporal logics of knowledge [19]. In this paper we presented the current state of our attempt to provide proof methods for the logics of the KARO framework, whose expressiveness exceeds those of previous theories of rational agency.

The presentation of the proof methods in Sections 3 and 4, and the discussion in Section 5, shows that although our proof methods already cover an interesting core fragment of the KARO framework, there are still essential gaps. We believe that this is not a sign that our approach is insufficient, but due to the fact that combinations of interacting logic inherently pose difficult proof theoretical problems, which have not received the necessary attention.

One of the motivations for pursuing two different approaches at the same time is the fact that the strength of the approaches lies within different areas of the KARO framework. The translation approach allows a quite elegant treatment of the informational component of KARO. On the other hand, the clausal temporal resolution approach has a better potential to provide a complete calculus for the dynamic component of KARO.

A promising approach is the possibility of combining both proof methods. In [25] we present a combination of clausal temporal resolution (restricted to a linear time temporal logic) and the translation approach plus first-order resolution (restricted to extension of the multi-modal logic $\mathsf{K}_{(\mathsf{m})}$), and we were able to show soundness, completeness, and termination of this combination for a range of combined logics.

References

[1] M. Wooldridge, *Reasoning about Rational Agents*, MIT Press, 2000.

[2] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi, *Reasoning About Knowledge*, MIT Press, 1996.

[3] A. S. Rao and M. P. Georgeff, "Modeling agents withing a BDI-architecture," in *Proc. KR-91*, pp. 473–484, Morgan Kaufmann, 1991.

[4] B. van Linder, W. van der Hoek, and J.-J. Ch. Meyer, "Formalising motivational attitudes of agents: On preferences, goals and commitments," in *Intelligent Agents II*, vol. 1037 of *LNAI*, pp. 17–32. Springer, 1996.

[5] A. S. Rao and M. P. Georgeff, "Decision procedures for BDI logics," *J. Logic & Computat.*, vol. 8, no. 3, pp. 293–343, 1998.

[6] B. van Linder, W. van der Hoek, and J.-J. Ch. Meyer, "Formalizing abilities and opportunities of agents," *Fundamenta Informaticae*, vol. 34, no. 1,2, pp. 53–101, 1998.

[7] B. van Linder, W. van der Hoek, and J.-J. Ch. Meyer, "Communicating rational agents," in *Proc. KI-94*, vol. 861 of *LNAI*, pp. 202–213, Springer, 1994.

[8] J.-J. Ch. Meyer and W. van der Hoek, *Epistemic Logic for AI and Computer Science*, Cambridge University Press, 1995.

[9] W. van der Hoek, B. van Linder, and J.-J. Ch. Meyer, "Unravelling nondeterminism: On having the ability to choose (extended abstract)," in *Proc. AIMSA'94*, pp. 163–172, World Scientific, 1994.

[10] H. De Nivelle, R. A. Schmidt, and U. Hustadt, "Resolution-based methods for modal logics," *Logic Journal of the IGPL*, vol. 8, no. 3, pp. 265–292, 2000.

[11] U. Hustadt, *Resolution-Based Decision Procedures for Subclasses of First-Order Logic*, Ph.D. thesis, Universität des Saarlandes, Saarbrücken, Germany, 1999.

[12] U. Hustadt and R. A. Schmidt, "Using resolution for testing modal satisfiability and building models," To appear in the *SAT 2000* Special Issue of *J. Automated Reasoning*, 2001.

[13] R. A. Schmidt, "Decidability by resolution for propositional modal logics," *J. Automated Reasoning*, vol. 22, no. 4, pp. 379–396, 1999.

[14] H. De Nivelle, "Translation of S4 into GF and 2VAR," Unpublished manuscript, May 1999.

[15] H. J. Ohlbach, "Combining Hilbert style and semantic reasoning in a resolution framework," in *Proc. CADE-15*, vol. 1421 of *LNAI*, pp. 205–219, Springer, 1998.

[16] C. Fermüller, A. Leitsch, U. Hustadt, and T. Tammet, "Resolution decision procedures," in *Handbook of Automated Reasoning*. Elsevier, 2001.

[17] L. Bachmair and H. Ganzinger, "Resolution theorem proving," in *Handbook of Automated Reasoning*. Elsevier, 2001.

[18] L. Georgieva, U. Hustadt, and R. A. Schmidt, "Computational space efficiency and minimal model generation for guarded formulae," Submitted, 2001.

[19] C. Dixon, M. Fisher, and M. Wooldridge, "Resolution for temporal logics of knowledge," *J. Logic & Computat.*, vol. 8, no. 3, pp. 345–372, 1998.

[20] G. Mints, "Gentzen-type systems and resolution rules. Part I: Propositional logic," in *Proc. COLOG-88*, vol. 417 of *LNCS*, pp. 198–231, Springer, 1990.

[21] C. Dixon, M. Fisher, and A. Bolotov, "Resolution in a Logic of Rational Agency," in *Proc. ECAI 2000*, IOS Press, 2000.

[22] M. Fisher, C. Dixon, and M. Peim, "Clausal Temporal Resolution," *ACM Transactions on Computational Logic*, vol. 2, no. 1, 2001.

[23] R. A. Schmidt and D. Tishkovsky, "On calculi and kripke semantics for multi-agent systems within the karo framework," Submitted, 2001.

[24] C. Dixon and M. Fisher, "Clausal Resolution for Logics of Time and Knowledge with Synchrony and Perfect Recall," in *Proc. ICTL 2000*, pp. 43–52, 2000.

[25] U. Hustadt, C. Dixon, R. A. Schmidt, and M. Fisher, "Normal forms and proofs in combined modal and temporal logics," in *Proc. FroCoS'2000*, vol. 1794 of *LNAI*, pp. 73–87, Springer, 2000.