Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Research Report
RR-92-56

Integrating a Modal Logic of Knowledge into Terminological Logics

Armin Laux

December 1992

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum
für
Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern and Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Atlas Elektronik, Daimler-Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Mannesmann-Kienzle, SEMA Group, Siemens and Siemens-Nixdorf. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct *systems with technical knowledge and common sense* which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- ❑ Intelligent Engineering Systems
- ❑ Intelligent User Interfaces
- ❑ Computer Linguistics
- ❑ Programming Systems
- ❑ Deduction and Multiagent Systems
- ❑ Document Analysis and Office Automation.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Friedrich J. Wendl
Director

Integrating a Modal Logic of Knowledge into Terminological Logics

Armin Laux

DFKI-RR-92-56

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-8903 0 and IWT-9201).

Integrating a Modal Logic of Knowledge into Terminological Logics

Armin Laux

Deutsches Forschungszentrum für künstliche Intelligenz

Stuhlsatzenhausweg 3

W-6600 Saarbrücken 11, Germany

e-mail: laux@dfki.uni-sb.de

Abstract

If we want of group of autonomous agents to act and to cooperate in a world, each of them needs knowledge about this world, about the knowledge of other agents, and about his own knowledge. To describe such knowledge we introduce the language $\mathcal{ALC}_{\mathcal{K}}$ which extends the concept language $\mathcal{ALC}$ by a new operator $\Box_i$. Thereby, $\Box_i\varphi$ is to be read as "agent i knows φ". This knowledge operator is interpreted in terms of possible worlds. That means, besides the real world, agents can imagine a number of other worlds to be possible. An agent is then said to know a fact φ if φ is true in all worlds he considers possible.

In this paper we use an axiomatization of the knowledge operator which has been proposed by Moore. Thereby, knowledge of agents is interpreted such that (i) agents are able to reason on the basis of their knowledge, (ii) anything that is known by an agent is true, and (iii) if an agent knows something then he knows that he knows it. We will give tableaux-based algorithms for deciding whether a set of $\mathcal{ALC}_{\mathcal{K}}$ sentences is satisfiable, and whether such a set entails a given $\mathcal{ALC}_{\mathcal{K}}$ sentence.

Contents

1 Introduction

Most artificial intelligence (AI) research investigates how a single system can exhibit intelligent behaviour. However, the recognition that much human problem solving involves groups of people and the development of concurrent computers provoked interest in concurrency and distribution in AI ([BG88]). Thus, research on the field of distributed AI deals with the question how a group of autonomous intelligent systems, often called agents, can cooperate to solve complex problems (see, e.g., [Huh87, BG88, GH89]). Thereby, the representation and use of knowledge is one of the most important foundations. Hence, if we want a group of autonomous agents to act and to cooperate in a world we have to equip them with a knowledge representation and reasoning component. Within this component the agents' knowledge about the world, about knowledge of other agents, and about their own knowledge should be represented.

Since the work of Hintikka [Hin62], modal logics have been widely accepted to be an adequate formalism for representing knowledge of an agent. The intuitive idea here is that besides the real world, agents can imagine a number of other worlds to be possible. An agent is then said to *know* a fact φ if φ is true in all worlds he thinks possible. For example, an agent knows that there is a monster of Loch Ness if there is such a monster in all worlds he considers possible. In order to express the knowledge of an agent a, in this approach a binary operator $KNOW(a, \varphi)$ is used, where φ is a formula over some logical language $\mathcal{L}$. Now, if we want to devise a formalism for representing knowledge of agents we have to take two decisions. Firstly, we have to choose a certain logical representation language $\mathcal{L}$ to describe the knowledge of agents. Secondly, we have to decide what the general properties of knowledge are we want this formalism to capture. That means, we have to give an axiomatization of the $KNOW$ operator.

When investigating multi-agent applications to model shipping departments and loading docks, we had to equip agents with knowledge about, e.g., trucks and goods. For the representation of such kind of knowledge about a world, terminological logics provide a structured and adequate formalism. Thus, we will use them as representation language $\mathcal{L}$ to describe our agents' world knowledge. Terminological logics are based on the work of Brachman and Schmolze [BS85] and can be used to define the relevant concepts of a particular problem domain. Starting with atomic concepts (unary predicates) and roles (binary predicates), one thereby defines complex concepts with the help of operators provided by a so-called concept language. In addition, objects can be associated with concepts while relationships between objects can be defined via roles.

Terminological logics provide a well-investigated and decidable subclass of first-order logics and can, e.g., be used to represent facts like "each truck is a vehicle" or "John owns a vehicle which is a truck". However, they do not provide an adequate formalism to represent what agents know about the knowledge of other agents and, especially, about their own knowledge. Additionally, terminological logics do not provide a formalism for representing an axiomatization of the represented knowledge as, e.g., if an agent knows φ, then he knows that he knows φ ("positive introspection"). In this paper we will introduce a formalism which overcomes both problems while maintaining the adequate representation of the agents' world knowledge.

In the past, a lot of different axiomatizations of the *KNOW* operator have been given (see, e.g., [Moo80, Moo85, Hal86, HM90, MvdHV91a, MvdHV91b, HM92]). But none of these axiomatizations has been accepted to be the "ultimate" logical approach; each of them is more adequate in some domains and less adequate in some others. So, what are the relevant properties of knowledge for agents in applications like shipping departments and loading docks? We feel, that the properties of knowledge Moore described in his seminal work [Moo80, Moo85] are adequate in these applications, since he argued these properties to be relevant to planning and acting. The first of these properties is that anything that is known by an agent must be true. For example, if an agent knows that a certain truck can transport gasoline this must be true. Otherwise, he must not assign this truck for a gasoline transportation order. Secondly, we assume that, if an agent knows something, then he knows that he knows it. This principle is needed especially for agents to planning. Suppose an agent plans to achieve a goal and therefore needs to know whether a certain truck can transport gasoline and whether John owns this truck. If he already knows the first fact to be true and has to perform some action *act* to find out whether the second fact is also true, then he needs to know whether he still knows that the truck can transport gasoline after performing *act*. Finally, the most important fact about knowledge we want to capture is, of course, that agents can reason on the basis of their knowledge. For example, suppose agent a knows that each gasoline truck can transport gasoline and he knows John to own truck-1 which is a gasoline truck. In this case, agent a should be able to conclude that John's truck truck-1 can be used to transport gasoline, and thus may negotiate with John for a gasoline transportation order.

Now we have decided which representation language to use for the knowledge of agents, and what the relevant properties of knowledge are. But how can we bring both things together, and how can we overcome the above mentioned problems which appear when using terminological logics as representation language $\mathcal{L}$?

In this paper we will present the language $\mathcal{ALC}_{\mathcal{K}}$ which extends the concept language $\mathcal{ALC}$ by a new operator $\Box_i$ for each agent i. These new operators are interpreted in terms of possible worlds and $\Box_i\varphi$ represents the fact "agent i knows φ". The extended language $\mathcal{ALC}_{\mathcal{K}}$ provides an adequate formalism for both, to represent the agents' world knowledge and to represent what agents know about the knowledge of other agents and about their own knowledge. For example, the fact "agent a knows that agent b does not know φ" is represented by $\Box_a\neg\Box_b\varphi$. Furthermore, the knowledge operator can be interpreted w.r.t. a given axiomatization as, e.g., positive introspection. We will show that the resulting modal logic is decidable when using Moore's axiomatization of knowledge.

In Section 2 we will formally introduce syntax and semantics of the concept language $\mathcal{ALC}$ and of its extension $\mathcal{ALC_K}$. Of course, we are not only interested in the representation of knowledge by a set of $\mathcal{ALC_K}$-axioms, but we want to test the represented knowledge on satisfiability and are interested in computing logical consequences. Thus, in Sections 3 and 4 we will present an algorithm for deciding whether a given set of $\mathcal{ALC_K}$-axioms is satisfiable. Finally, in Section 5, we will show how to decide whether an $\mathcal{ALC_K}$-axiom is a logical consequence of a set of given $\mathcal{ALC_K}$-axioms.

2 Syntax and Semantics of $\mathcal{ALC_K}$

In this section we will formally introduce the language $\mathcal{ALC_K}$ which extends the concept language $\mathcal{ALC}$ by a new operator $\Box_i$ for each agent i. Thereby, $\Box_i\varphi$ can be read as "agent i knows φ". Syntax and semantics of $\mathcal{ALC}$ and $\mathcal{ALC_K}$ are given in Subsections 2.1 and 2.2, respectively.

2.1 The Concept Language $\mathcal{ALC}$

Terminological logics provide two formalisms to describe a problem domain: a terminological formalism to represent taxonomical knowledge by defining concepts, which can be seen as sets of objects, and an assertional formalism which can be used to describe concrete objects. Therefore, one starts with a set of *atomic concepts* (unary predicates) and a set of *roles* (binary predicates).

In the concept language $\mathcal{ALC}$ *concepts* are built up from atomic concepts, the *top concept* $\top$, the *bottom concept* $\bot$, and roles inductively by:

1. Each atomic concept, $\top$, and $\bot$ are concepts.
2. If C and D are concepts and R is a role, then
   (a) $C \sqcap D$ (*concept conjunction*),
   (b) $C \sqcup D$ (*concept disjunction*),
   (c) $\neg C$ (*concept negation*),
   (d) $\forall R.C$ (*value restriction*), and
   (e) $\exists R.C$ (*exists restriction*)

   are concepts.

An interpretation I is a function over some non-empty domain Δ^I which maps each atomic concept C to a subset C^I of Δ^I, each role R to a subset R^I of $\Delta^I \times \Delta^I$, $\top$ to Δ^I, and $\bot$ to $\emptyset$. Furthermore, $\sqcap$ is interpreted as set intersection, $\sqcup$ as set union, and $\neg$ as set complement w.r.t. Δ^I. The value and the exists restrictions are interpreted by

$$[\forall R.C]^I = \{d \in \Delta^I \mid \forall d' : (d, d') \in R^I \rightarrow d' \in C^I\}$$
$$[\exists R.C]^I = \{d \in \Delta^I \mid \exists d' : (d, d') \in R^I \wedge d' \in C^I\}$$

For example, if *man* and *truck* are atomic concepts and *owns* is a role we can define the concept of men who own a truck by $man \sqcap \exists\, owns.truck$.

The taxonomical knowledge of a problem domain can be defined by an *$\mathcal{ALC}$-TBox* (*terminology*), which consists of a finite set of terminological axioms. A *terminological axiom* is of the form

- $C = D$ (concept equivalence) or
- $C \neq D$ (negated concept equivalence)

where C, D are concepts. An interpretation I satisfies $C = D$ iff $C^I = D^I$ and it satisfies $C \neq D$ iff $C^I \neq D^I$. An interpretation I satisfies an $\mathcal{ALC}$-TBox $\mathcal{T}$ iff I satisfies each axiom in $\mathcal{T}$. For example, if *carrier*, *person*, and *truck* are concepts and *owns* is a role, we can define exactly the persons who own a truck to be a carrier by

$$carrier = person \sqcap \exists\, owns.truck.$$

The assertional formalism of $\mathcal{ALC}$ allows to introduce concrete objects by stating that they are instances of concepts and roles: If a is an object and C a concept, then $a : C$ is a *concept instance*. If a and b are objects and R is a role, then aRb is a *role instance*. Concept instances and role instances are called *assertional axioms*, and a finite set of assertional axioms is called an *$\mathcal{ALC}$-ABox*. An interpretation I maps objects to elements of its domain Δ^I and satisfies $a : C$ iff $a^I \in C^I$, and aRb iff $(a^I, b^I) \in R^I$. We assume that different objects in an ABox are mapped to different elements in Δ^I (*unique name assumption*). An interpretation I satisfies an $\mathcal{ALC}$-ABox $\mathcal{A}$ iff I satisfies each axiom in $\mathcal{A}$. As an example, if *John* and *truck-1* are objects, we can express that John owns truck-1 which is a truck by the assertional axioms

John owns truck-1 and *truck-1 : truck*.

Thus, we can describe the relevant concepts of a problem domain by terminological axioms, i.e., by an $\mathcal{ALC}$-TBox, and properties of objects as well as relations between them by assertional axioms, i.e., by an $\mathcal{ALC}$-ABox. We say an interpretation I satisfies a set $Ax_1, \ldots, Ax_n$ of terminological and assertional axioms iff I satisfies each of these axioms. We then write $I \models Ax_1, \ldots, Ax_n$.

For sake of simplicity we will sometimes use the expressions $C \sqsubseteq D$ and $C \not\sqsubseteq D$ where C and D are concepts. An interpretation I satisfies $C \sqsubseteq D$ iff $C^I \subseteq D^I$ and it satisfies $C \not\sqsubseteq D$ iff $C^I \not\subseteq D^I$. The next lemma states that these expressions are abbreviations for certain terminological axioms.

Lemma 2.1 *Let C and D be concepts, and let I be an interpretation. Then*

1. *I satisfies $C \sqsubseteq D$ iff I satisfies $\neg C \sqcup D = \top$.*

2. *I satisfies $C \not\sqsubseteq D$ iff I satisfies $\neg C \sqcup D \neq \top$.*

Proof: For 1., firstly suppose I satisfies $C \sqsubseteq D$. Then for each element d in Δ^I either $d \in [\neg C]^I$ or both $d \in C^I$ and $d \in D^I$ holds. That means, I satisfies $\neg C \sqcup (C \sqcap D) = \top$ what can be simplified to $\neg C \sqcup D = \top$. Conversely, suppose I satisfies $\neg C \sqcup D = \top$. Then for each element $d \in \Delta^I$ either $d \notin C^I$ or $d \in D^I$ holds. Thus, from $d \in C^I$ follows $d \in D^I$, i.e., $C^I \subseteq D^I$. The proof of 2. is analogous. □

For example, if *truck* and *vehicle* are concepts we can define each truck to be a vehicle by *truck* $\sqsubseteq$ *vehicle*, what is an abbreviation for $\neg$*truck* $\sqcup$ *vehicle* $= \top$.

2.2 The Extended Language $\mathcal{ALC}_\mathcal{K}$

Now we will introduce the language $\mathcal{ALC}_\mathcal{K}$ which extends $\mathcal{ALC}$ by a new operator $\Box_i$ for each agent i.[1] We allow these operators in front of terminological and assertional axioms. Thereby, the operator $\Box_i$, read as "agent i knows", allows us to express the knowledge agent i has about the world, about knowledge of other agents, and about his own knowledge. We extend the definition of terminological and assertional axioms as follows.

- If *TA* is a terminological axiom, then $\Box_i$ *TA* and $\neg\Box_i$*TA* are terminological axioms as well.
- If *CI* is a concept instance, then $\Box_i$ *CI* and $\neg\Box_i$ *CI* are concept instances as well.
- If *RI* is a role instance, then $\Box_i$ *RI* is a role instance as well.

These extended assertional and terminological axioms are called $\mathcal{ALC}_\mathcal{K}$*-axioms* and can, e.g., be used to state that agent i knows that each truck is a vehicle by

$$\Box_i\ (truck \sqsubseteq vehicle).$$

[1] In the following, we will abbreviate agents by numbers, and we suppose only a finite number of agents to be given.

Analogously, the $\mathcal{ALC_K}$-axioms $\Box_i \neg \Box_j$ (*vehicle-1 : truck*) and $\Box_i \neg \Box_i$ (*vehicle-1 : truck*) are to be read as "agent i knows that agent j doesn't know that vehicle-1 is a truck" and "agent i knows that he doesn't know truck-1 to be a truck", respectively. It is not reasonable to allow $\Box_i$ immediately in front of concepts, since the fact "agent i knows C" makes not much sense if C is a concept.

We will interpret the operators $\Box_i$ in terms of *possible worlds*, i.e., besides the real world there exist a number of worlds agents consider to be possible. If agent i considers world w' possible at world w, we say w' is *accessible from* w by agent i. The *accessibility relation* of agent i is given by all pairs (w, w') such that w' is accessible from w by agent i. Since different worlds are possible in our approach, the interpretation of concepts and roles in $\mathcal{ALC_K}$-axioms depends on the world we are currently speaking of. That means, in different worlds concepts may contain different objects and roles may contain different pairs of objects. This will be expressed by taking an additional parameter, the *world parameter*, into consideration when interpreting concepts and roles. Formally, we use the notion of a *K-interpretation* K_I which consists of a non-empty domain Δ^{K_I} and maps objects to elements in Δ^{K_I}, atomic concepts to subsets of $\Delta^{K_I} \times \mathcal{W}$, $\top$ to $\Delta^{K_I} \times \mathcal{W}$, $\bot$ to $\emptyset \times \mathcal{W}$, and roles to subsets of $\Delta^{K_I} \times \Delta^{K_I} \times \mathcal{W}$. Furthermore, $\sqcap$ is interpreted as set intersection, $\sqcup$ as set union, and $\neg$ as set complement w.r.t. $\Delta^{K_I} \times \mathcal{W}$, and the value and exists restrictions are interpreted by

$$[\forall R.C]^{K_I} = \{(d, w) \mid (d', w) \in C^{K_I} \text{ for each } d' \text{ with } (d, d', w) \in R^{K_I}\}$$
$$[\exists R.C]^{K_I} = \{(d, w) \mid (d', w) \in C^{K_I} \text{ for some } d' \text{ with } (d, d', w) \in R^{K_I}\}.$$

Definition 2.2 *A Kripke structure K is a triple $(\mathcal{W}, \Gamma, K_I)$. Thereby, $\mathcal{W}$ is a non-empty set of worlds, Γ is a finite set of accessibility relations, one accessibility relation γ_i for each agent i, and K_I is a K-interpretation.*

The *satisfiability* of an $\mathcal{ALC_K}$-axiom F *in a Kripke structure* $K = (\mathcal{W}, \Gamma, K_I)$ *and a world* $w \in \mathcal{W}$, written as $K, w \models F$, is recursively defined by:

$$\begin{array}{lll}
K, w \models C = D & \text{iff} & \{d \mid (d, w) \in C^{K_I}\} = \{d \mid (d, w) \in D^{K_I}\} \\
K, w \models C \neq D & \text{iff} & \{d \mid (d, w) \in C^{K_I}\} \neq \{d \mid (d, w) \in D^{K_I}\} \\
K, w \models a : C & \text{iff} & (a, w) \in C^{K_I} \\
K, w \models aRb & \text{iff} & (a, b, w) \in R^{K_I} \\
K, w \models \Box_i G & \text{iff} & K, w' \models G \text{ for each world } w' \text{ with } (w, w') \in \gamma_i \\
K, w \models \neg\Box_i G & \text{iff} & \text{there is a world } w' \text{ with } (w, w') \in \gamma_i \text{ and } K, w' \not\models G
\end{array}$$

where G is an $\mathcal{ALC_K}$-axiom, C, D are concepts, a, b are objects, and R is a role.

A set $F_1, \ldots, F_n$ of $\mathcal{ALC_K}$-axioms is *satisfiable* iff there exists a Kripke structure $K = (\mathcal{W}, \Gamma, K_I)$ and a world $w_0 \in \mathcal{W}$ such that $K, w_0 \models F_i$ for $i = 1, \ldots, n$. We then write $K \models F_1, \ldots, F_n$.

Note, that we do not allow axioms of the form $\neg\Box_i(aRb)$. The reason for this restriction is that such axioms would be equivalent to stating that there exists a world in which the role instance aRb does *not* hold. And it is not yet clear how to treat negation of roles in terminological logics.

In the following we will use the notion *modality* to denote (negated) indexed $\Box$ operators, and $\mathcal{ALC}_\mathcal{K}$-axioms without any modalities are called $\mathcal{ALC}$*-axioms*. For example, the $\mathcal{ALC}_\mathcal{K}$-axiom $\Box_i\neg\Box_j(\textit{vehicle-1} : \textit{truck})$ contains the modalities $\Box_i$ and $\neg\Box_j$, and the $\mathcal{ALC}_\mathcal{K}$-axiom *vehicle-1 : truck* is an $\mathcal{ALC}$-axiom.

3 Testing Satisfiability of $\mathcal{ALC}_\mathcal{K}$-axioms

Using $\mathcal{ALC}_\mathcal{K}$-axioms, a "real world" and knowledge of agents can be defined as follows. The real world is given by a finite set of $\mathcal{ALC}$-axioms, and the knowledge of agent i is given by a finite set of $\mathcal{ALC}_\mathcal{K}$-axioms with the leading operator $\Box_i$. Of course, we do not only want to represent a world and knowledge of agents, but we are interested in algorithms to test (i) consistency of the represented facts, i.e., whether a given set of $\mathcal{ALC}_\mathcal{K}$-axioms is satisfiable, and (ii) whether an $\mathcal{ALC}_\mathcal{K}$-axiom is a logical consequence of a given set of $\mathcal{ALC}_\mathcal{K}$-axioms. In this section we will give an algorithm for testing satisfiability of a set of $\mathcal{ALC}_\mathcal{K}$-axioms. Building upon this we will show how to decide whether of not an $\mathcal{ALC}_\mathcal{K}$-axiom is a logical consequence from a set of $\mathcal{ALC}_\mathcal{K}$-axioms in Section 5.

3.1 The Frame Algorithm

We will now present an algorithm for testing satisfiability of a set $F_1, \ldots, F_n$ of $\mathcal{ALC}_\mathcal{K}$-axioms. To keep notation simple we use $\Diamond_i F$ as an abbreviation for $\neg\Box_i\neg F$, and transform $\mathcal{ALC}_\mathcal{K}$-axioms into negation normal form. An $\mathcal{ALC}_\mathcal{K}$-axiom (concept) is in *negation normal form* iff in the axiom (concept) negation signs occur immediately in front of atomic concepts only. Concepts can be transformed into an equivalent negation normal form by the rules

$$\begin{array}{lllllllll} \neg\neg C & \rightarrow & C & \neg(C \sqcap D) & \rightarrow & \neg C \sqcup \neg D & \neg(\forall R.C) & \rightarrow & \exists R.\neg C \\ \neg\top & \rightarrow & \bot & \neg(C \sqcup D) & \rightarrow & \neg C \sqcap \neg D & \neg(\exists R.C) & \rightarrow & \forall R.\neg C \\ \neg\bot & \rightarrow & \top & & & & & & \end{array}$$

where C is a concept and R is a role (see, e.g., [Hol90]). Building upon this it is easy to verify that $\mathcal{ALC}_\mathcal{K}$-axioms can be transformed into an equivalent negation normal form by the rules

$$\begin{array}{rclrclrcl} \neg\neg F & \to & F & \neg\Box_i F & \to & \Diamond_i\neg F & \neg(C = D) & \to & C^* \neq D^* \\ & & & \neg\Diamond_i F & \to & \Box_i\neg F & \neg(C \neq D) & \to & C^* = D^* \\ & & & & & & \neg(a : C) & \to & a : [\neg C]^* \end{array}$$

where F is an $\mathcal{ALC_K}$-axiom, C, D are concepts, a is an object, and C^*, D^*, and $[\neg C]^*$ are the negation normal forms of the concepts C, D, and $\neg C$, respectively. For example, the negation normal form of the $\mathcal{ALC_K}$-axiom

$$\neg\Box_i\big(\neg(A \sqcup B) = \neg(\forall R.C)\big) \qquad \text{is} \qquad \Diamond_i\big((\neg A \sqcap \neg B) \neq (\exists R.\neg C)\big).$$

In the following we suppose each $\mathcal{ALC_K}$-axiom to be given in negation normal form. It is easy to verify that the negation normal forms of the abbreviations $C \sqsubseteq D$ and $C \not\sqsubseteq D$ are given by $C^* \not\sqsubseteq D^*$ and $C^* \sqsubseteq D^*$, respectively.

By definition, a set $F_1, \ldots, F_n$ of $\mathcal{ALC_K}$-axioms is satisfiable iff there exists a Kripke structure K such that $K \models F_1, \ldots, F_n$. Of course, we are not interested in arbitrary Kripke structures to satisfy $F_1, \ldots, F_n$, but only in Kripke structures which interpret the knowledge operators $\Box_i$ in the sense of Moore, i.e., which satisfiy the three properties of the knowledge operator which have been introduced in Section 1. We therefore introduce the notion of $S4$ Kripke structures.

Definition 3.1 *A set $F_1, \ldots, F_n$ of $\mathcal{ALC_K}$-axioms is $S4$-satisfiable iff there exists a Kripke structure $K = (\mathcal{W}, \Gamma, K_I)$ which satisfies $F_1, \ldots, F_n$ and has the following properties:*

(P1) if $K, w \models \Box_i F$ then $K, w \models F$

(P2) if $K, w \models \Box_i F$ then $K, w \models \Box_i\Box_i F$

for each $\mathcal{ALC_K}$-axiom F, for each agent i, and for each world $w \in \mathcal{W}$. A Kripke structure which satisfies (P1) and (P2) is called $S4$ Kripke structure.

Property (P1) corresponds to "anything that is known by an agent must be true" and (P2) to "if an agent knows something, then he knows that he knows it". The third property, "agents must be able to reason on the basis of their knowledge", is guaranteed by choosing Kripke structures for the representation of knowledge (for details, see [Moo80]).

It is a well-known fact that K is an $S4$ Kripke structure if the accessibility relation γ_i of each agent i is reflexive and transitive (see, e.g., [Che80, HC68]).

To formulate a calculus for the frame algorithm we introduce the notions of labelled $\mathcal{ALC_K}$-axioms and of a world constraint system. A *labelled $\mathcal{ALC_K}$-axiom* consists of an $\mathcal{ALC_K}$-axiom F together with a label w, written as $F \,||\, w$. Thereby, w is a constant which represents a world in which F holds. A *world constraint* is either a labelled

1. $W \rightarrow_{\Box} \{F \,||\, w\} \cup W$

   if $\Box_i F \,||\, w$ is in W, and $F \,||\, w$ is not in W.

2. $W \rightarrow_{\Diamond} \{w \bowtie_i v, F \,||\, v, G_1 \,||\, v, \Box_i G_1 \,||\, v, \ldots, G_n \,||\, v, \Box_i G_n \,||\, v\} \cup W$

   if $\Diamond_i F \,||\, w$ is in W, there is no label $\tilde{w}$ in W such that $\Diamond_i F \,||\, w$ is covered by $\tilde{w}$, $\Box_i G_1 \,||\, w, \ldots, \Box_i G_n \,||\, w$ are exactly the world constraints in W of the form $\Box_i G \,||\, w$, and v is a new label.

3. $W \rightarrow_{\circ} \{w \bowtie_i v\} \cup W$

   if $\Diamond_i F \,||\, w$ is in W, $\Diamond_i F \,||\, w$ is covered by label v in W, and for each label $\tilde{w}$ in W such that $\Diamond_i F$ is covered by $\tilde{w}$, $w \bowtie_i \tilde{w}$ is not in W.

Figure 1: Propagation rules of the frame algorithm.

$\mathcal{ALC}_{\mathcal{K}}$-axiom or a term $w \bowtie_i w'$. The constants w and w' represent worlds and $\bowtie_i$ represents the accessibility relation of agent i. A *world constraint system* is a finite, non-empty set of world constraints.

A Kripke structure K satisfies a world constraint system W iff for each label w in W there is a world $w^K \in \mathcal{W}$ such that (i) $K, w^K \models F$ for each world constraint $F \,||\, w$ in W and (ii) $(w^K, v^K) \in \gamma_i$ for each world constraint $w \bowtie_i v$ in W. A world constraint system W is ($S4$-)*satisfiable* iff there exists an ($S4$) Kripke structure which satisfies W.

For testing $S4$-satisfiability of a set $F_1, \ldots, F_n$ of $\mathcal{ALC}_{\mathcal{K}}$-axioms, we firstly translate them into a world constraint system. The world constraint system W is *induced by* $F_1, \ldots, F_n$ iff $W = \{F_1 \,||\, w_0, \ldots, F_n \,||\, w_0\}$, where w_0 is a new constant (which represents the real world). Obviously, $F_1, \ldots F_n$ are $S4$-satisfiable iff W is $S4$-satisfiable.

For testing $S4$-satisfiability of a world constraint system W we will use the three propagation rules which are given in Figure 1 and which successively add new world constraints to W. The main idea behind these rules is as follows. Firstly, we add as much information about w_0 to W as possible. That means, if there is a world constraint $\Box_i F \,||\, w_0$ in W we extend the world constraint system W by $F \,||\, w_0$. Then, if there is a labelled $\mathcal{ALC}_{\mathcal{K}}$-axiom $\Diamond_i F \,||\, w_0$ in W, we "jump" to a new label, say w, while inheriting as much information as possible from w_0 to w. For example, if $\Box_i G \,||\, w_0$ is a world constraint in W, we add $G \,||\, w$ to W. Now, w becomes the current label and is handled as described for w_0 above. This process is iterated until no more propagation rule is applicable (cf., e.g., [Fit83, HC68, Gor92]).

Let us now have a closer look at the three propagation rules. Firstly, if there is a labelled $\mathcal{ALC}_{\mathcal{K}}$-axiom $\Box_i F \,||\, w$ in W, then each $S4$ Kripke structure K which satisfies $\Box_i F \,||\, w$ also satisfies $F \,||\, w$ (because of property (P1) of $S4$ Kripke structures). Thus, the $\rightarrow_\Box$ rule adds the world constraint $F \,||\, w$ to W whenever there is a world constraint $\Box_i F \,||\, w$ in W and $F \,||\, w$ is not in W.

Secondly, suppose the world constraint system W contains a labelled $\mathcal{ALC}_\mathcal{K}$-axiom $\Diamond_i F \,||\, w$. If there exists an $S4$ Kripke structure $K = (\mathcal{W}, \Gamma, K_I)$ which satisfies W, then there is a world, say v, in $\mathcal{W}$ such that $(w^K, v^K) \in \gamma_i$ and $K, v^K \models F$. Furthermore, for each labelled $\mathcal{ALC}_\mathcal{K}$-axiom $\Box_i G \,||\, w$ in W holds

1. $K, v^K \models G$ since $(w^K, v^K) \in \gamma_i$.

2. $K, w^K \models \Box_i \Box_i G$ because of Property (P2) of $S4$ Kripke structures, and thus $K, v^K \models \Box_i G$ since $(w^K, v^K) \in \gamma_i$.

Let now $\Diamond_i F \,||\, w$ be a world constraint in W, and let $\Box_i G_1 \,||\, w, \ldots, \Box_i G_n \,||\, w$ be exactly the world constraints of the form $\Box_i G \,||\, w$ in W. Then, the $\rightarrow_\Diamond$ rule adds the world constraints $w \bowtie_i v$, $F \,||\, v$, $G_1 \,||\, v$, $\Box_i G_1 \,||\, v, \ldots, G_n \,||\, v$, $\Box_i G_n \,||\, v$ to W, where v is a new label.

Unfortunately, if we use the $\rightarrow_\Diamond$ rule as described above it may be applicable to a world constraint system W an infinite number of times. Consider, for example, the world constraint system W which is given by $\{F \,||\, w, \Diamond_1 F \,||\, w, \Box_1 \Diamond_1 F \,||\, w\}$. Since the $\mathcal{ALC}_\mathcal{K}$-axioms $\Diamond_1 F \,||\, w$ and $\Box_1 \Diamond_1 F \,||\, w$ are in W, an application of the $\rightarrow_\Diamond$ rule would introduce a new label v, and extend W by $\{w \bowtie_1 v, F \,||\, v, \Diamond_1 F \,||\, v, \Box_1 \Diamond_1 F \,||\, v\}$. In this case we would have "duplicated" our world constraint system, and the same duplication step could be applied to $\Diamond_1 F \,||\, v$, and so on. That means, an infinite number of world constraints would be introduced into W. To overcome this problem we introduce the notion of a covering.

Definition 3.2 *Given a world constraint system W and a label v in W, the $\mathcal{ALC}_\mathcal{K}$-axiom $\Diamond_i F \,||\, w$ is* covered *by label v (w.r.t. W) iff W contains the world constraints $F \,||\, v$ and for each world constraint $\Box_i G \,||\, w$ in W it contains both $G \,||\, v$ and $\Box_i G \,||\, v$.*

In the above world constraint system $W = \{F \,||\, w, \Diamond_1 F \,||\, w, \Box_1 \Diamond_1 F \,||\, w\}$, the label w is covered by itself. Therefore, instead of generating a new world in which F, $\Diamond_1 F$, and $\Box_1 \Diamond_1 F$ holds, the world constraint $w \bowtie_1 w$ is added to W. This case is handled by the $\rightarrow_\circ$ rule.

Building upon these three propagation rules we define the *frame algorithm* which is given in Figure 2. It has a world constraint system W as input which is induced by a set of $\mathcal{ALC}_\mathcal{K}$-axioms to be tested on $S4$-satisfiability. Starting with W it constructs a chain $W = W_0 \rightarrow_1 W_1 \rightarrow_2 \ldots \rightarrow_n W_n$ such that there is no more propagation rule applicable to W_n and $\rightarrow_i \in \{\rightarrow_\Box, \rightarrow_\Diamond, \rightarrow_\circ\}$. Thereby, the $\rightarrow_\Box$ rule has to be applied as often as possible before applying the $\rightarrow_\Diamond$ or the $\rightarrow_\circ$ rule. Output of the frame

1. Let $i := 0$ and let W_0 be the input world constraint system.

2. While there is a propagation rule in $\{\rightarrow_\Box, \rightarrow_\Diamond, \rightarrow_\circ\}$ applicable to W_i do

   (a) if $\rightarrow_\Box$ is applicable to W_i then apply $\rightarrow_\Box$ to W_i
   else if $\rightarrow_\Diamond$ is applicable to W_i then apply $\rightarrow_\Diamond$ to W_i
   else if $\rightarrow_\circ$ is applicable to W_i then apply $\rightarrow_\circ$ to W_i.

   (b) $i := i + 1$.

3. return W_i.

Figure 2: The frame algorithm.

algorithm is the (extended) world constraint system W_n. From this we can construct a *frame*, i.e., a pair $(\mathcal{W}, \Gamma)$ consisting of a set of worlds $\mathcal{W}$ and a set Γ of accessibility relations γ_i (one for each agent i). Thereby, $\mathcal{W}$ consists of all labels in W, and γ_i is given by the reflexive and transitive closure of the set $\{(w, v) \mid w \bowtie_i v \text{ is in } W\}$.

Example 3.3 Suppose a to be an agent who knows that each gasoline truck can transport gasoline, and that John owns truck-1 which is a gasoline truck. Furthermore, suppose that neither agent a nor agent b knows that truck-1 can transport gasoline. This can be expressed by the following five $\mathcal{ALC}_\mathcal{K}$-axioms.

$$\Box_a(\textit{gasoline-truck} \sqsubseteq \textit{can-transport-gasoline})$$
$$\Box_a(\textit{John owns truck-1})$$
$$\Box_a(\textit{truck-1} : \textit{gasoline-truck})$$
$$\Diamond_a(\textit{truck-1} : \neg\textit{can-transport-gasoline})$$
$$\Diamond_b(\textit{truck-1} : \neg\textit{can-transport-gasoline})$$

The world constraint system W_0 which is induced by these five $\mathcal{ALC}_\mathcal{K}$-axioms is the set which consists of the following world constraints

$$(1)\quad \Box_a(\textit{gasoline-truck} \sqsubseteq \textit{can-transport-gasoline}) \parallel w_0$$
$$(2)\quad \Box_a(\textit{John owns truck-1}) \parallel w_0$$
$$(3)\quad \Box_a(\textit{truck-1} : \textit{gasoline-truck}) \parallel w_0$$
$$(4)\quad \Diamond_a(\textit{truck-1} : \neg\textit{can-transport-gasoline}) \parallel w_0$$
$$(5)\quad \Diamond_b(\textit{truck-1} : \neg\textit{can-transport-gasoline}) \parallel w_0$$

By applications of the $\rightarrow_\Box$ rule to (1), (2), and (3), respectively we obtain the world constraint system W_1 which extends W_0 by

$$(6)\quad \textit{gasoline-truck} \sqsubseteq \textit{can-transport-gasoline} \parallel w_0$$
$$(7)\quad \textit{John owns truck-1} \parallel w_0$$
$$(8)\quad \textit{truck-1} : \textit{gasoline-truck} \parallel w_0$$

Then, by one application of the $\rightarrow_{\diamond}$ rule to (4), we obtain the world constraint system W_2 which extends W_1 by

$$
\begin{array}{rl}
w_0 \bowtie_a w_1 & \\
\mathit{truck\text{-}1} : \neg \mathit{can\text{-}transport\text{-}gasoline} \mid\mid w_1 & \text{from (4)} \\
\Box_a(\mathit{gasoline\text{-}truck} \sqsubseteq \mathit{can\text{-}transport\text{-}gasoline}) \mid\mid w_1 & \text{from (1)} \\
\mathit{gasoline\text{-}truck} \sqsubseteq \mathit{can\text{-}transport\text{-}gasoline} \mid\mid w_1 & \text{from (1)} \\
\Box_a(\mathit{John\ owns\ truck\text{-}1}) \mid\mid w_1 & \text{from (2)} \\
\mathit{John\ owns\ truck\text{-}1} \mid\mid w_1 & \text{from (2)} \\
\Box_a(\mathit{truck\text{-}1} : \mathit{gasoline\text{-}truck}) \mid\mid w_1 & \text{from (3)} \\
\mathit{truck\text{-}1} : \mathit{gasoline\text{-}truck} \mid\mid w_1 & \text{from (3)}
\end{array}
$$

Now, the labelled $\mathcal{ALC}_{\mathcal{K}}$-axiom (5),

$$\Diamond_b(\mathit{truck\text{-}1} : \neg \mathit{can\text{-}transport\text{-}gasoline}) \mid\mid w_0,$$

is covered by w_1, such that we obtain $W_3 = W_2 \cup \{w_0 \bowtie_b w_1\}$ by the $\rightarrow_{\circ}$ rule. Now, no more rules are applicable and the frame algorithm results the extended world constraint system W_3. From W_3 we can construct the frame $(\mathcal{W}, \Gamma)$ with $\mathcal{W} = \{w_0, w_1\}$, and Γ consists of $\gamma_a = \gamma_b = \{(w_0, w_0), (w_0, w_1), (w_1, w_1)\}$. Each world w_i consists of the $\mathcal{ALC}_{\mathcal{K}}$-axioms which are labelled with w_i. ■

We will now show that the frame algorithm has the following two important properties. Firstly, it terminates with a world constraint system W as input which is induced by a finite set $F_1, \ldots, F_n$ of $\mathcal{ALC}_{\mathcal{K}}$-axioms. Secondly, the result of the frame algorithm with input W is $S4$-satisfiable iff the set $F_1, \ldots, F_n$ of $\mathcal{ALC}_{\mathcal{K}}$-axioms is $S4$-satisfiable.

Theorem 3.4 *If W is a world constraint system which is induced by a finite set of $\mathcal{ALC}_{\mathcal{K}}$-axioms, the frame algorithm terminates with input W.*

Proof: Let W be induced by the $\mathcal{ALC}_{\mathcal{K}}$-axioms $F_1, \ldots, F_n$. The main idea is that only a finite number of new labels are introduced to W by the frame algorithm. Therefore, let us firstly have a look at the labelled $\mathcal{ALC}_{\mathcal{K}}$-axioms which are added to W by applications of the propagation rules.

1. An application of the $\rightarrow_{\Box}$ rule to $\Box_i F \mid\mid w$ adds the labelled $\mathcal{ALC}_{\mathcal{K}}$-axiom $F \mid\mid w$ to W. Thereby, the left hand side F of $F \mid\mid w$ is a subformula of the left hand side $\Box_i F$ of $\Box_i F \mid\mid w$, to which the $\rightarrow_{\Box}$ rule has been applied.

2. An application of the $\rightarrow_{\Diamond}$ rule to $\Diamond_i F \mid\mid w$ adds the labelled $\mathcal{ALC}_{\mathcal{K}}$-axioms

$$F \mid\mid v, G_1 \mid\mid v, \Box_i G_1 \mid\mid v, \ldots, G_m \mid\mid v, \Box_i G_m \mid\mid v$$

to W, if $\Box_i G_1 \,||\, w, \ldots, \Box_i G_m \,||\, w$ are exactly the world constraints of the form $\Box_i G || w$ in W and v is a new label. Again, each left hand side of the added labelled $\mathcal{ALC}_\mathcal{K}$-axioms is a (sub)formula of the left hand side of one of the labelled $\mathcal{ALC}_\mathcal{K}$-axioms $\Diamond_i F \,||\, w, \Box_i G_1 \,||\, w, \ldots, \Box_i G_m \,||\, w$ to which the $\rightarrow_\Diamond$ rule has been applied.

3. The $\rightarrow_\circ$ rule does not add new labelled $\mathcal{ALC}_\mathcal{K}$-axioms to W at all.

Thus, since the frame algorithm starts with a world constraint system W which is given by $\{F_1 \,||\, w_0, \ldots, F_n \,||\, w_0\}$, it only adds labelled $\mathcal{ALC}_\mathcal{K}$-axioms $F \,||\, w$ to W where F is a (sub)formula of one of the $\mathcal{ALC}_\mathcal{K}$-axioms $F_1, \ldots, F_n$. Let now $\mathcal{S}$ be the set of all possible sets consisting of (sub)formulas of $F_1, \ldots, F_n$. Obviously, $\mathcal{S}$ is finite since $\{F_1, \ldots, F_n\}$ is finite.

This consideration in mind, it is easy to verify that the $\rightarrow_\Diamond$ rule can be applied to W only a finite number of times: If the $\rightarrow_\Diamond$ rule is applied to a labelled $\mathcal{ALC}_\mathcal{K}$-axiom $\Diamond_i F \,||\, w$, and $\Box_i G_1 \,||\, w, \ldots, \Box_i G_m \,||\, w$ are exactly the world constraints of the form $\Box_i G \,||\, w$ in W, then $\{\Diamond_i F, \Box_i G_1, \ldots, \Box_i G_m\}$ is an element of $\mathcal{S}$. The application of $\rightarrow_\Diamond$ then extends W by $\{F \,||\, v, G_1 \,||\, v, \Box_i G_1 \,||\, v, \ldots, G_m \,||\, v, \Box_i G_m \,||\, v\}$ where v is a new label. Thus, given an arbitrary label $\tilde{w}$, the $\rightarrow_\Diamond$ rule can no more be applied to $\Diamond_i F \,||\, \tilde{w}$ if $\Box_i G_1 \,||\, \tilde{w}, \ldots, \Box_i G_m \,||\, \tilde{w}$ are exactly the world constraints in W of the form $\Box_i G \,||\, \tilde{w}$ since $\tilde{w}$ is now covered by v. Summing up, to each set in $\mathcal{S}$ the $\rightarrow_\Diamond$ rule can be applied at most once. Thus, this rule can be applied only a finite number of times, i.e., the frame algorithm can only add a finite number of new labels to W.

Obviously, given a fixed label w in W, the $\rightarrow_\Box$ rule can only be applied a finite number of times. This is the case because the labelled $\mathcal{ALC}_\mathcal{K}$-axiom which is added to W by this rule is syntactically shorter than the labelled $\mathcal{ALC}_\mathcal{K}$-axiom to which the rule has been applied. Finally, the $\rightarrow_\circ$ rule can be applied to each pair (w, v) of labels in W at most once, and therefore only a finite number of times. $\Box$

Thus, the application of the frame algorithm to a world constraint system W induced by the $\mathcal{ALC}_\mathcal{K}$-axioms $F_1, \ldots, F_n$ terminates and results a world constraint system, say W'. To test $S4$-satisfiability of W', for each label w in W' we compute the set of all those $\mathcal{ALC}_\mathcal{K}$-axioms in W' which are labelled by w and which do not contain any indexed $\Box$ or $\Diamond$ operator. That means, such a set contains only $\mathcal{ALC}$-axioms and is therefore called the $\mathcal{ALC}$ *test set of label* w. More formally, if W is a world constraint system, the $\mathcal{ALC}$ test set $A(w)$ of label w in W is given by the set

$$\{F \mid F \,||\, w \in W, \text{ and } F \text{ does not contain any modality}\}.$$

We are now going to show that a set $F_1, \ldots, F_n$ of $\mathcal{ALC}_\mathcal{K}$-axioms is $S4$-satisfiable iff the $\mathcal{ALC}$ test set $A(w)$ of each label w in W' is satisfiable. Thereby, W' is the result of the frame algorithm with input $\{F_1 \,||\, w_0, \ldots, F_n \,||\, w_0\}$. We will firstly prove the following two lemmata.

Lemma 3.5 *Let W be a world constraint system which is induced by $\mathcal{ALC}_\mathcal{K}$-axioms $F_1, \ldots, F_n$, and let W' be the result of the frame algorithm with input W. If $K = (\mathcal{W}, \Gamma, K_I)$ is an S4 Kripke structure which satisfies W, then for each label w in W' there is a world $w^K \in \mathcal{W}$ such that $K, w^K \models F$ for each labelled $\mathcal{ALC}_\mathcal{K}$-axiom $F \mid\mid w$ in W'.*

Proof: If W' is the result of the frame algorithm with input W there is a chain $W = W_0 \rightarrow_1 W_1 \rightarrow_1 \ldots \rightarrow_k W_k = W'$ with $\rightarrow_i \in \{\rightarrow_\Box, \rightarrow_\Diamond, \rightarrow_\circ\}$ for $i \in \{1, \ldots, k\}$. We will now show that K satisfies each labelled axiom in W' by induction over the number of rule applications. By assumption, $K = (\mathcal{W}, \Gamma, K_I)$ satisfies $W_0 = \{F_1 || w_0, \ldots, F_n || w_0\}$, i.e., there is a world w_0^K in $\mathcal{W}$ such that $K, w_o^K \models F_1, \ldots, K, w_o^K \models F_n$.

We thus can assume that, after j rule applications, for each label w in W_j there is a world w^K in $\mathcal{W}$ such that $K, w^K \models F$ for each labelled $\mathcal{ALC}_\mathcal{K}$-axiom $F \mid\mid w$ in W_j. If $W_j \rightarrow_j W_{j+1}$ there are three possibilities. Firstly, suppose $W_j \rightarrow_\Box W_{j+1}$. Then there is a labelled $\mathcal{ALC}_\mathcal{K}$-axiom $\Box_i F \mid\mid w$ in W_j, and $W_{j+1} = W_j \cup \{F \mid\mid w\}$. By induction hypothesis, $K, w^K \models \Box_i F$ for some world w^K in $\mathcal{W}$. Because of property (P1) of $S4$ Kripke structures, therefore $K, w^K \models F \mid\mid w$ holds. That means, K satisfies each labelled $\mathcal{ALC}_\mathcal{K}$-axiom in W_{j+1}.

Secondly, suppose $W_j \rightarrow_\Diamond W_{j+1}$. In this case, there are labelled $\mathcal{ALC}_\mathcal{K}$-axioms $\Diamond_i F \mid\mid w, \Box_i G_1 \mid\mid w, \ldots, \Box_i G_m \mid\mid w$ in W_j, and

$$W_{j+1} = W_j \cup \{w \bowtie_i v, F \mid\mid v, \Box_i G_1 \mid\mid v, G_1 \mid\mid v, \ldots, \Box_i G_m \mid\mid v, G_m \mid\mid v\}$$

where v is a new label. By induction hypothesis, there is a world w^K in $\mathcal{W}$ such that (*i*) $K, w^K \models \Diamond_i F$ and (*ii*) $K, w^K \models \Box_i G_j$ for $j = 1, \ldots, m$. Because of (*i*) there is a world v^K in $\mathcal{W}$ (not neccessarily different from w^K) such that $(w^K, v^K) \in \gamma_i$ and $K, v^K \models F$. Furthermore, because of (*ii*) and property (P2) of $S4$ Kripke structures, both $K, w^K \models \Box_i G_j$ and $K, w^K \models \Box_i \Box_i G_j$ holds for $j = 1, \ldots, m$. And thus, since $(w^K, v^K) \in \gamma_i$, especially $K, v^K \models G_j$ and $K, v^K \models \Box_i G_j$ holds for $j = 1, \ldots, m$. That means, K satisfies each labelled $\mathcal{ALC}_\mathcal{K}$-axiom in W_{j+1}.

Finally, if $W_j \rightarrow_\circ W_{j+1}$, there is nothing to show since the $\rightarrow_\circ$ rule does not introduce new labelled $\mathcal{ALC}_\mathcal{K}$-axioms to W_j. $\Box$

The next lemma states that a world constraint W', which is the result of the frame algorithm, is $S4$-satisfiable if the $\mathcal{ALC}$ test set of each label in W' is satisfiable.

Lemma 3.6 *Let W be a world constraint system which is induced by a finite set of $\mathcal{ALC}_\mathcal{K}$-axioms, and let W' be the result of the frame algorithm with input W. If the $\mathcal{ALC}$ test set $A(w)$ of each label w in W' is satisfiable, then W is S4-satisfiable.*

Proof: Let K be the Kripke structure $(\mathcal{W}, \Gamma, K_I)$ with

- $\mathcal{W}$ is given by the set of all labels in W'.
- Γ consists of one accessibility relation γ_i for each agent i. Thereby, γ_i is given by the reflexive and transitive closure of the set $\{(w, v) \mid w \bowtie_i v \text{ in } W\}$.
- K_I is given such that $K, w \models F$ for each labelled $\mathcal{ALC}_{\mathcal{K}}$-axiom $F \,||\, w$ in W' where F does not contain any indexed $\Box$ or $\Diamond$ operator.[2]

Obviously, K is an $S4$ Kripke structure, since each accessibility relation is reflexive and transitive. We will now show that K satisfies each world constraint c in W'. If c is of the form $w \bowtie_i v$ there is nothing to show because of the definition of K.

The fact $K \models F \,||\, w$ for each labelled $\mathcal{ALC}_{\mathcal{K}}$-axiom $F \,||\, w$ in W' can be shown by induction over the number of modalities in F. If F does not contain any modalities, then $K, w \models F$ because of the construction of K. Thus we can assume that $K, w \models F$ for each labelled $\mathcal{ALC}_{\mathcal{K}}$-axiom $F \,||\, w$ in W' such that F contains n modalities.

If F contains $n+1$ modalities, there are two possibilities: the leading operator is either $\Box_i$ or $\Diamond_i$. Firstly, suppose W' contains a world constraint $\Box_i F \,||\, w$, where F has n modalities. We then have to show that $K, w \models \Box_i F$, i.e., that $K, v \models F$ for each v such that $(w, v) \in \gamma_i$. If $w = v$, then $K, v \models F$ because $F \,||\, w$ is in W' if $\Box_i F \,||\, w$ is in W' (by an application of the $\rightarrow_\Box$ rule), and F contains only n modalities. If $w \neq v$ and $(w, v) \in \gamma_i$, then, because of the definition of γ_i, there is a path

$$w = w_{i_1} \bowtie_i w_{i_2}, w_{i_2} \bowtie_i w_{i_3}, \ldots, w_{i_{k-1}} \bowtie_i w_{i_k} = v$$

in W'. That means, during the frame algorithm world constraint systems $W_{i_1}, \ldots, W_{i_k}$ have been constructed such that the world constraint $w_{i_j} \bowtie_i w_{i_{j+1}}$ is introduced to W' by $W_{i_j} \rightarrow_{i_j} W_{i_{j+1}}$ with $\rightarrow_{i_j} \in \{\rightarrow_\Diamond, \rightarrow_\circ\}$ for $1 \leq j \leq k-1$.

It is easy to verify that after applying $\rightarrow_{i_1}$ to W_{i_1} no further labelled $\mathcal{ALC}_{\mathcal{K}}$-axioms with label $w_{i_1} (= w)$ are added to the world constraint system: The $\rightarrow_\Box$ rule has already been applied as often as possible, the $\rightarrow_\Diamond$ rule only introduces labelled $\mathcal{ALC}_{\mathcal{K}}$-axioms with a *new* label, and the $\rightarrow_\circ$ rule does not introduce new labelled $\mathcal{ALC}_{\mathcal{K}}$-axioms at all. By assumption, we know $\Box_i F \,||\, w$ is in W', and therefore $\Box_i F \,||\, w$ is in W_{i_1}. Thus, by definition of the $\rightarrow_\Diamond$ and the $\rightarrow_\circ$ rule, both $\Box_i F \,||\, w_{i_2}$ and $F \,||\, w_{i_2}$ are in W_{i_2}. Analogously, W_{i_k} contains $\Box_i F \,||\, w_{i_k}$ and $F \,||\, w_{i_k}$, such that $F \,||\, v \in W'$ since $w_{i_k} = v$ and $W_{i_k} \subseteq W'$. By induction hypothesis we know $K, v \models F$ because F contains only n modalities.

Suppose now W' contains $\Diamond_i F \,||\, w$. We then have to show that $K \models F \,||\, v$ for some world v such that $(w, v) \in \gamma_i$. If $\Diamond_i F \,||\, w$ is in W', then the world constraints $F \,||\, v$ and

[2]Note that such a K-interpretation K_I exists, since we assumed the $\mathcal{ALC}$ test set of each label in W' to be satisfiable. Given interpetations $I_1, \ldots, I_n$ which satisfy the $\mathcal{ALC}$ test sets of each label in W' respectively, the construction of K_I is straightforward.

1. Let W be the world constraint system which is induced by $F_1, \ldots, F_n$.
2. Let W' be the result of the frame algorithm with input W.
3. For each label w in W' do: If the $\mathcal{ALC}$ test set of w is not satisfiable, then STOP and return "$S4$-unsatisfiable".
4. Return "$S4$-satisfiable".

Figure 3: The $S4$-satisfiability algorithm.

$w \bowtie_i v$ are in W' because either the $\rightarrow_\diamond$ or the $\rightarrow_\circ$ rule has been applied to $F \,||\, w$. By construction of γ_i we then know $(w, v) \in \gamma_i$, and thus $K \models F \,||\, v$ follows by induction hypothesis. □

The following theorem summarizes the previous results.

Theorem 3.7 *Let $F_1, \ldots, F_n$ be a finite set of $\mathcal{ALC}_\mathcal{K}$-axioms, and let W be the world constraint system which is induced by $F_1, \ldots, F_n$. If W' is the result of the frame algorithm with input W, then the set $F_1, \ldots, F_n$ is $S4$-satisfiable iff the $\mathcal{ALC}$ test set $A(w)$ of each label w in W' is satisfiable.*

Proof: By definition, the set $F_1, \ldots, F_n$ of $\mathcal{ALC}_\mathcal{K}$-axioms is $S4$-satisfiable iff W is $S4$-satisfiable. Firstly, suppose K is an $S4$ Kripke structure which satisfies W. Then, because of Lemma 3.5, for each label w in W' there is a world $w^K \in \mathcal{W}$ such that $K, w \models F$ for each $\mathcal{ALC}_\mathcal{K}$-axiom $F \,||\, w$ in W'. Thus, especially the $\mathcal{ALC}$ test set of each label w in W' is satisfied by K. Conversely, suppose that the $\mathcal{ALC}$ test set $A(w)$ of each label w in W' is satisfiable. Then W is $S4$-satisfiable because of Lemma 3.6. □

Thus, we obtain the algorithm for testing $S4$-satisfiability of a set $F_1, \ldots, F_n$ of $\mathcal{ALC}_\mathcal{K}$-axioms which is given in Figure 3. An algorithm for testing satisfiability of $\mathcal{ALC}$ test sets will be given in the next section.

4 Testing Satisfiability of $\mathcal{ALC}$ Test Sets

In this section we will show how to test satisfiability of a given $\mathcal{ALC}$ test set $A(w)$. We procedd as follows. Firstly, we will show that satisfiability of $A(w)$ is equivalent to the problem whether there exists an interpretation I such that I satisfies a given $\mathcal{ALC}$-ABox $\mathcal{A}$ and such that $D_0^I = \Delta^I$ for a given concept D_0. This test will be called top consistency test. In Subsection 4.1 we will prove this equivalence, show how to construct $\mathcal{A}$ and D_0 from $A(w)$, and give an algorithm for deciding top consistency. In Subsection 4.2 we will show that this algorithm terminates and that it is sound and complete.

4.1 Testing Top Consistency

An $\mathcal{ALC}$ test set $A(w)$ consists of a finite number of $\mathcal{ALC}_{\mathcal{K}}$-axioms without any indexed modalities, i.e., of terminological and assertional axioms of the form

$C = D, C \neq D$	(negated) concept equivalence
$a : C$	concept instance
aRb	role instance

only, where C, D are concepts, R is a role, and a, b are objects.

As a result of the previous section, $S4$-satisfiability of a set of $\mathcal{ALC}_{\mathcal{K}}$-axioms can be reduced to satisfiability tests of a number of $\mathcal{ALC}$ test sets (cf. Theorem 3.7). Observe that the concept instances and the role instances in an $\mathcal{ALC}$ test set $A(w)$ define an $\mathcal{ALC}$-ABox. That means, testing satisfiability of an $\mathcal{ALC}$ test set is equivalent to testing satisfiability of an $\mathcal{ALC}$-ABox together with a set of (negated) concept equivalences. The next theorem states that this test can be performed by an algorithm which checks top consistency of a single concept w.r.t. an $\mathcal{ALC}$-ABox. We say a concept C is *top consistent w.r.t. an $\mathcal{ALC}$-ABox $\mathcal{A}$* iff there exists an interpretation I such that $I \models \mathcal{A}$ and C is interpreted as the top concept, i.e., $C^I = \top^I$.

Theorem 4.1 *Let $\mathcal{A}$ be an $\mathcal{ALC}$-ABox and let C_i, D_i, E_j, F_j be concepts. There exists an interpretation which satisfies $\mathcal{A}$, $C_1 = D_1, \ldots, C_n = D_n$, and $E_1 \neq F_1, \ldots, E_m \neq F_m$ iff the concept $\big((C_1 \sqcap D_1) \sqcup (\neg C_1 \sqcap \neg D_1)\big) \sqcap \ldots \sqcap \big((C_n \sqcap D_n) \sqcup (\neg C_n \sqcap \neg D_n)\big)$ is top consistent w.r.t. $\mathcal{A} \cup \{a_1 : (E_1 \sqcap \neg F_1) \sqcup (\neg E_1 \sqcap F_1), \ldots, a_m : (E_m \sqcap \neg F_m) \sqcup (\neg E_m \sqcap F_m)\}$.*

Proof: Firstly, let I be an interpretation which satisfies $\mathcal{A}$, $C_i = D_i$, and $E_j \neq F_j$ ($i = 1, \ldots, n, j = 1, \ldots, m$). Since I satisfies $E_j \neq F_j$ there exists an element $d_j \in \Delta^I$ such that $d_j \in [E_j \sqcap \neg F_j]^I$ or $d_j \in [\neg E_j \sqcap F_j]^I$. Let now I' be the interpretation which extends I as follows: for each of the elements d_j ($j = 1, \ldots, m$) a new element d'_j is added to the universe Δ^I of I. Then, each of these new elements d'_j is interpreted by I' exactly as d_j is interpreted by I.[3] More formally, $\Delta^{I'} := \Delta^I \cup \{d'_1, \ldots, d'_m\}$ where d'_j is a new element. Furthermore, each atomic concept A is interpreted by $A^{I'} := A^I \cup \{d'_j \mid d_j \in A^I\}$, each role R is interpreted by $R^{I'} := R^I \cup \{(d'_j, d) \mid (d_j, d) \in$

[3]Note that this can be done only in concept languages which are not expressive enough to state that a given concept contains at most n elements ($n > 0$). It is easy to verify that $\mathcal{ALC}$ satisfies this condition.

$R^I\} \cup \{(d, d'_j) \mid (d, d_j) \in R^I\}$, each object o in $\mathcal{A}$ is interpreted by $o^{I'} := o^I$, and, finally, the new objects $a_1, \ldots, a_m$ are interpreted by $a_j^{I'} = d'_j$. Note that I' is defined in such a way that we can guarantee unique name interpretation of each object the ABox in $\mathcal{A} \cup \{a_1 : (E_1 \sqcap \neg F_1) \sqcup (\neg E_1 \sqcap F_1), \ldots, a_m : (E_m \sqcap \neg F_m) \sqcup (\neg E_m \sqcap F_m)\}$.

It is easy to verify that I' satisfies $\mathcal{A}$. Furthermore, I' satisfies $a_j : (E_j \sqcap \neg F_j) \sqcup (\neg E_j \sqcap F_j)$ for $j = 1, \ldots, m$ since d'_j is in $[E_j \sqcap \neg F_j]^{I'}$ or in $[\neg E_j \sqcap F_j]^{I'}$ iff d_j is in $[E_j \sqcap \neg F_j]^I$ or in $[\neg E_j \sqcap F_j]^I$. Analogously, I' satisfies the concept equivalences $C_i = D_i$ since they are satisfied by I. That means, for each element $d \in \Delta^{I'}$ either d is in $C^{I'}$ and in $D^{I'}$, or d is in $[\neg C]^{I'}$ and in $[\neg D]^{I'}$. Thus, I' satisfies $[(C_i \sqcap D_i) \sqcup (\neg C_i \sqcap \neg D_i)]^I = \Delta^I$ for $i = 1, \ldots, n$. And thus I' interprets $\big((C_1 \sqcap D_1) \sqcup (\neg C_1 \sqcap \neg D_1)\big) \sqcap \ldots \sqcap \big((C_n \sqcap D_n) \sqcup (\neg C_n \sqcap \neg D_n)\big)$ as $\Delta^{I'}$.

Conversely, suppose I satisfies $\mathcal{A} \cup \{a_1 : (E_1 \sqcap \neg F_1) \sqcup (\neg E_1 \sqcap F_1), \ldots, a_m : (E_m \sqcap \neg F_m) \sqcup (\neg E_m \sqcap F_m)\}$ and I interprets $\big((C_1 \sqcap D_1) \sqcup (\neg C_1 \sqcap \neg D_1)\big) \sqcap \ldots \sqcap \big((C_n \sqcap D_n) \sqcup (\neg C_n \sqcap \neg D_n)\big)$ as Δ^I. Since I satisfies $a_j : (E_j \sqcap \neg F_j) \sqcup (\neg E_j \sqcap F_j)$, obviously $E_j \not\equiv F_j$ is satisfied by I for $j = 1, \ldots, m$. Furthermore, since $[(C_i \sqcap D_i) \sqcup (\neg C_i \sqcap \neg D_i)]^I = \Delta^I$, each element $d \in \Delta^I$ is in C^I iff d is in D^I. That means, I satisfies $C_i = D_i$. □

Thus, for testing satisfiability of an $\mathcal{ALC}$ test set $A(w)$ we need an algorithm which tests top consistency of a concept D_0 w.r.t. an $\mathcal{ALC}$-ABox $\mathcal{A}$. We will now give such an algorithm which is based on the notion of a (concept) constraint system. A *constraint system* is a finite non-empty set of constraints $a : C$ or aRb, where C is a concept, R is role, and a, b are objects. A constraint system S contains a *clash* iff (i) S contains two concept instances of the form $a : A$ and $a : \neg A$ where a is an object and A is an atomic concept or (ii) S contains a constraint $a : \bot$ for some object a. We say S is *clash-free* iff S does not contain a clash. A constraint system S is *satisfiable* iff there exists an interpretation I such that $I \models s$ for each constraint s in S.

Given an $\mathcal{ALC}$-ABox $\mathcal{A}$ and a concept D_0, we say the constraint system S is *induced* by $\mathcal{A}$ and D_0 iff $S = \mathcal{A} \cup \{a_0 : D_0^*, a_1 : D_0^*, \ldots, a_n : D_0^*\}$ where a_o is a new object, D_0^* is the negation normal form of D_0, and $a_1, \ldots, a_n$ are exactly the objects in $\mathcal{A}$.

The *top consistency algorithm* has an $\mathcal{ALC}$-ABox $\mathcal{A}$ and a concept D_0 as input. The algorithm starts with a constraint system S which is induced by an $\mathcal{ALC}$-ABox $\mathcal{A}$ and a concept D_0, and successively adds new constraints to S by the five propagation rules defined in Figure 4. Thereby, it works as follows. Let S_0 be the constraint system which is induced by $\mathcal{A}$ and D_0. If there exists a chain $S_0 \hookrightarrow_1 S_1 \hookrightarrow_2 \ldots \hookrightarrow_n S_n$ such that (i) each $\hookrightarrow_i$ is the first rule is the sequence $\rightarrow_\sqcap, \rightarrow_\sqcup, \rightarrow_\forall, \rightarrow_{\exists_1}, \rightarrow_{\exists_2}$ which is applicable to S_i and (ii) S_n is complete and clash-free, then return "top consistent" else return "not top consistent". A constraint system S is called *complete* iff no propagation rule is applicable to S.

1. $S \rightarrow_{\sqcap} \{a : C_1, a : C_2\} \cup S$

   if $a : C_1 \sqcap C_2$ is in S
   and S does not contain both constraints $a : C_1$ and $a : C_2$.

2. $S \rightarrow_{\sqcup} \{a : D\} \cup S$

   if $a : C_1 \sqcup C_2$ is in S,
   neither $a : C_1$ nor $a : C_2$ is in S, and D is either C_1 or C_2.

3. $S \rightarrow_{\forall} \{b : C\} \cup S$

   if $a : \forall R.C$ and aRb are in S
   and $b : C$ is not in S.

4. $S \rightarrow_{\exists_1} \{aRb, b : C, b : D_0^*\} \cup S$

   if $a : \exists R.C$ is in S,
   $D_1, \ldots, D_n$ are exactly the concepts occuring in constraints of the form $a : \forall R.D_i$ in S, there exists no c such that $c : C, c : D_1, \ldots, c : D_n, c : D_0^*$ are all in S, and b is a new object.

5. $S \rightarrow_{\exists_2} \{aRc\} \cup S$

   if $a : \exists R.C$ is in S,
   $D_1, \ldots, D_n$ are exactly the concepts occuring in constraints of the form $a : \forall R.D_i$ in S, and for some c the constraints $c : C, c : D_1, \ldots, c : D_n, c : D_0^*$ are all in S and aRc is not in S.

Figure 4: Propagation rules of the top consistency test.

4.2 Properties of the Top Consistency Algorithm

In this subsection we will show that the top consistency algorithm is sound, complete, and terminates. That means, if we apply this algorithm to an $\mathcal{ALC}$-ABox $\mathcal{A}$ and a concept D_0, the algorithm terminates and returns "top consistent" iff D_0 is top consistent w.r.t. $\mathcal{A}$.

Thus, if S is the constraint system which is induced by $\mathcal{A}$ and D_0 we firstly show: Each chain of rule applications starting with S which can be constructed by the top consistency algorithm is finite. Note, that the top consistency algorithm may construct more than one such chain if S contains concept disjunctions, e.g., $a : C \sqcup D$.

Lemma 4.2 *Let S be a constraint system which is induced by an $\mathcal{ALC}$-ABox $\mathcal{A}$ and a concept D_0. Then the top consistency algorithm cannot construct an infinite chain* $S = S_0 \rightarrow_1 S_1 \rightarrow_2 \ldots$ *with* $\rightarrow_i \in \{\rightarrow_{\sqcap}, \rightarrow_{\sqcup}, \rightarrow_{\forall}, \rightarrow_{\exists_1}, \rightarrow_{\exists_2}\}$.

Proof: Firstly, we show that the $\rightarrow_{\exists_1}$ rule can be applied to S only a finite number of times. It is easy to verify that, if $b : C$ is added to S_i by a propagation rule, the concept C is a (sub)concept of the concepts occurring in S_0. Let $\mathcal{P}$ be the set of all possible sets of concepts which can be built up from (sub)concepts in S_0. Obviously, $\mathcal{P}$ is finite.

Suppose now, in some S_i the $\rightarrow_{\exists_1}$ rule is applied to a constraint $a : \exists R.C$, where $D_1, \ldots, D_n$ are exactly the concepts in the constraints of the form $a : \forall R.D_l$ in S_i. Then S_i is extended by $aRb, b : C, b : D_0^*$ where b is a new object. Furthermore, before we can apply the $\rightarrow_{\exists_1}$ rule again, the $\rightarrow_{\forall}$ rule has to be applied to aRb and $a : \forall R.D_1, \ldots, a : \forall R.D_n$, respectively. Thereby, the constraints $b : D_1, \ldots, b : D_n$ are added. That means, we will obtain a constraint system, say S_j $(j > i)$, which contains at least the constraints $b : C, b : D_0^*, b : D_1, \ldots, b : D_n$, where $\{C, D_0^*, D_1, \ldots, D_n\}$ is an element in $\mathcal{P}$. Suppose now, there is a constraint of the form $a' : \exists R'.C$ in some constraint system S_k, $k \geq j$, where a' is an arbitrary object and R' is an arbitrary role. Furthermore, let $D_1, \ldots, D_n$ be the concepts in the constraints of the form $a' : \forall R'.D_l$ in S_k. In this case, the $\rightarrow_{\exists_1}$ rule is *not* applicable to S_k (because of its precondition). The reason for this is due to the fact that there is already an object, namely b, in S_k such that the constraints $b : C, b : D_1, \ldots, b : D_n, b : D_0^*$ are all in S_k. In this case only the $\rightarrow_{\exists_2}$ can eventually be applied to $a' : \exists R'.C$, adding $a'Rb$ to S_k. Thus, since $\mathcal{P}$ is finite, the $\rightarrow_{\exists_1}$ is applicable only a finite number of times.

As an immediate consequence, only a finite number of new objects are added to S because none of the other propagation rules introduces new objects to a constraint system. Thus, each S_i contains only a finite number of objects since the number of objects in S_0 is finite. With this, it is easy to verify that the remaining rules $\rightarrow_{\sqcap}$, $\rightarrow_{\sqcup}$, $\rightarrow_{\forall}$, and $\rightarrow_{\exists_2}$ can be applied only a finite number of times: These rules are applied to $a : C_1 \sqcap C_2$, $a : C_1 \sqcup C_2$, $a : \forall R.C$, and $a : \exists R.C$, respectively, and add strictly shorter constraints to S than the constraint they have been applied to. Thereby, a is an object in S and C_1, C_2, $\forall R.C$, and $\exists R.C$ are (sub)formulas of concepts in S_0. Furthermore, the $\rightarrow_{\sqcap}$ ($\rightarrow_{\sqcup}$) rule can be applied to each constraint of the form $a : C_1 \sqcap C_2$ ($a : C_1 \sqcup C_2$) only once. The $\rightarrow_{\forall}$ rule can be applied to the pair $a : \forall R.C$ and aRb only once. And, finally, the $\rightarrow_{\exists_2}$ rule can be applied to each pair (a, b) of objects in S at most once. □

To prove soundness and completeness of the top consistency algorithm, we introduce the following important lemma.

Lemma 4.3 *Each complete and clash-free constraint system is satisfiable.*

Proof: Let S be a complete and clash-free constraint system, and let I be an interpretation such that

- Δ^I is the set of objects in S.

- $A^I := \{a \mid a : A \text{ is in } S\}$ for each atomic concept A in S
- $R^I := \{(a, b) \mid aRb \text{ is in } S\}$ for each role R in S
- $a^I := a \in \Delta^I$ for each object a in S

We will now show that $I \models s$ for each constraint s in S: If s is of the form aRb then $I \models s$ by definition of I.

If s is of the form $a : C$, then $I \models s$ can be shown by induction over the structure of C: If C is an atomic concept, then $I \models a : C$ because of the definition of I. If $C = \top$ then $I \models a : C$ because of $\top^I = \Delta^I$, and C cannot be $\bot$ since S is clash-free. For the induction step we have to show that $I \models a : C$ if $a : C$ is in S, and C is of the form $\neg C_1$, $C_1 \sqcap C_2$, $C_1 \sqcup C_2$, $\forall R.C_1$ or $\exists R.C_1$.

Firstly, let C be of the form $\neg C_1$. Since we assumed the input of the top consistency algorithm to be in negation normal form, and none of the five propagation rules introduces concepts which are not in negation normal form, C_1 is an atomic concept. Furthermore, since S is clash-free, $a : C_1$ is not in S. Therefore $I \not\models a : C_1$ and thus $I \models a : \neg C_1$.

If C is of the form $C_1 \sqcap C_2$ ($C_1 \sqcup C_2$) we know $a : C_1$ and (or) $a : C_2$ to be in S because S is complete. In this case, by induction hypothesis, $I \models a : C_1$ and (or) $I \models a : C_2$. Thus, the induction step is trivial.

Let now C be of the form $\exists R.C_1$. Since S is complete neither the $\rightarrow_{\exists_1}$ nor the $\rightarrow_{\exists_2}$ rule is applicable to S. Therefore, one of these rules has already been applied to $a : \exists R.C_1$, i.e., aRb and $b : C_1$ are in S for some object b. By construction of I we know that $I \models aRb$ and, by induction hypothseis, $I \models b : C_1$. Thus, $I \models a : \exists R.C_1$.

Finally, let C be of the form $\forall R.C_1$. If there does not exist an object b in S such that aRb is in S, then $(a^I, u) \notin R^I$ for each element $u \in \Delta^I$, i.e., $I \models a : \forall R.C_1$. Else, for each object b such that aRb is in S, the $\rightarrow_\forall$ rule has been applied to $a : \forall R.C_1$ since S is complete. Thus, $b : C_1$ is in S if b is an arbitrary object such that aRb is in S. By induction hypothesis we obtain $I \models a : \forall R.C_1$. □

We are now going to show that the top consistency algorithm with $\mathcal{ALC}$-ABox $\mathcal{A}$ and concept D_0 as input results "top consistent" iff D_0 is top consistent w.r.t. $\mathcal{A}$, i.e., iff there exists an interpretation I such that $I \models \mathcal{A}$ and $D_0^I = \top^I (= \Delta^I)$. To prove this, we introduce two lemmata which state: If we start with a constraint system S_0 which is induced by an $\mathcal{ALC}$-ABox and a concept, the top consistency algorithm can construct a chain $S_0 \rightarrow_1 S_1 \rightarrow_2 \ldots \rightarrow_n S_n$ such that S_n is complete and clash-free iff D_0 is top consistent.

Lemma 4.4 *Let $\mathcal{A}$ be an $\mathcal{ALC}$-ABox, let D_0 be a concept, and let S_0 be the constraint system which is induced by $\mathcal{A}$ and D_0. If D_0 is top consistent w.r.t. $\mathcal{A}$ then the top* *consistency algorithm can construct a finite chain* $S_0 \rightarrow_1 S_1 \rightarrow_2 \ldots \rightarrow_n S_n$ *with* $\rightarrow_i \in \{\rightarrow_\sqcap, \rightarrow_\sqcup, \rightarrow_\forall, \rightarrow_{\exists_1}, \rightarrow_{\exists_2}\}$ *such that* S_n *is complete and clash-free.*

Proof: We will show that there exists a chain $S_0 \rightarrow_1 S_1 \rightarrow_2 \ldots \rightarrow_n S_n$, where $\rightarrow_i$ is the first rule in the sequence $\rightarrow_\sqcap, \rightarrow_\sqcup, \rightarrow_\forall, \rightarrow_{\exists_1}, \rightarrow_{\exists_2}$ which is applicable to S_i, such that

(*i*) each S_i is satisfiable ($i = 0, \ldots, n$) and

(*ii*) there is no more propagation rule applicable to S_n.

This will be done by induction over the number i of rule applications. Since we assumed D_0 to be top consistent w.r.t. $\mathcal{A}$, there exists an interpretation I such that $I \models \mathcal{A}$ and $D_0^I = \Delta^I$, i.e., there exists an interpretation I such that $I \models s$ for each constraint s in S_0. Thus, we can assume that there exists an interpretation I_i which satisfies S_i. There are five possibilities for $S_i \rightarrow_{i+1} S_{i+1}$: If the $\rightarrow_\sqcap$ rule is applicable to S_i, let $S_i \rightarrow_\sqcap S_{i+1}$. Then $I_{i+1} := I_i$ obviously satisfies S_{i+1}.

Else, if the $\rightarrow_\sqcup$ rule is applicable to $a : C_1 \sqcup C_2$ in S_i, let S_{i+1} be $S_i \cup \{a : C_1\}$ if $I_i \models a : C_1$, and $S_i \cup \{a : C_2\}$ if $I_i \not\models a : C_1$. Again, $I_{i+1} := I_i$ satisfies S_{i+1}.

Else, if the $\rightarrow_\forall$ rule is applicable to $a : \forall R.C$ and aRb in S_i, let $S_i \rightarrow_\forall S_{i+1}$. Then, S_i is extended by $b : C$. By induction hypothesis we know that $I_i \models aRb$ and $I_i \models a : \forall R.C$. As an immediate consequence, $I_{i+1} := I_i$ satisfies S_{i+1}.

Else, if the $\rightarrow_{\exists_1}$ rule is applicable to $a : \exists R.C$ in S_i, let $S_i \rightarrow_{\exists_1} S_{i+1}$. Then, S_{i+1} extends S_i by the elements aRb, $b : C$, and $b : D_0^*$. Thereby, b is a new object. By induction hypothesis we know $I_i \models a : \exists R.C$, i.e., the exists an element $u \in U^{I_i}$ such that $R^{I_i}(a^{I_i}, u)$ and $u \in C^{I_i}$. If I_{i+1} is identical with I_i but $b^{I_{i+1}} := u$, thus $I_{i+1} \models aRb$ and $I_{i+1} \models b : C$. Furthermore, $b^{I_{i+1}} = u \in U^{I_i} = U^{I_{i+1}}$, i.e., $I_{i+1} \models b : D_0^*$ since $D_0^{I_i} = U^{I_i}$ and D_0 is equivalent to its negation normal form D_0^*.

Finally, if only the $\rightarrow_{\exists_2}$ rule is applicable to $a : \exists R.C$ in S_i, let $S_i \rightarrow_{\exists_2} S_{i+1}$. It is easy to verify that after applying the $\rightarrow_{\exists_2}$ rule once, no other propagation rule will be applicable any more: The $\rightarrow_{\exists_2}$ rule extends S_i by aRb where b is an object occuring in S_i. Obviously, the only precondition which could be satisfied as a consequence of adding aRb to S_i is the precondition of the $\rightarrow_\forall$ rule which—theoretically—could extend the constraint system by $b : C$ for some concept C. But since the $\rightarrow_{\exists_2}$ rule has been applied to S_i, the constraint $b : C$ must be in S_i and thus the $\rightarrow_\forall$ rule cannot be applied. Thus, when applying $\rightarrow_{\exists_2}$ to $a : \exists R.C$ all information about objects b with aRb has been made explicit and is satisfied by I. Therefore, the interpretation I_{i+1} which is identical to I_i, but where in adition $R^{I_{i+1}}(a^{I_i}, b^{I_i})$ holds, satisfies S_{i+1}.

Summing up, there exists a chain $S_0 \rightarrow_1 S_1 \rightarrow_2 \ldots \rightarrow_n S_n$, with $S_i \rightarrow_{i+1} S_{i+1}$ by the first propagation rule in the sequence $\rightarrow_\sqcap, \rightarrow_\sqcup, \rightarrow_\forall, \rightarrow_{\exists_1}, \rightarrow_{\exists_2}$ which is applicable to S_i, such that each S_i is satisfiable. Because of Lemma 4.2, the top consistency algorithm cannot construct an infinite chain $S_0 \rightarrow_1 S_1 \rightarrow_2 \ldots$, such that we obtain a complete system S_n after n rule applications. Furtheremore, since S_n is satisfiable, S_n cannot contain a clash. □

Lemma 4.5 *Let $\mathcal{A}$ be an $\mathcal{ALC}$-ABox, let D_0 be a concept, and let S_0 be the constraint system which is induced by $\mathcal{A}$ and D_0. If the top consistency algorithm can construct a chain $S_0 \rightarrow_1 S_1 \rightarrow_2 \ldots \rightarrow_n S_n$ with $\rightarrow_i \in \{\rightarrow_\sqcap, \rightarrow_\sqcup, \rightarrow_\forall, \rightarrow_{\exists_1}, \rightarrow_{\exists_2}\}$ such that S_n is complete and clash-free, then D_0 is top consistent.*

Proof: Since S_n is complete and clash-free, S_n is satisfiable because of Lemma 4.3. In the proof of this lemma we especially showed that the following interpretation I satisfies S_n:

- Δ^I is the set of objects in S_n.
- $A^I := \{a \mid a : A \text{ is in } S_n\}$ for each atomic concept A in S_n.
- $R^I := \{(a, b) \mid aRb \text{ is in } S_n\}$ for each role R in S_n.
- $a^I := a \in \Delta^I$ for each object a in S_n.

We still have to show that $D_0^I = \top^I (= \Delta^I)$, i.e., $u \in D_0^I$ for each element $u \in \Delta^I$. This is equivalent to $I \models a : D_0^*$ for each object a in S_n because of the definition of Δ^I, and because the negation normal form D_0^* of D_0 is equivalent to D_0.

If a occurs in S_0, then $a : D_0^*$ is in S_0 by construction of the start constraint system S_0. In this case $I \models a : D_0^*$ since I satisfies S_n and $S_0 \subseteq S_n$. If, on the other hand, a does not occur in S_0 then a has been added to some constraint system S_i $(0 \leq i < n)$ by the $\rightarrow_{\exists_1}$ rule. This rule then also has added $a : D_0^*$ to S_i. Because of $I \models S_n$ and $S_i \subseteq S_n$ in this case $I \models a : D_0^*$ holds. □

Summing up the results in this subsection we obtain the following theorem.

Theorem 4.6 *Let $\mathcal{A}$ be an $\mathcal{ALC}$-ABox and let D_0 be a concept. Then the top consistency algorithm with input $\mathcal{A}$ and D_0 terminates and results "top consistent" iff D_0 is top consistent w.r.t. $\mathcal{A}$.*

Proof: Because of Lemma 4.2, the top consistency algorithm only constructs finite chains $S_0 \rightarrow_1 S_1 \rightarrow \ldots$ with $\rightarrow_i \in \{\rightarrow_\sqcap, \rightarrow_\sqcup, \rightarrow_\forall, \rightarrow_{\exists_1}, \rightarrow_{\exists_2}\}$. Except from the $\rightarrow_\sqcup$ rule all propagation rules determine exactly one new constraint system. The $\rightarrow_\sqcup$ rule determines exactly two possible constraint systems, i.e., there is only a finite number of possible finite chains since S_0 is finite. Thus, the top consistency algorithm terminates. Because of Lemmata 4.4 and 4.5 we know D_0 to be top consistent w.r.t. $\mathcal{A}$ iff there exists a chain $S_0 \rightarrow_1 \ldots \rightarrow_n S_n$ such that S_n is complete and clash-free. Exactly this is tested by the top consistency algorithm. □

5 Computing $\mathcal{ALC}_\mathcal{K}$ Inferences

Now we are going to show how to decide whether or not a given formula is a logical consequence from a set of $\mathcal{ALC}_\mathcal{K}$-axioms. Therefore, we start with a set of $\mathcal{ALC}_\mathcal{K}$-axiom which describe the actual world as well as the knowledge of agents. As implied by using the word "axiom", these formulas are assumed to be true under all circumstances. In contrast to this we now introduce the notion of *$\mathcal{ALC}_\mathcal{K}$-formulas* which have the same syntax and semantics as $\mathcal{ALC}_\mathcal{K}$-axioms but differ in the intuitive meaning: While $\mathcal{ALC}_\mathcal{K}$-axioms will only be used to define an axiomatization of a world and of agents' knowledge, some $\mathcal{ALC}_\mathcal{K}$-formulas may be entailed by such an axiomatization while some others may not be entailed. That means, we need a test whether an $\mathcal{ALC}_\mathcal{K}$-formula is a logical consequence from a set of $\mathcal{ALC}_\mathcal{K}$-axioms.

We will show how to use the $S4$-satisfiability algorithm to test whether or not a given $\mathcal{ALC}_\mathcal{K}$-formula is a logical consequence from a set $F_1, \ldots, F_n$ of $\mathcal{ALC}_\mathcal{K}$-axioms. Again, we are only interested in $S4$ Kripke structures and thus define: F is an $S4$ *consequence* of $F_1, \ldots, F_n$ iff for each $S4$ Kripke structure $K = (\mathcal{W}, \Gamma, K_I)$ and for each world w in $\mathcal{W}$ holds: if $K, w \models F_1, \ldots, F_n$, then $K, w \models F$.

Firstly, let F be an $\mathcal{ALC}_\mathcal{K}$-formula of the form $\Box^*(C = D)$, $\Box^*(C \neq D)$, or $\Box^*(a : C)$, where $\Box^*$ is an abbreviation for a possibly empty sequence of modalities. Then, F is an $S4$ consequence of $F_1, \ldots, F_n$ iff the set $F_1, \ldots, F_n, [\neg F]^*$ of $\mathcal{ALC}_\mathcal{K}$-formulas is not $S4$-satisfiable, where $[\neg F]^*$ denotes the negation normal form of $\neg F$. Note, that $\neg F$ is an $\mathcal{ALC}_\mathcal{K}$-formula if F is of the above described form.

If, on the other hand, F is of the form $\Box^*(aRb)$, where $\Box^*$ is an abbreviation for a possibly empty sequence of non-negated indexed $\Box$ operators, we cannot use this test method since negation signs are not allowed in $\mathcal{ALC}_\mathcal{K}$-formulas which contain a role instance. To handle this case, we extend the notion of $\mathcal{ALC}_\mathcal{K}$-formulas as follows: if R is a role, a, b are objects, and $i_1, \ldots, i_m$ are agents, then $\Diamond_{i_1} \ldots \Diamond_{i_m}(aRb)$ is an $\mathcal{ALC}_\mathcal{K}$-formula.

Alternatively, these $\mathcal{ALC}_\mathcal{K}$-formulas could be defined by $\circ_{i_1} \ldots \circ_{i_m} (aR'b)$ where (i) each $\circ_{i_j}$ is either $\Box_{i_j}$ or $\neg\Box_{i_j}$, (ii) R' is either R or $\neg R$, and (iii) the number of negation signs in $\circ_{i_1} \ldots \circ_{i_m} (aR'b)$ is even. Using this definition it is easy to see that the negation normal form of the new $\mathcal{ALC}_\mathcal{K}$-formulas does not contain negation of roles. Therefore, on a technical level we could allow such formulas as $\mathcal{ALC}_\mathcal{K}$-axioms in Section 2. But a restriction like "the number of negation signs is even" seems not to be adequate when defining a language to describe knowledge of agents. However, for testing whether or not an $\mathcal{ALC}_{\mathcal{K}}$-formula is entailed by a set $F_1, \ldots, F_n$ of $\mathcal{ALC}_{\mathcal{K}}$-axioms, this definition turns out to be reasonable.

Note, that $S4$-satisfiability of a set of $\mathcal{ALC}_{\mathcal{K}}$-formulas can be handled by the $S4$-satisfiability algorithm in Section 3 even if we use the above introduced extended definition of $\mathcal{ALC}_{\mathcal{K}}$-formulas: Firstly, the algorithm only treats the modalities of $\mathcal{ALC}_{\mathcal{K}}$-formulas, i.e., it works independently from the syntactical structure of formulas without modalities. Secondly, satisfiability of the resulting $\mathcal{ALC}$ test set still can be tested, since they do not contain negation of roles. And, finally, it does not matter whether aRb is in an $\mathcal{ALC}$ test set because of $\Box_{i_1} \ldots \Box_{i_m}(aRb)$, or because of $\Diamond_{i_1} \ldots \Diamond_{i_m}(aRb)$. Summing up, when using the extended definition of $\mathcal{ALC}_{\mathcal{K}}$-formulas we need not to change the $S4$-satisfiability algorithm at all.

The following lemma provides a nice property of $\mathcal{ALC}_{\mathcal{K}}$-formulas and will be used in the proof theorem 5.2.

Lemma 5.1 *Let $F_1, \ldots, F_n$ be an $S4$-satisfiable set of $\mathcal{ALC}_{\mathcal{K}}$-axioms and let R' be a role which does not occur in $F_1, \ldots, F_n$. Then the set $F_1, \ldots, F_n, \Diamond_{i_1} \ldots \Diamond_{i_m}(aR'b)$ of $\mathcal{ALC}_{\mathcal{K}}$-formulas is $S4$-satisfiable for each sequence $\Diamond_{i_1} \ldots \Diamond_{i_m}$ of indexed $\Diamond$ operators and for each pair a, b of objects.*

Proof: Let W' be the result of the frame algorithm with input $F_1, \ldots, F_n$. Since we supposed these $\mathcal{ALC}_{\mathcal{K}}$-axioms to be $S4$ satisfiable, the $\mathcal{ALC}$ test set $A(w)$ is satisfiable for each label w in W'. Thus, especially the $\mathcal{ALC}$ test set $A(w_0)$ of the real world w_0 is satisfiable.

Let us reconsider the $S4$ Kripke structure K in the proof of Lemma 3.6, i.e.,

- $\mathcal{W}$ is given by the set of all labels in W'.
- Γ consists of one accessibility relation γ_i for each agent i. Thereby, γ_i is given by the reflexive and transitive closure of the set $\{(w, w') \mid w \bowtie_i w' \in W\}$.
- K_I is given such that $K, w \models F$ for each labelled $\mathcal{ALC}_{\mathcal{K}}$-formula $F \mid\mid w$ in W' where F does not contain any modality.

In the proof of Lemma 3.6 we have already shown that $K, w_0 \models F_1, \ldots, F_n$. Obviously, we can modify K_I such that $K, w_0 \models aR'b$ and $K, w_0 \models F_1, \ldots, F_n$. This is due to the fact that R' does not occur in $F_1, \ldots, F_n$ and thus does not occur in the $\mathcal{ALC}$ test set $A(w_0)$. Since $(w_0, w_0) \in \gamma_i$ for each agent i, in this case $K, w_0 \models \Diamond_{i_1} \ldots \Diamond_{i_m}(aR'b)$. □

Note, that this lemma does not hold for *arbitrary* Kripke structures $\mathcal{K}$. For example, it may hold $\mathcal{K}, w \models \Box_i(a : C), \Box_i(a : \neg C)$ if $\mathcal{K}$ is a Kripke structure such that there is no world accessible from w by agent i. But, obviously, $\mathcal{K}, w \not\models \Box_i(a : C), \Box_i(a : \neg C), \Diamond_i(aR'b)$.

For the following theorem we define syntactical operations on sequences of indexed $\Box$ operators. The $S4$ *normal form* of $\Box_{i_1} \ldots \Box_{i_m}$ is given by successively replacing each occurrance of a subsequence $\Box_j \ldots \Box_j$ in $\Box_{i_1} \ldots \Box_{i_m}$ by $\Box_j$. For example, the $S4$ normal form of $\Box_4\Box_2\Box_2\Box_3\Box_1\Box_1\Box_1$ is given by $\Box_4\Box_2\Box_3\Box_1$. Conversely, an *expanded version* of a sequence $\Box_{i_1} \ldots \Box_{i_m}$ is given by replacing one or more operators $\Box_j$ by a sequence $\Box_j \ldots \Box_j$. Using these definitions, we say $\Box_{i_1} \ldots \Box_{i_m}$ *matches* a sequence $\Diamond_{j_1} \ldots \Diamond_{j_k}$ iff $\Box_{j_1} \ldots \Box_{j_k}$ is an expanded version of the $S4$ normal form of $\Box_{i_1} \ldots \Box_{i_m}$. For example, $\Box_1\Box_1\Box_2$ matches $\Diamond_1\Diamond_2$ and $\Diamond_1\Diamond_2\Diamond_2$, but it neither matches $\Diamond_1\Diamond_2\Diamond_3$ nor $\Diamond_2\Diamond_1$.

Theorem 5.2 provides a test whether or not an $\mathcal{ALC}_\mathcal{K}$-formula $\Box_{i_1} \ldots \Box_{i_m}(aRb)$ is entailed by a set of $\mathcal{ALC}_\mathcal{K}$-axioms.

Theorem 5.2 *Let $F_1, \ldots, F_n$ be an $S4$-satisfiable set of $\mathcal{ALC}_\mathcal{K}$-axioms and let F be an $\mathcal{ALC}_\mathcal{K}$-formula of the form $\Box_{i_1} \ldots \Box_{i_m}(aRb)$, where $\Box_{i_1} \ldots \Box_{i_m}$ is a possibly empty sequence of indexed $\Box$ operators. Then F is an $S4$ consequence of $F_1, \ldots, F_n$ iff one of the F_j is of the form $\Box_*\Box^M(aRb)$ where $\Box_*$ is a possibly empty sequence of indexed $\Box$ operators, and $\Box^M$ is a sequence of indexed $\Box$ operators which matches $\Diamond_{i_1} \ldots \Diamond_{i_m}$.*

Proof: Firstly, let F be of the form aRb. If one of the F_j is of the form $\Box_*(aRb)$, then $K, w \models aRb$ holds for each $S4$ Kripke structure K with $K, w \models F_j$. This is an immediate consequence of property (P1) of $S4$ Kripke structures.

Conversely, suppose that none of the F_j is of the form $\Box_*(aRb)$. Let W be the world constraint system $\{F_1 \,||\, w_0, \ldots, F_n \,||\, w_0\}$ which is induced by $F_1, \ldots, F_n$ and let W' be the result of the frame algorithm with input W. Since the set $F_1, \ldots, F_n$ of $\mathcal{ALC}_\mathcal{K}$-axioms is $S4$-satisfiable, the $\mathcal{ALC}$ test set $A(w)$ of each label w in W' is satisfiable. Furthermore, since none of the F_j is of the form $\Box_*(aRb)$, the $\mathcal{ALC}$ test set $A(w_0)$ does not contain the formula aRb. It is easy to verify that in this case $A(w_0)$ is satisfiable by an interpretation I such that $I \not\models aRb$ (e.g., by considering the propagation rules in [Hol90]). Let now K be the $S4$ Kripke structure which is constructed from W' as in the proof of Lemma 3.6, but K_I is modified such that $K, w_0 \not\models aRb$. Then, $K, w_0 \models F_1, \ldots, F_n$ but $K, w_0 \not\models aRb$, i.e., aRb is not an $S4$ consequence of $F_1, \ldots, F_n$.

Suppose now, we want to test whether or not $\Box_{i_1} \ldots \Box_{i_m}(aRb)$ is an $S4$ consequence of $F_1, \ldots, F_n$. This is equivalent to testing whether or not $F_1, \ldots, F_n, \Diamond_{i_1} \ldots \Diamond_{i_m}(a\neg Rb)$ is $S4$-satisfiable. Since $\Diamond_{i_1} \ldots \Diamond_{i_m}(a\neg Rb)$ is not an $\mathcal{ALC}_\mathcal{K}$-axiom, this case cannot be handled by the $S4$-satisfiability algorithm of Section 3. Alternatively, let us have a look at the application of the frame algorithm to the world constraint system W which is induced by $\{F_1, \ldots, F_n, \Diamond_{i_1} \ldots \Diamond_{i_m}(aR'b)\}$ where R' is a role which does not occur in $F_1, \ldots, F_n$. Because of Lemma 5.1, $F_1, \ldots, F_n, \Diamond_{i_1} \ldots \Diamond_{i_m}(aR'b)$ is $S4$-satisfiable. Thus, this application of the frame algorithm results a world constraint system W' such that the $\mathcal{ALC}$ test set $A(w)$ is satsifiable for each label w in W'. Furthermore, it is easy to verify that there is exactly one label, say $\tilde{w}$, in W' such that $A(\tilde{w})$ contains the formula $aR'b$.

Let us now consider R' as an abbreviation for $\neg R$. Obviously, this does not influence the construction of W' but it may influence satisfiability of the $\mathcal{ALC}$ test set $A(\tilde{w})$. As mentioned above, this test set $A(\tilde{w})$ is unsatisfiable iff it contains aRb as well, i.e., iff $aRb \,||\, \tilde{w}$ is in W'. It is easy to check that $aRb \,||\, \tilde{w}$ is in W' iff $\Box_{k_1} \ldots \Box_{k_l}(aRb) \,||\, w_0$ is in W', whereby $\Box_{k_1} \ldots \Box_{k_l}$ matches $\Diamond_{i_1} \ldots \Diamond_{i_m}$: Since $\Diamond_{i_1} \ldots \Diamond_{i_m}(aR'b)||w_0$ is in W', there are world constraints $w_0 \bowtie_{i_1} w_1, \ldots, w_{m-1} \bowtie_{i_m} w_m (= \tilde{w})$ in W'. If $\Box_{i_1} \ldots \Box_{i_m}(aRb) \,||\, w_0$ is in W', then $aRb \,||\, w_m$ is in W' because of the definition of the $\rightarrow_\Diamond$ and the $\rightarrow_\Box$ rule. Replacing some operator $\Box_{i_j}$ in $\Box_{i_1} \ldots \Box_{i_m}$ by $\Box_{i_j}\Box_{i_j}$ does not influence the existence of $aRb \,||\, \tilde{w}$ in W' since $\Box_{i_j}F \,||\, w$ is in W' if $\Box_{i_j}\Box_{i_j}F \,||\, w$ is in W'. This holds for arbitrary formulas F and labels w because of the $\rightarrow_\Box$ rule. Replacing a sequence $\Box_{i_j} \ldots \Box_{i_j}$ in $\Box_{i_1} \ldots \Box_{i_m}$ by $\Box_{i_j}$ doesn't influence the existence of $aRb \,||\, \tilde{w}$ in W' as well, since, by definition of the $\rightarrow_\Diamond$ and the $\rightarrow_\circ$ rule, if $\Box_{i_j}F \,||\, w$ is in W' then $\Box_{i_j}F \,||\, w'$ is in W' for each world w' such that the world constraints $w \bowtie_{i_j} w_1, \ldots, w_p \bowtie_{i_j} w'$ are all in W'. In other words, if the world constraints $\Diamond_{i_j} \ldots \Diamond_{i_j} G \,||\, w$ and $\Box_{i_j}F \,||\, w$, and $w \bowtie_{i_j} w_1, \ldots, w_p \bowtie_{i_j} w'$ are all in W', then $F \,||\, w'$ is in W'. It is easy to verify that there are no other possibilities such that $aRb \,||\, \tilde{w}$ is in W'.

Finally, $\Box_{k_1} \ldots \Box_{k_l}(aRb) \,||\, w_0$ is in W' iff one of the $\mathcal{ALC}_\mathcal{K}$-formulas F_j is of the form $\Box_*\Box_{k_1} \ldots \Box_{k_l}(aRb)$, where $\Box_*$ is a possibly empty sequence of indexed $\Box$ operators. This follows immediately by the above argumentation and the definition of the $\rightarrow_\Box$ rule. $\Box$

Now we have given algorithms for deciding $S4$-satisfiability of a given set of $\mathcal{ALC}_\mathcal{K}$-axioms, and, building upon this, for deciding whether or not a given $\mathcal{ALC}_\mathcal{K}$-formula F is an $S4$ consequence of a given set $F_1, \ldots, F_n$ of $\mathcal{ALC}_\mathcal{K}$-axioms. Let us finally present a possible application and a technical example.

Example 5.3 a) Suppose there are two shippings, s_1 and s_2, which are considered as agents in the following. Both agents are competitors and want to earn as much money as possible. We assume that there exist two transportation orders, o_1 and o_2. Thus, we need at least the following two $\mathcal{ALC}_\mathcal{K}$-axioms to describe the world.

(1) o_1 : transportation-order
(2) o_2 : transportation-order.

Both shippings know that transportation orders are orders they can carry out, called possible orders, and each agent knows that the other agent has this knowledge. This is represented by

$$
\begin{array}{ll}
(3) & \Box_{s_1}(\text{transportation-order} \sqsubseteq \text{possible-order}) \\
(4) & \Box_{s_2}(\text{transportation-order} \sqsubseteq \text{possible-order}) \\
(5) & \Box_{s_1}\Box_{s_2}(\text{transportation-order} \sqsubseteq \text{possible-order}) \\
(6) & \Box_{s_2}\Box_{s_1}(\text{transportation-order} \sqsubseteq \text{possible-order}).
\end{array}
$$

While both agents know that there is a transportation order o_1, only s_2 knows that there is still a second transportation order, namely o_2:

$$
\begin{array}{ll}
(7) & \Box_{s_1}(o_1 : \text{transportation-order}) \\
(8) & \Box_{s_2}(o_1 : \text{transportation-order}) \\
(9) & \neg\Box_{s_1}(o_2 : \text{transportation-order}) \\
(10) & \Box_{s_2}(o_2 : \text{transportation-order}).
\end{array}
$$

Finally, we suppose that s_1 knows that s_2 knows o_1 to be a possible order, while s_2 knows that s_1 does *not* know that o_2 is a possible order. This is represented by

$$
\begin{array}{ll}
(11) & \Box_{s_1}\Box_{s_2}(o_1 : \text{possible-order}) \\
(12) & \Box_{s_2}\neg\Box_{s_1}(o_2 : \text{possible-order}).
\end{array}
$$

It is easy to verify that the set $\{(1),\ldots,(12)\}$ of $\mathcal{ALC}_\mathcal{K}$-axioms is $S4$-satisfiable.

Provided that the agents can plan and reason on the basis of their knowledge, how do they act in the world? Let us firstly have a look at agent s_1. Obviously, he can conclude that o_1 is an order he can carry out because of (3) and (7) or, alternatively, because of (11). Analogously, he can conclude that also agent s_2 knows o_1 to be a possible order because of (11). Since he cannot derive the existence of another possible order he will offer a low price for order o_1.

In the same way, agent s_2 will conclude that both agents know o_1 to be a possible order. But additionally, he knows that there is still a second possible order, namely o_2, what can be derived from (4) and (10). Furthermore, he knows that agent s_1 does not know o_2 to be a possible order (because of (12)). Thus, he may act as follows: He will offer a high price for order o_2 since he cannot derive the existence of another shipping which knows o_2 to be a possible order. (Note that this can be risky, e.g., if there is another agent s_3 and s_2 does not know anything about the knowledge of agent s_3). For order o_1 he may offer a low or a medium price.

Summing up, the behaviour of agents in the world is not only influenced by their knowledge about the world, but may be influenced by their knowledge about the knowledge of other agents as well.

b) Suppose the following two $\mathcal{ALC}_\mathcal{K}$-axioms to be given:

$$
\begin{array}{ll}
(F_1) & \Box_a(\mathit{John} : \forall\, \mathit{owns}.\neg \mathit{gasoline\text{-}truck}) \\
(F_2) & \Box_a(\mathit{truck\text{-}1} : \mathit{gasoline\text{-}truck})
\end{array}
$$

Applying the frame algorithm to the world constraint system W which is induced by F_1, F_2 and the $\mathcal{ALC}_{\mathcal{K}}$-formula

$$(F) \qquad \Diamond_a(\textit{John owns truck-1})$$

we obtain the world constraint system W' which is given by

$$\begin{aligned}
\Box_a(\textit{John} : \forall\, \textit{owns}.\neg\textit{gasoline-truck}) &\,||\, w_0\\
\textit{John} : \forall\, \textit{owns}.\neg\textit{gasoline-truck} &\,||\, w_0\\
\Box_a(\textit{truck-1} : \textit{gasoline-truck}) &\,||\, w_0\\
\textit{truck-1} : \textit{gasoline-truck} &\,||\, w_0\\
\Diamond_a(\textit{John owns truck-1}) &\,||\, w_0\\
w_0 \bowtie_a w_1&\\
\textit{John owns truck-1} &\,||\, w_1\\
\Box_a(\textit{John} : \forall\, \textit{owns}.\neg\textit{gasoline-truck}) &\,||\, w_1\\
\textit{John} : \forall\, \textit{owns}.\neg\textit{gasoline-truck} &\,||\, w_1\\
\Box_a(\textit{truck-1} : \textit{gasoline-truck}) &\,||\, w_1\\
\textit{truck-1} : \textit{gasoline-truck} &\,||\, w_1
\end{aligned}$$

Obviously, the $\mathcal{ALC}$ test set $A(w_1)$ is unsatisfiable. That means, the $\mathcal{ALC}_{\mathcal{K}}$-formula $\Box_a(\textit{John} \neg\textit{owns truck-1})$, is an $S4$ consequence of F_1 and F_2.

Note, that we have concluded agent a to know that John does *not* own truck-1, though we cannot explicitely express an agents' knowledge about negated roles when using the definition of $\mathcal{ALC}_{\mathcal{K}}$-axioms in Section 2. This conclusion became possible because of the above extended definition of $\mathcal{ALC}_{\mathcal{K}}$-formulas for computing $\mathcal{ALC}_{\mathcal{K}}$-inferences. ■

6 Conclusion

We have presented an extension of the concept language $\mathcal{ALC}$ by a knowledge operator $\Box$ which is indexed by agents. This language can be used to describe a real world by an $\mathcal{ALC}$-TBox and an $\mathcal{ALC}$-ABox, i.e., by $\mathcal{ALC}_{\mathcal{K}}$-axioms without modalities. But additionally, it can be used to describe the knowledge agent i has about the world, about the knowledge of other agents, and about his own knowledge by $\mathcal{ALC}_{\mathcal{K}}$-axioms with the leading operator $\Box_i$.

In this paper we used an axiomatization of the knowledge operator which has been proposed by Moore [Moo80, Moo85]. We have given an algorithm for deciding whether a set $F_1, \ldots, F_n$ of $\mathcal{ALC}_{\mathcal{K}}$-axioms is $S4$-satisfiable. And, building upon this, we have shown how to test whether an $\mathcal{ALC}_{\mathcal{K}}$-formula F is an $S4$ consequence of $F_1, \ldots, F_n$. Both tests are of practical interest: The first one can be used to test consistency of the represented knowledge, and the second one to find out whether a given $\mathcal{ALC}_{\mathcal{K}}$-formula is implied by an agents' knowledge. An extension of our terminological representation system $\mathcal{KRIS}$ [BH91] with the knowledge operator $\Box$ is under work. Note, that the presented algorithms cannot directly be used for an implementation. Of course, appropriate data structures and optimization techniques have to be developed for concrete applications.

Future work will mainly concern with two questions. Firstly, how to extend the present approach, e.g., by operators $E\varphi$ (everyone knows φ) and $C\varphi$ (it is common knowledge that φ). Secondly, an interesting task will be to catalogue multi agent applications by deciding what the general properties of knowledge in these applications are, and to devise algorithms to handle the resulting axiomatizations of the knowledge operator.

Acknowledgements. I am grateful to Hans-Jürgen Bürckert, Bernhard Hollunder, and Andreas Nonnengart for many discussions on the topic of this paper and for reading earlier drafts.

References

[BG88] A. Bond and L. Gasser. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann, Los Angeles, CA, 1988.

[BH91] F. Baader and B. Hollunder. $\mathcal{KRIS}$: $\mathcal{K}$nowledge $\mathcal{R}$epresentation and $\mathcal{I}$nference $\mathcal{S}$ystem. *SIGART Bulletin*, 2(3):8–14, 1991.

[BS85] R. J. Brachman and J. G. Schmolze. An overwiev of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[Che80] B. F. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, 1980.

[Fit83] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*, volume 169 of *Synthese Library*. D. Reidel Publishing Company, 1983.

[GH89] L. Gasser and M.N. Huhns. *Distributed Artificial Intelligence, Volume II*. Research Notes in Artificial Intelligence. Morgan Kaufmann, San Mateo, CA, 1989.

[Gor92] R. Goré. Analytic tableaux for propositional modal logics of nonmonotonicity. Technical report, Department of Computer Science at the University of Manchester, England, 1992.

[Hal86] J. Y. Halpern. Reasoning about knowledge: an overview. In *Proceedings of TARK'86*, pages 1–15, 1986.

[HC68] G. E. Hughes and M. J. Cresswell. *An Introduction to Modal Logic*. J. W. Arrowsmith Ltd, Bristol, 1968.

[Hin62] J. Hintikka, editor. *Knowledge and Belief*. Cornell University Press, 1962.

[HM90] J. Y. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587, 1990.

[HM92] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.

[Hol90] B. Hollunder. Hybrid inferences in KL-ONE-based knowledge representation systems. In *14th German Workshop on Artificial Intelligence*, volume 251 of *Informatik-Fachberichte*, pages 38–47, Ebingerfeld, Germany, 1990. Springer.

[Huh87] M.N. Huhns. *Distributed Artificial Intelligence.* Pitman/Morgan Kaufmann, San Mateo, CA, 1987.

[Moo80] R. C. Moore. Reasoning about knowledge and action. Technical Report Technical Note 191, SRI International, 1980.

[Moo85] R. C. Moore. A formal theory of knowledge and action. In J. R. Hobbs and R. C. Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358. Ablex Publishing Corporation, 1985.

[MvdHV91a] J.-J. C. Meyer, W. van der Hoek, and G. A. W. Vreeswijk. Epistemic logic for computer science: A tutorial (part one). In *Bulletin of the EATCS*, volume 44, pages 242–270. European Association for Theoretical Computer Science, 1991.

[MvdHV91b] J.-J. C. Meyer, W. van der Hoek, and G. A. W. Vreeswijk. Epistemic logic for computer science: A tutorial (part two). In *Bulletin of the EATCS*, volume 45, pages 256–287. European Association for Theoretical Computer Science, 1991.

Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
D-6750 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all published papers so far can be ordered from the above address.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-92-11
Susane Biundo, Dietmar Dengler, Jana Koehler:
Deductive Planning and Plan Reuse in a Command Language Environment
13 pages

RR-92-13
Markus A. Thies, Frank Berger:
Planbasierte graphische Hilfe in objektorientierten Benutzungsoberflächen
13 Seiten

RR-92-14
Intelligent User Support in Graphical User Interfaces:
1. InCome: A System to Navigate through Interactions and Plans
   Thomas Fehrle, Markus A. Thies
2. Plan-Based Graphical Help in Object-Oriented User Interfaces
   Markus A. Thies, Frank Berger

22 pages

RR-92-15
Winfried Graf: Constraint-Based Graphical Layout of Multimodal Presentations
23 pages

RR-92-16
Jochen Heinsohn, Daniel Kudenko, Berhard Nebel, Hans-Jürgen Profitlich: An Empirical Analysis of Terminological Representation Systems
38 pages

RR-92-17
Hassan Aït-Kaci, Andreas Podelski, Gert Smolka:
A Feature-based Constraint System for Logic Programming with Entailment
23 pages

RR-92-18
John Nerbonne: Constraint-Based Semantics
21 pages

RR-92-19
Ralf Legleitner, Ansgar Bernardi, Christoph Klauck: PIM: Planning In Manufacturing using Skeletal Plans and Features
17 pages

RR-92-20
John Nerbonne: Representing Grammar, Meaning and Knowledge
18 pages

RR-92-21
Jörg-Peter Mohren, Jürgen Müller
Representing Spatial Relations (Part II) -The Geometrical Approach
25 pages

RR-92-22
Jörg Würtz: Unifying Cycles
24 pages

RR-92-23
Gert Smolka, Ralf Treinen:
Records for Logic Programming
38 pages

RR-92-24
Gabriele Schmidt: Knowledge Acquisition from Text in a Complex Domain
20 pages

RR-92-25
Franz Schmalhofer, Ralf Bergmann, Otto Kühn, Gabriele Schmidt: Using integrated knowledge acquisition to prepare sophisticated expert plans for their re-use in novel situations
12 pages

RR-92-26
Franz Schmalhofer, Thomas Reinartz, Bidjan Tschaitschian: Intelligent documentation as a catalyst for developing cooperative knowledge-based systems
16 pages

RR-92-27
Franz Schmalhofer, Jörg Thoben: The model-based construction of a case-oriented expert system
18 pages

RR-92-29
Zhaohui Wu, Ansgar Bernardi, Christoph Klauck: Skeletel Plans Reuse: A Restricted Conceptual Graph Classification Approach
13 pages

RR-92-30
Rolf Backofen, Gert Smolka
A Complete and Recursive Feature Theory
32 pages

RR-92-31
Wolfgang Wahlster
Automatic Design of Multimodal Presentations
17 pages

RR-92-33
Franz Baader: Unification Theory
22 pages

RR-92-34
Philipp Hanschke: Terminological Reasoning and Partial Inductive Definitions
23 pages

RR-92-35
Manfred Meyer:
Using Hierarchical Constraint Satisfaction for Lathe-Tool Selection in a CIM Environment
18 pages

RR-92-36
Franz Baader, Philipp Hanschke:
Extensions of Concept Languages for a Mechanical Engineering Application
15 pages

RR-92-37
Philipp Hanschke: Specifying Role Interaction in Concept Languages
26 pages

RR-92-38
Philipp Hanschke, Manfred Meyer:
An Alternative to Θ-Subsumption Based on Terminological Reasoning
9 pages

RR-92-40
Philipp Hanschke, Knut Hinkelmann: Combining Terminological and Rule-based Reasoning for Abstraction Processes
17 pages

RR-92-41
Andreas Lux: A Multi-Agent Approach towards Group Scheduling
32 pages

RR-92-42
John Nerbonne:
A Feature-Based Syntax/Semantics Interface
19 pages

RR-92-43
Christoph Klauck, Jakob Mauss: A Heuristic driven Parser for Attributed Node Labeled Graph Grammars and its Application to Feature Recognition in CIM
17 pages

RR-92-44
Thomas Rist, Elisabeth André: Incorporating Graphics Design and Realization into the Multimodal Presentation System WIP
15 pages

RR-92-45
Elisabeth André, Thomas Rist: The Design of Illustrated Documents as a Planning Task
21 pages

RR-92-46
Elisabeth André, Wolfgang Finkler, Winfried Graf, Thomas Rist, Anne Schauder, Wolfgang Wahlster: WIP: The Automatic Synthesis of Multimodal Presentations
19 pages

RR-92-47
Frank Bomarius: A Multi-Agent Approach towards Modeling Urban Traffic Scenarios
24 pages

RR-92-48
Bernhard Nebel, Jana Koehler:
Plan Modifications versus Plan Generation: A Complexity-Theoretic Perspective
15 pages

RR-92-49
Christoph Klauck, Ralf Legleitner, Ansgar Bernardi:
Heuristic Classification for Automated CAPP
15 pages

RR-92-50
Stephan Busemann:
Generierung natürlicher Sprache
61 Seiten

RR-92-51
Hans-Jürgen Bürckert, Werner Nutt:
On Abduction and Answer Generation through Constrained Resolution
20 pages

RR-92-52
Mathias Bauer, Susanne Biundo, Dietmar Dengler, Jana Koehler, Gabriele Paul: PHI - A Logic-Based Tool for Intelligent Help Systems
14 pages

RR-92-54
Harold Boley: A Direkt Semantic Characterization of RELFUN
30 pages

RR-92-55
John Nerbonne, Joachim Laubsch, Abdel Kader Diagne, Stephan Oepen: Natural Language Semantics and Compiler Technology
17 pages

RR-92-56
Armin Laux: Integrating a Modal Logic of Knowledge into Terminological Logics
34 pages

RR-92-58
Franz Baader, Bernhard Hollunder: How to Prefer More Specific Defaults in Terminological Default Logic
31 pages

RR-92-59
Karl Schlechta and David Makinson: On Principles and Problems of Defeasible Inheritance
14 pages

RR-93-02
Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans-Jürgen Profitlich, Thomas Rist: Plan-based Integration of Natural Language and Graphics Generation
50 pages

RR-93-03
Franz Baader, Berhard Hollunder, Bernhard Nebel, Hans-Jürgen Profitlich, Enrico Franconi: An Empirical Analysis of Optimization Techniques for Terminological Representation Systems
28 pages

RR-93-05
Franz Baader, Klaus Schulz: Combination Techniques and Decision Problems for Disunification
29 pages

RR-92-60
Karl Schlechta: Defaults, Preorder Semantics and Circumscription
18 pages

DFKI Technical Memos

TM-91-12
Klaus Becker, Christoph Klauck, Johannes Schwagereit: FEAT-PATR: Eine Erweiterung des D-PATR zur Feature-Erkennung in CAD/CAM
33 Seiten

TM-91-13
Knut Hinkelmann: Forward Logic Evaluation: Developing a Compiler from a Partially Evaluated Meta Interpreter
16 pages

TM-91-14
Rainer Bleisinger, Rainer Hoch, Andreas Dengel: ODA-based modeling for document analysis
14 pages

TM-91-15
Stefan Bussmann: Prototypical Concept Formation An Alternative Approach to Knowledge Representation
28 pages

TM-92-01
Lijuan Zhang: Entwurf und Implementierung eines Compilers zur Transformation von Werkstückrepräsentationen
34 Seiten

TM-92-02
Achim Schupeta: Organizing Communication and Introspection in a Multi-Agent Blocksworld
32 pages

TM-92-03
Mona Singh: A Cognitiv Analysis of Event Structure
21 pages

TM-92-04
Jürgen Müller, Jörg Müller, Markus Pischel, Ralf Scheidhauer: On the Representation of Temporal Knowledge
61 pages

TM-92-05
Franz Schmalhofer, Christoph Globig, Jörg Thoben: The refitting of plans by a human expert
10 pages

TM-92-06
Otto Kühn, Franz Schmalhofer: Hierarchical skeletal plan refinement: Task- and inference structures
14 pages

TM-92-08
Anne Kilger: Realization of Tree Adjoining Grammars with Unification
27 pages

DFKI Documents

D-92-06
Hans Werner Höper: Systematik zur Beschreibung von Werkstücken in der Terminologie der Featuresprache
392 Seiten

D-92-07
Susanne Biundo, Franz Schmalhofer (Eds.): Proceedings of the DFKI Workshop on Planning
65 pages

D-92-08
Jochen Heinsohn, Bernhard Hollunder (Eds.): DFKI Workshop on Taxonomic Reasoning Proceedings
56 pages

D-92-09
Gernod P. Laufkötter: Implementierungsmöglichkeiten der integrativen Wissensakquisitionsmethode des ARC-TEC-Projektes
86 Seiten

D-92-10
Jakob Mauss: Ein heuristisch gesteuerter Chart-Parser für attributierte Graph-Grammatiken
87 Seiten

D-92-11
Kerstin Becker: Möglichkeiten der Wissensmodellierung für technische Diagnose-Expertensysteme
92 Seiten

D-92-12
Otto Kühn, Franz Schmalhofer, Gabriele Schmidt: Integrated Knowledge Acquisition for Lathe Production Planning: a Picture Gallery (Integrierte Wissensakquisition zur Fertigungsplanung für Drehteile: eine Bildergalerie)
27 pages

D-92-13
Holger Peine: An Investigation of the Applicability of Terminological Reasoning to Application-Independent Software-Analysis
55 pages

D-92-14
Johannes Schwagereit: Integration von Graph-Grammatiken und Taxonomien zur Repräsentation von Features in CIM
98 Seiten

D-92-15
DFKI Wissenschaftlich-Technischer Jahresbericht 1991
130 Seiten

D-92-16
Judith Engelkamp (Hrsg.): Verzeichnis von Softwarekomponenten für natürlichsprachliche Systeme
189 Seiten

D-92-17
Elisabeth André, Robin Cohen, Winfried Graf, Bob Kass, Cécile Paris, Wolfgang Wahlster (Eds.): UM92: Third International Workshop on User Modeling, Proceedings
254 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-92-18
Klaus Becker: Verfahren der automatisierten Diagnose technischer Systeme
109 Seiten

D-92-19
Stefan Dittrich, Rainer Hoch: Automatische, Deskriptor-basierte Unterstützung der Dokumentanalyse zur Fokussierung und Klassifizierung von Geschäftsbriefen
107 Seiten

D-92-21
Anne Schauder: Incremental Syntactic Generation of Natural Language with Tree Adjoining Grammars
57 pages

D-92-23
Michael Herfert: Parsen und Generieren der Prolog-artigen Syntax von RELFUN
51 Seiten

D-92-24
Jürgen Müller, Donald Steiner (Hrsg.): Kooperierende Agenten
78 Seiten

D-92-25
Martin Buchheit: Klassische Kommunikations- und Koordinationsmodelle
31 Seiten

D-92-26
Enno Tolzmann:
Realisierung eines Werkzeugauswahlmoduls mit Hilfe des Constraint-Systems CONTAX
28 Seiten

D-92-27
Martin Harm, Knut Hinkelmann, Thomas Labisch: Integrating Top-down and Bottom-up Reasoning in COLAB
40 pages

D-92-28
Klaus-Peter Gores, Rainer Bleisinger: Ein Modell zur Repräsentation von Nachrichtentypen
56 Seiten

Integrating a Modal Logic of Knowledge into Terminological Logics

Armin Laux

RR-92-56
Research Report