Modal description logics: modalizing roles

Frank Wolter and Michael Zakharyaschev
Institut für Informatik, Universität Leipzig
Augustus-Platz 10-11, 04109 Leipzig, Germany,
e-mail: wolter@informatik.uni-leipzig.de;
Keldysh Institute for Applied Mathematics
Russian Academy of Sciences
Miusskaya Square 4, 125047 Moscow, Russia
e-mail: mz@cs.msu.su

Abstract

In this paper, we construct a new concept description language intended for representing dynamic and intensional knowledge. The most important feature distinguishing this language from its predecessors in the literature is that it allows applications of modal operators to *all* kinds of syntactic terms: concepts, roles and formulas. Moreover, the language may contain both local (i.e., state-dependent) and global (i.e., state-independent) concepts, roles and objects. All this provides us with the most complete and natural means for reflecting the dynamic and intensional behaviour of application domains. We construct a satisfiability checking (mosaic-type) algorithm for this language (based on $\mathcal{ALC}$) in (i) arbitrary multimodal frames, (ii) frames with universal accessibility relations (for knowledge) and (iii) frames with transitive, symmetrical and euclidean relations (for beliefs). On the other hand, it is shown that the satisfaction problem becomes undecidable if the underlying frames are arbitrary strict linear orders, $\langle \mathbb{N}, < \rangle$, or the language contains the common knowledge operator for $n \geq 2$ agents.

1 Introduction

Description logics are often characterized as logic-based formalisms intended for representing knowledge about concept hierarchies and supplied with effective reasoning procedures and a Tarski-style declarative semantics. A standard example is the description logic $\mathcal{ALC}$ (see [Schmidt-Schauß and Smolka, 1991]) in the syntax of which the "definition" above can be represented as follows:

$$\mathsf{Description\_logic} = \mathsf{Knowledge\_representation\_language} \wedge$$
$$\mathsf{Logic} \wedge \exists \mathsf{is\_decided\_by.Algorithm} \wedge \exists \mathsf{has.Tarski\_semantics},$$
$$\mathcal{ALC} : \mathsf{Description\_logic}.$$

Here Description_logic, Knowledge_representation_language, Logic, Algorithm, Tarski_semantics are concept names (unary predicates), is_decided_by and has are role names (binary predicates) and $\mathcal{ALC}$ is an object name (individual constant).

Created in the 1980s as a direct successor of semantic networks and Minsky frames, description logic has found a wide range of applications (e.g. to information systems [Catarci and Lenzerini, 1993], databases [Borgida, 1995], software engineering [Wright *et al.*, 1993]), a dozen of knowledge representation systems based on various description logics has been implemented on computer (e.g. BACK [Quantz and Kindermann, 1990; Peltason, 1991], CLASSIC [Borgida *et al.*, 1989; Brachman *et al.*, 1991], DLP [Patel-Schneider, 1998], FACT [Horrocks, 1998], KRIS [Baader and Hollunder, 1991], and SB-ONE [Kobsa, 1989]), and a rich family of description languages of different expressive power and complexity has been constructed (see e.g. [Brachman and Schmolze, 1985; Donini *et al.*, 1996; Calvanese *et al.*, 1999]). But as the application areas are becoming more and more sophisticated, new, more expressive description logics are being called for. Sometimes it is possible to comply with the application demands by enriching a "standard" description language with new constructs and retaining basically the same semantical paradigm. For instance, De Giacomo and Lenzerini [1996] extend $\mathcal{ALC}$ by providing means to form the union, composition, inversion, transitive reflexive closure of roles and to use the number restrictions for quantification over roles, while Baader and Hanschke [1991] enrich $\mathcal{ALC}$ with concrete domains. However, some constructs require more drastic changes in the standard semantics. This happens, for instance, when one has to take into account various dynamic aspects of knowledge representation, like time-, agent- or action-dependence of knowledge.

Several approaches to the design of "dynamic" description logics were developed in the 1990s (see e.g. [Schmiedel, 1990; Schild, 1993; Laux, 1994; Gräber *et al.*, 1995; Baader and Ohlbach, 1995; Baader and Laux, 1995; Donini *et al.*, 1992; Wolter and Zakharyaschev, 1998c; Wolter and Zakharyaschev, 1998a; Wolter and Zakharyaschev, 1998b]), and all of them share one important feature: their models become multi-dimensional in the sense that besides the usual "object dimension" they may contain a time axis, possible worlds or states for beliefs or actions, etc.

Perhaps the most general multi-dimensional perspective was proposed by Baader and Ohlbach [1993; 1995]. Roughly, each dimension (object, time, belief, etc.) is represented by a set D_i (of objects, moments of time, possible belief worlds, etc.), concepts are interpreted as subsets of the Cartesian product $\prod_{i=1}^{n} D_i$ and roles of dimension i as binary relations between n-tuples that may differ only in the ith coordinate. And one can quantify over roles not only concepts, but also roles themselves and concept equations (examples are given below). However, the constructed language turned out to be too expressive. At least no sound and complete reasoning procedure for it has appeared (Baader and Ohlbach provide only a sound satisfiability algorithm for a restricted fragment of their language). Moreover, under the natural assumption that some dimensions may be "independent" the language becomes undecidable.[1]

Trying to simplify this semantics, Baader and Laux [1995] noticed that different dimensions may have a different status. For instance, time should probably be the same for all objects inhabiting the object dimension of our knowledge base. This observation led to a somewhat more transparent semantics: models now consist of worlds (or states) which represent—in terms of some standard

[1]Franz Baader has kindly informed us that the language is undecidable without this assumption as well.

description logic—the "current states of affairs"; these worlds may change with time passing by or under certain actions, or they may have a number of alternative worlds reflecting the beliefs of agents, and the connection between concepts and roles from different worlds is described by means of the corresponding temporal, dynamic, epistemic, or some other "modal" operators.

There are several "degrees of freedom" within this semantical paradigm.

1. The worlds in models may have arbitrary, or expanding (with respect to the accessibility relation between worlds), or constant domains. Of course, the choice depends on the application we deal with. However, from the technical point of view the most important is the *constant domain assumption*: it will be shown that if the satisfaction problem is decidable in models with constant domains then it is decidable in models with expanding domains as well. This is the reason why in this paper we adopt the constant domain assumption.

2. The concept, role and object names of the underlying description language may be *local* or *global*. Global names have the same extensions in all worlds, while local ones may have different extensions. (For example, an agent A may regard the role loves to be local, while the role believes to be global.) In principle, we may need both kinds of names. However technically, local object names present no difficulty as compared with global ones, and global concepts are expressible via local concepts and the modal operators.

3. In general, one may need modal operators applicable to concepts, roles and formulas. For example,

$$\mathsf{Received} = \mathsf{Mail} \wedge \langle \textit{sometime in the past} \rangle\, \exists \mathsf{is\_in}.\mathsf{Mailbox},$$
$$John : \exists [always] \mathsf{loves}.\mathsf{Woman},$$
$$[John\ believes]\, \langle eventually \rangle\, (\exists \mathsf{loves}.\top = \top)$$

(i.e., a received mail has been put into the mailbox some time ago, John will always love the same woman, and John believes that sometime in the future everybody will love somebody).

And finally, depending on the application domain we may choose between various kinds of modal operators (e.g. temporal, epistemic, action, etc.), the corresponding accessibility relations (say, linear for time, universal for knowledge, arbitrary for actions), and between the underlying pure description logics.

The main objective of this paper is to analyze a number of basic multi-dimensional modal description logics based on $\mathcal{ALC}$ and having the most expressive combination of the listed parameters. In particular, we show that the satisfaction problem (and so many other reasoning problems as well) for the logics with modal operators applicable to arbitrary concepts, (local and global) roles and formulas is *decidable* in the class of all (multi-modal) frames, in the class of universal frames (corresponding to the modality "agent A knows") and in the class of transitive, symmetrical and euclidean frames (corresponding to the modality "agent A believes").

Multi-dimensional modal description logics of such a great expressive power have never been considered in the literature. Languages with modal operators applicable only to axioms were studied by Finger and Gabbay [1992] and Laux [1994]; Schild [1993] allows applications of temporal operators only to concepts. Baader and Laux [1995] prove the decidability of the satisfaction problem for $\mathcal{ALC}$ extended with modal operators applicable to concepts and axioms, but only in the class of arbitrary frames and under the expanding domain assumption. Wolter and Zakharyaschev [1998c; 1998a; 1998b] have obtained a series of decidability results for the most important epistemic, temporal and dynamic description logics (based on the expressive description logic $\mathcal{CIQ}$ of [Giacomo and Lenzerini, 1996]) under the constant domain assumption and with modal operators applicable to both concept and formulas.

However, the computational behaviour of the modalized roles (i.e., binary predicates) has remained unclear. It should be emphasized that this problem is not of only technical interest. Modalized roles are really required for expressing the dynamic features of roles while passing from one state to another (which is usually much more difficult than to reflect the dynamic behaviour of concepts). For instance, to describe the class of people always voting for the same party we can use the axiom

$$\mathsf{Faithful\_voter} = \mathsf{Voter} \wedge \exists[always]\mathsf{votes.Party}.$$

(By swapping $\exists$ and $[always]$ we get the class of people always voting for *some* party.)

The price we have to pay for this extra expressive power is that only a limited number of logics in this language enjoy decidability. We show, for instance, that the satisfaction problem in linear frames or in universal frames with the common knowledge operator for $n \geq 2$ agents is undecidable (but it becomes decidable if the language contains neither global nor modalized roles).

To simplify presentation, we will be considering first description logics with only one modal operator and only local roles. Then we will generalize the obtained results to systems of multimodal description logic with both local and global roles.

The organization of the paper is as follows. In the next section we define the modal description language $\mathcal{ALC}_{\mathcal{M}}$ and its semantics. We show also that the satisfaction problem for $\mathcal{ALC}_{\mathcal{M}}$-formulas in models with expanding domains can be reduced to the same problem but in models with constant domains. In Section 3 we represent $\mathcal{ALC}_{\mathcal{M}}$-models in a special form—so-called quasimodels—and then, in Section 4, show that these quasimodels can be constructed like mosaics from a finite number of finite pattern pieces. Section 5 and Section 6 prove the decidability and undecidability results mentioned above. A number of open problems are discussed in Section 7.

Acknowledgments: The work of the second author was partially supported by U.K. EPSRC grant no. GR/M36748.

2 The language and its models

Definition 1 (language). The primitive symbols of the *modal concept description language* $\mathcal{ALC}_{\mathcal{M}}$ are:

- *concept names*: $C_0, C_1, \ldots$;
- *role names*: $R_0, R_1, \ldots$;
- *object names*: $a_0, a_1, \ldots$.

Starting from these we construct compound *concepts* and *roles* in the following way. Let R be a role, C, D concepts, and let $\Box$ and $\Diamond$ be the (dual) "necessity" and "possibility" operators, respectively. Then

- $\Diamond R$, $\Box R$ are roles, and
- $\top$, $C \wedge D$, $\neg C$, $\Diamond C$, $\exists R.C$ are concepts.

Atomic formulas are expressions of the form $\top$, $C = D$, aRb, $a : C$, where a and b are object names. If φ and ψ are formulas then so are $\Diamond\varphi$, $\neg\varphi$, and $\varphi \wedge \psi$.

Other boolean operators and the "necessity" operator $\Box$ (for concepts and formulas) are defined as abbreviations in the standard way; for instance, $\varphi \to \psi = \neg(\varphi \wedge \neg\psi$ and $\Box\varphi = \neg\Diamond\neg\varphi$.

For a formula φ, let us denote by $ob\varphi$, $con\varphi$, $rol\varphi$ and $sub\varphi$ the sets of all object names, concepts, roles, and subformulas occurring in φ, respectively.

Definition 2 (model). An $\mathcal{ALC}_{\mathcal{M}}$*-model* based on a frame $\mathfrak{G} = \langle W, \lhd \rangle$[2] is a pair $\mathfrak{M} = \langle \mathfrak{G}, I \rangle$ in which I is a function associating with each $w \in W$ an $\mathcal{ALC}$-model

$$I(w) = \left\langle \Delta, R_0^{I,w}, \ldots, C_0^{I,w}, \ldots, a_0^{I,w}, \ldots \right\rangle,$$

where Δ is a non-empty set, the *domain* of $\mathfrak{M}$, $R_i^{I,w}$ are binary relations on Δ, $C_i^{I,w}$ subsets of Δ, and $a_i^{I,w}$ are objects in Δ such that $a_i^{I,u} = a_i^{I,v}$, for any $u, v \in W$.

Remark 3. Without loss of generality we may identify $a_i^{I,w}$ with a_i, thus assuming that all object names belong to Δ.

Definition 4 (satisfaction). For a model $\mathfrak{M} = \langle \mathfrak{G}, I \rangle$ and a world w in it, the *values* $C^{I,w}$, $R^{I,w}$ of a concept C and a role R in w, and the *truth-relation* $(\mathfrak{M}, w) \models \varphi$ (or simply $w \models \varphi$, if $\mathfrak{M}$ is understood) are defined inductively as follows:

1. $\top^{I,w} = \Delta$, $C^{I,w} = C_i^{I,w}$, $R^{I,w} = R_j^{I,w}$, for $C = C_i$, $R = R_j$;
2. $x(\Diamond R)^{I,w} y$ iff $\exists v \rhd w \; xR^{I,v}y$;
3. $x(\Box R)^{I,w} y$ iff $\forall v \rhd w \; xR^{I,v}y$;
4. $(C \wedge D)^{I,w} = C^{I,w} \cap D^{I,w}$;
5. $(\neg C)^{I,w} = \Delta - C^{I,w}$;
6. $x \in (\Diamond C)^{I,w}$ iff $\exists v \rhd w \; x \in C^{I,v}$;
7. $x \in (\exists R.C)^{I,w}$ iff $\exists y \in C^{I,w} \; xR^{I,w}y$;
8. $w \models C = D$ iff $C^{I,w} = D^{I,w}$;
9. $w \models a : C$ iff $a^{I,w} \in C^{I,w}$;
10. $w \models aRb$ iff $a^{I,w}R^{I,w}b^{I,w}$;
11. $w \models \Diamond\varphi$ iff $\exists v \rhd w \; v \models \varphi$;

[2] W is a non-empty set of *worlds* and $\lhd$ a binary *accessibility relation* on W.

12. $w \models \varphi \wedge \psi$ iff $w \models \varphi$ and $w \models \psi$;

13. $w \models \neg\varphi$ iff $w \not\models \varphi$.

A formula φ is *satisfiable* if there is a model and a world w in it such that $w \models \varphi$.

Remark 5. The constructed language $\mathcal{ALC}_{\mathcal{M}}$ may be regarded as a fragment of modal predicate logic with constant domains and rigid designators (for definitions consult e.g. [Hughes and Cresswell, 1996]). To show this we define a translation † from $\mathcal{ALC}_{\mathcal{M}}$ into the language of modal predicate logic that extends the standard embedding of $\mathcal{ALC}$ into first-order logic [Schild, 1991]. Let us fix two distinct variables x and y. The translation $R^\dagger$ of a (possibly modalized) role R is defined inductively as follows:

$$\begin{array}{rcl} R_i^\dagger & = & R_i(x,y),\ R_i \text{ a role name}, \\ (\Box R)^\dagger & = & \Box R(x,y), \\ (\Diamond R)^\dagger & = & \Diamond R(x,y). \end{array}$$

The translation $C^\dagger$ of a concept C is a formula with at most one free variable defined in the following way:

$$\begin{array}{rcl} C_i^\dagger & = & P_i(x),\ C_i \text{ a concept name}, \\ \top^\dagger & = & \top, \\ (\exists R.C)^\dagger & = & \exists y(R^\dagger(x,y) \wedge C^\dagger\{y/x\}), \\ (C \wedge D)^\dagger & = & C^\dagger \wedge D^\dagger, \\ (\neg C)^\dagger & = & \neg C^\dagger, \\ (\Box C)^\dagger & = & \Box C^\dagger. \end{array}$$

Finally, the translation $\varphi^\dagger$ of a formula φ is a closed formula defined by:

$$\begin{array}{rcl} (C = D)^\dagger & = & \forall x\ (C^\dagger \leftrightarrow D^\dagger), \\ (a : C)^\dagger & = & C^\dagger\{a/x\}, \\ (aRb)^\dagger & = & R^\dagger(a,b), \\ (\Box\varphi)^\dagger & = & \Box\varphi^\dagger, \\ (\varphi \wedge \psi)^\dagger & = & \varphi^\dagger \wedge \psi^\dagger, \\ (\neg\varphi)^\dagger & = & \neg\varphi^\dagger. \end{array}$$

Here P_i and R in the right-hand side are unary and binary predicate symbols, respectively, a and b are constants. It is an easy exercise to prove that an $\mathcal{ALC}_{\mathcal{M}}$-formula φ is satisfiable iff $\varphi^\dagger$ is satisfiable in a Kripke model of modal predicate logic with constant domain and rigid designators. Observe that formulas of the form $\varphi^\dagger$ do not contain more than two individual variables. One might be tempted to conjecture that the satisfaction problem for the two-variable fragment of modal predicate logic is decidable, which would provide us with a quick decidability proof for the modal description language under consideration. Unfortunately, as was shown by [Gabbay and Shehtman, 1993], the two-variable fragment of modal predicate logic with constant domain turns out to be undecidable. This fact along with the decidability results to be obtained below may actually serve as one more strong evidence that the language constructed in this paper is an optimal compromise between the expressive power and complexity.

Before coming to the study of the satisfaction problem for $\mathcal{ALC}_{\mathcal{M}}$-formulas, let us make three simple observations. First, this problem for models with expanding domains can be reduced to the satisfaction problem for models with constant domains. To show this, we introduce a concept ex the intended meaning of which is to contain in each world precisely those objects that are assumed to exist (under the expanding domain assumption) in this world. By relativizing all concepts and formulas to the concept ex, one can simulate expanding domains using constant ones.

In the proof we shall be using the notion of the *modal depth* of a formula φ, $md(\varphi)$ in symbols. Informally, $md(\varphi)$ is the length of the longest chain of nested modal operators in φ (including those in the concepts and roles occurring in φ). Here is a more precise inductive definition:

$$\begin{array}{rcl}
md(R_i) & = & md(C_i) = 1, \\
md(C \wedge D) & = & \max\{md(C), md(D)\}, \\
md(\neg C) & = & md(C), \\
md(\Diamond C) & = & md(C) + 1, \\
md(\exists R.C) & = & \max\{md(R), md(C)\}, \\
md(\Diamond R) & = & md(\Box R) = md(R) + 1, \\
md(a : C) & = & md(C), \\
md(aRb) & = & md(R), \\
md(C = D) & = & \max\{md(C), md(D)\}, \\
md(\varphi \wedge \psi) & = & \max\{md(\varphi), md(\psi)\}, \\
md(\neg\varphi) & = & md(\varphi), \\
md(\Diamond\varphi) & = & md(\varphi) + 1.
\end{array}$$

Given a formula φ, we define inductively the formula $\Box^{\leq m}\varphi$ by taking $\Box^{\leq 0}\varphi = \varphi$ and $\Box^{\leq m+1}\varphi = \Box^{\leq m}\varphi \wedge \Box^{m+1}\varphi$.

Theorem 6. *If the satisfaction problem for $\mathcal{ALC}_{\mathcal{M}}$-formulas is decidable in models with constant domains, then it is decidable in models with expanding domains as well.*

Proof Assuming we are given a formula φ, let ex be a concept name having no occurrences in φ. By induction on the construction of a concept C define its relativization $C{\downarrow}\mathsf{ex}$:

$$\begin{array}{rcl}
C_i{\downarrow}\mathsf{ex} & = & C_i \wedge \mathsf{ex},\ C_i \text{ a concept name}, \\
(C \wedge D){\downarrow}\mathsf{ex} & = & (C{\downarrow}\mathsf{ex}) \wedge (D{\downarrow}\mathsf{ex}), \\
(\neg C){\downarrow}\mathsf{ex} & = & \mathsf{ex} \wedge \neg(C{\downarrow}\mathsf{ex}), \\
(\exists R.C){\downarrow}\mathsf{ex} & = & \mathsf{ex} \wedge \exists R.(C{\downarrow}\mathsf{ex}), \\
(\Diamond C){\downarrow}\mathsf{ex} & = & \mathsf{ex} \wedge \Diamond(C{\downarrow}\mathsf{ex}).
\end{array}$$

The relativization of φ is then defined inductively as follows:

$$\begin{aligned}
(aRb){\downarrow}\mathsf{ex} &= aRb \wedge (a : \mathsf{ex}) \wedge (b : \mathsf{ex}),\\
(a : C){\downarrow}\mathsf{ex} &= a : (C{\downarrow}\mathsf{ex}),\\
(C = D){\downarrow}\mathsf{ex} &= ((C{\downarrow}\mathsf{ex}) = (D{\downarrow}\mathsf{ex})),\\
(\neg\varphi){\downarrow}\mathsf{ex} &= \neg(\varphi{\downarrow}\mathsf{ex}),\\
(\varphi \wedge \psi){\downarrow}\mathsf{ex} &= (\varphi{\downarrow}\mathsf{ex}) \wedge (\psi{\downarrow}\mathsf{ex}),\\
(\Diamond\varphi){\downarrow}\mathsf{ex} &= \Diamond(\varphi{\downarrow}\mathsf{ex}).
\end{aligned}$$

Suppose now that $\mathfrak{F} = \langle W, \lhd \rangle$ is a frame and $m = md(\varphi)$. Then φ is satisfied in a model based on $\mathfrak{F}$ and having expanding domains iff the formula

$$\varphi' = \varphi{\downarrow}\mathsf{ex} \wedge \neg(\mathsf{ex} = \bot) \wedge \Box^{\leq m}((\mathsf{ex} \to \Box^{\leq 1}\mathsf{ex}) = \top \wedge \bigwedge_{a \in ob\varphi} a : \mathsf{ex})$$

is satisfied in a model based on $\mathfrak{F}$ and having constant domains. Indeed, assume that φ is satisfied in a model $\mathfrak{M} = \langle \mathfrak{F}, I \rangle$ with expanding domains and that

$$I(w) = \left\langle \Delta^{I,w}, R_0^{I,w}, \ldots, C_0^{I,w}, \ldots, a_0^{I,w}, \ldots \right\rangle,$$

for every $w \in W$ (so that $\Delta^{I,u} \subset \Delta^{I,v}$ whenever $u \lhd v$). Then we construct a model $\mathfrak{N} = \langle \mathfrak{F}, J \rangle$ with constant domains by taking

$$J(w) = \left\langle \bigcup_{w \in W} \Delta^{I,w}, R_0^{I,w}, \ldots, C_0^{I,w}, \ldots, \mathsf{ex}^{J,w}, a_0^{I,w}, \ldots \right\rangle,$$

where $\mathsf{ex}^{J,w} = \Delta^{I,w}$. It is readily checked by induction that for any $\psi \in sub\varphi$ and any $w \in W$, $(\mathfrak{M}, w) \models \psi$ iff $(\mathfrak{N}, w) \models \psi{\downarrow}\mathsf{ex}$. It follows that φ' is satisfied in $\mathfrak{N}$.

Conversely, suppose φ' is satisfied in a world v in a model $\mathfrak{N} = \langle \mathfrak{F}, J \rangle$ with constant domain and that

$$J(w) = \left\langle \Delta, R_0^{J,w}, \ldots, C_0^{J,w}, \ldots, \mathsf{ex}^{J,w}, a_0^{J,w}, \ldots \right\rangle,$$

for $w \in W$. Consider the model $\mathfrak{M} = \langle \mathfrak{F}, I \rangle$ in which

$$I(w) = \left\langle \mathsf{ex}^{J,w}, R_0^{I,w}, \ldots, C_0^{I,w}, \ldots, a_0^{J,w}, \ldots \right\rangle,$$

where $R_i^{I,w}$ and $C_i^{I,w}$ are the restrictions of $R_i^{J,w}$ and $C_i^{J,w}$ to $\mathsf{ex}^{J,w}$, respectively, for every w accessible from v in $\leq m$ steps, and $I(w) = J(w)$ for all the other worlds w in $\mathfrak{F}$. Since $(\mathfrak{N}, v) \models \neg(\mathsf{ex} = \bot)$, the domains of worlds in $\mathfrak{M}$ are not empty, and the conjunct $(\mathsf{ex} \to \Box^{\leq 1}\mathsf{ex}) = \top$ ensures that they are expanding. Now, using the fact that the truth-value of φ in v depends only on the worlds accessible from v in $\leq m$ steps, one can show by induction that for every $\psi \in sub\varphi$, $(\mathfrak{N}, v) \models \psi{\downarrow}\mathsf{ex}$ iff $(\mathfrak{M}, v) \models \psi$. ❑

Our second observation concerns the reasoning tasks that are regarded to be important in knowledge representation systems. These are *concept satisfiability*, *subsumption*, *entailment*, *instance checking*, and some others (see [Donini *et al.*, 1996] for a discussion of different forms of reasoning in description logic). However, all of these tasks are reducible to the satisfaction problem. For instance,

- *Subsumption*: clearly, C is subsumed by D in every model iff the formula $\neg(C \to D = \top)$ is not satisfiable.
- *Entailment*. Say that a finite set of formulas Γ *entails* φ if for every model $\mathfrak{M}$ and every world w in it, we have $(\mathfrak{M}, w) \models \varphi$ whenever $(\mathfrak{M}, w) \models \Gamma$. It should be clear that Γ entails φ iff $\bigwedge \Gamma \wedge \neg\varphi$ is not satisfiable.
- *Instance checking*: given a finite set of formulas Γ, a concept name a, and a concept C, to decide whether Γ entails $a : C$. This problem is reduced to the previous one.

Thus, we can focus our attention only on the satisfaction problem.

It is known in modal logic (see e.g. [Chagrov and Zakharyaschev, 1997]) that every satisfiable purely modal formula φ can be satisfied in a finite intransitive tree of depth $\leq md(\varphi)$. We remind the reader that a frame $\mathfrak{G} = \langle W, \lhd \rangle$ is called a *tree* if (i) $\mathfrak{G}$ is *rooted*, i.e., there is $w_0 \in W$ (a *root* of $\mathfrak{G}$) such that $w_0 \lhd^* w$ for every $w \in W$, where $\lhd^*$ is the transitive and reflexive closure of $\lhd$, and (ii) for every $w \in W$, the set $\{v \in W : v \lhd^* w\}$ is finite and linearly ordered by $\lhd^*$. The *depth* of a tree is the length of its longest branch. The *depth* $d(w)$ of a world w in it is the depth of the subtree generated by w. And by the *co-depth* of w we mean the number of worlds in the chain $\{v \in W : v \lhd^* w\}$. A tree $\mathfrak{G} = \langle W, \lhd \rangle$ is *intransitive* if every world v in $\mathfrak{G}$, save its root, has precisely one predecessor, i.e., $|\{u \in W : u \lhd v\}| = 1$, and the root w_0 is *irreflexive*, i.e., $\neg w_0 \lhd w_0$ (in fact, all worlds in an intransitive frame are irreflexive).

We formulate our third observation in the form of a lemma; it can be easily proved using the standard technique of modal logic.

Lemma 7. *Every satisfiable $\mathcal{ALC}_{\mathcal{M}}$-formula is satisfied in a model based on an intransitive tree of depth $\leq md(\varphi)$ (but possibly with infinitely many branches).*

3 Quasimodels

Let us fix an arbitrary $\mathcal{ALC}_{\mathcal{M}}$-formula φ. In general, $\mathcal{ALC}_{\mathcal{M}}$-models are rather complex structures with rich interactions between worlds, concepts and roles. That is why standard methods of establishing decidability (say, filtration) do not go through for them. Our idea is to factorize the models modulo φ in such a way that the resulting structures—we will call them *quasimodels*—can be constructed as mosaics from a finite number of relatively small finite pattern pieces called *blocks*.

To introduce the notion of a quasimodel, we require a number of auxiliary definitions.

Definition 8 (types). A *concept type* for φ is a subset t of $con\varphi$ such that

- $C \wedge D \in t$ iff $C, D \in t$, for every $C \wedge D \in con\varphi$;
- $\neg C \in t$ iff $C \notin t$, for every $\neg C \in con\varphi$.

A *named concept type* is the pair $t_a = \langle t, a \rangle$ in which t is a concept type and $a \in ob\varphi$. A *formula type* for φ is a subset Ξ of $sub\varphi$ such that

- $\psi \wedge \chi \in \Xi$ iff $\psi, \chi \in \Xi$, for every $\psi \wedge \chi \in sub\varphi$;

- $\neg\psi \in \Xi$ iff $\psi \notin \Xi$, for every $\neg\psi \in sub\varphi$.

A *named formula type* is the pair $\Xi_a = \langle \Xi, a\rangle$ in which Ξ is a formula type and $a \in ob\varphi$. Finally, by a *type* for φ we will mean the pair $\tau = (t, \Xi)$, t a concept type and Ξ a formula type for φ; $\tau_a = (t_a, \Xi_a)$ is a *named type* for φ.

To simplify notation we will write $C \in \tau$ and $\psi \in \tau$ whenever $\tau = (t, \Xi)$, $C \in t$ and $\psi \in \Xi$ (in case of named types $C \in \tau_a$ and $\psi \in \tau_a$ mean that $\tau_a = \langle t_a, \Xi_a\rangle$, $t_a = \langle t, a\rangle$, $\Xi_a = \langle \Xi, a\rangle$, and $C \in t$, $\psi \in \Xi$). This should not cause any ambiguity. Two types $\tau_1 = (t_1, \Xi_1)$ and $\tau_2 = (t_2, \Xi_2)$ are said to be *formula-equivalent* if $\Xi_1 = \Xi_2$.

Definition 9 (type tree). By a *type tree* for φ we mean a structure of the form $\mathfrak{T} = \langle T, <\rangle$, where T is a finite set of labelled types for φ (so that one type may have many occurrences in T with different labels) and $<$ an intransitive tree order on T such that

(a) for all $\tau \in T$ and $\Diamond C \in con\varphi$, we have $\Diamond C \in \tau$ iff $\exists \tau' > \tau\ C \in \tau'$;

(b) for all $\tau \in T$ and $\Diamond\psi \in sub\varphi$, we have $\Diamond\psi \in \tau$ iff $\exists \tau' > \tau\ \psi \in \tau'$;

(c) $\mathfrak{T}$ is of depth $\leq md(\varphi)$;

(d) if $\tau < \tau'$, $\tau < \tau''$ and $\tau' \neq \tau''$ then the subtrees of $\mathfrak{T}$ generated by τ' and τ'' are not isomorphic.[3]

Type trees are intended to represent the "behaviour" of a single object in standard models modulo φ.

It should be clear that there exist at most $N_d(\varphi)$ pairwise non-isomorphic type trees of depth d, where

$$N_1(\varphi) = 2^{|con\varphi|} \cdot 2^{|sub\varphi|}, \quad N_{n+1}(\varphi) = 2^{|con\varphi|} \cdot 2^{|sub\varphi|} \cdot 2^{N_n(\varphi)}.$$

So the number of types in each type tree for φ does not exceed

$$\sharp(\varphi) = 1 + \sum_{i=1}^{md(\varphi)-1} \prod_{0<j\leq i} N_{md(\varphi)-j}(\varphi) \leq (N_{md(\varphi)}(\varphi))^{md(\varphi)}.$$

Definition 10 (type forest). Let Δ be a non-empty set. A *type forest of depth m* over Δ is a set

$$\mathfrak{F} = \{\mathfrak{T}_x : x \in \Delta\},$$

where all $\mathfrak{T}_x = \langle T_x, <_x\rangle$ are type trees for φ of the same depth m, and $\mathfrak{T}_a$, for every $a \in \Delta \cap ob\varphi$, consists of only named types of the form τ_a.

To represent worlds in models with their inner complex structure we require the following definition.

Definition 11 (run). A *run of co-depth d* through a type forest $\mathfrak{F} = \{\mathfrak{T}_x : x \in \Delta\}$ over Δ is the pair of the form

$$r = \langle \Delta_r, \{R_r : R \in rol\varphi\}\rangle$$

in which Δ_r is a set containing precisely one type $r(x) \in T_x$ of co-depth d for every $x \in \Delta$ (so that $\Delta_r = \{r(x) : x \in \Delta\}$) and $R_r \subseteq \Delta_r \times \Delta_r$ such that:

[3]Two type trees $\mathfrak{T}_1 = \langle T_1, <_1\rangle$ and $\mathfrak{T}_2 = \langle T_2, <_2\rangle$ are isomorphic if there is a bijection $f : T_1 \to T_2$ such that $f(\tau) = \tau$ and $\tau <_1 \tau'$ iff $f(\tau) <_2 f(\tau')$. Actually, we can require these subtrees to be not bisimilar.

(e) all types in Δ_r are formula-equivalent to each other;

(f) $\exists R.C \in r(x)$ iff $\exists y \in \Delta\ (r(x) R_r r(y)\ \&\ C \in r(y))$, for every $\exists R.C \in con\varphi$;

(g) $C = D \in r(x)$ iff $\forall y \in \Delta\ (C \in r(y) \Leftrightarrow D \in r(y))$, for every $C = D \in sub\varphi$;

(h) $a : C \in r(x)$ iff $C \in r(a)$, for every $a : C \in sub\varphi$, provided that $a \in \Delta$;

(i) $aRb \in r(x)$ iff $r(a) R_r r(b)$, for every $aRb \in sub\varphi$, provided that $a, b \in \Delta$.

If only the ($\Leftarrow$)-part of (f) holds, we call r a *weak run* of co-depth d. And if a weak run r satisfies (f) for some particular x in Δ, then r is called a *weak x-saturated run* of co-depth d. Instead of $\psi \in r(x)$ we will write $r \models \psi$.

Definition 12 (quasimodel). A triple $\mathfrak{m} = \langle \mathfrak{F}, \mathfrak{R}, \lhd \rangle$ is a *quasimodel* for φ if $\mathfrak{F}$ is a type forest of depth $m \leq md(\varphi)$ for φ over some $\Delta \supseteq ob\varphi$, $\mathfrak{R}$ a set of runs through $\mathfrak{F}$ and $\lhd$ is an intransitive tree order on $\mathfrak{R}$ such that the following conditions hold:

(j) for every $d \leq m$, the set $\mathfrak{R}^d$ of runs of co-depth d in $\mathfrak{R}$ is non-empty;

(k) for any $r, r' \in \mathfrak{R}$, if $r \lhd r'$ then $r(x) <_x r'(x)$ for all $x \in \Delta$;

(l) for all $r \in \mathfrak{R}^d$, $x \in \Delta$, and $\tau \in T_x$, if $r(x) <_x \tau$ then there is $r' \in \mathfrak{R}^{d+1}$ such that $r'(x) = \tau$ and $r \lhd r'$;

(m) for all $x, y \in \Delta$, $d \leq m$, $\Diamond R \in rol\varphi$, and $r \in \mathfrak{R}^d$, we have $r(x)(\Diamond R)_r r(y)$ iff $\exists r' \rhd r\ r'(x) R_{r'} r'(y)$;

(n) for all $x, y \in \Delta$, $d \leq m$, $\Box R \in rol\varphi$, and $r \in \mathfrak{R}^d$, we have $r(x)(\Box R)_r r(y)$ iff $\forall r' \rhd r\ r'(x) R_{r'} r'(y)$.

We say $\mathfrak{m}$ *satisfies* φ if $r \models \varphi$ for some $r \in \mathfrak{R}$.

Remark 13. It follows from (l) that for all $r \in \mathfrak{R}$, $x, y \in \Delta$, the types $r(x)$ and $r(y)$ are of the same depth in, respectively, $\mathfrak{T}_x$ and $\mathfrak{T}_y$.

Theorem 14. *A formula φ is satisfied in an $\mathcal{ALC}_{\mathcal{M}}$-model iff φ is satisfied in some quasimodel for φ.*

Proof ($\Leftarrow$) Suppose φ is satisfied in a quasimodel $\mathfrak{m} = \langle \mathfrak{F}, \mathfrak{R}, \lhd \rangle$ over domain Δ. Construct a standard model $\mathfrak{M} = \langle \mathfrak{G}, I \rangle$ based on a frame $\mathfrak{G} = \langle W, \lhd \rangle$ by taking:

$$W = \bigcup \mathfrak{R}, \quad I(r) = \left\langle \Delta, R_0^{I,r}, \dots, C_0^{I,r}, \dots, a_0^{I,r}, \dots \right\rangle, \text{ for } r \in \mathfrak{R},$$
$$x R_i^{I,r} y \text{ iff } r(x)(R_i)_r r(y), \quad x \in C_i^{I,r} \text{ iff } C_i \in r(x), \quad a_i^{I,r} = a_i.$$

By a straightforward induction on the construction of concepts, roles and formulas one can check that for all $C \in con\varphi$, $R \in rol\varphi$, $\psi \in sub\varphi$, $x, y \in \Delta$, we have:

$$x R^{I,r} y \text{ iff } r(x) R_r r(y),$$
$$x \in C^{I,r} \text{ iff } C \in r(x),$$
$$(\mathfrak{M}, r) \models \psi \text{ iff } (\mathfrak{m}, r) \models \psi.$$

($\Rightarrow$) Suppose now that $\mathfrak{M} = \langle \mathfrak{G}, I \rangle$ is a model with domain $\Delta \supseteq ob\varphi$ satisfying φ. In view of Lemma 7, we may assume that $\mathfrak{G} = \langle W, < \rangle$ is an intransitive tree of depth $\leq md(\varphi)$. For every pair $x \in \Delta$, $w \in W$, let

$$t(x,w) = \{C \in con\varphi : x \in C^{I,w}\},$$
$$\Xi(w) = \{\psi \in sub\varphi : w \models \psi\},$$
$$\tau(x,w) = (t(x,w), \Xi(w)).$$

Clearly, $\tau(x,w)$ is a type for φ; $\tau(a,w)$ is regarded to be a type named by a, for $a \in ob\varphi$. Fix some $x \in \Delta$ and define an equivalence relation $u \sim_x v$ for $u, v \in W$ by induction on depth:

- if $d(u) = d(v) = 1$ then $u \sim_x v$ iff $\tau(x,u) = \tau(x,v)$;
- if $d(u) = d(v) = d + 1$ then $u \sim_x v$ iff
  - $\tau(x,u) = \tau(x,v)$,
  - $\forall w > u \exists w' > v \; w \sim_x w'$,
  - $\forall w' > v \exists w > u \; w \sim_x w'$.

Let $[u]_x$ denote the $\sim_x$-equivalence class generated by u. We construct a type tree $\mathfrak{T}_x = \langle T_x, <_x \rangle$ by induction. First we put $\tau(x, w_0)$ in T_x, w_0 the root of $\mathfrak{G}$. Suppose now that we have already put the type $\tau(x,u)$ in T_x and are looking for its successors. Then we take all non-empty sets of the form $[v]_x \cap \{w : w > u\}$ and select one representative in each of them; it will be called the *trace* of all the worlds in the set. All the selected traces are put into T_x as $<_x$-successors of $\tau(x,u)$. Clearly, $\mathfrak{T}_x$ is a type tree, and so $\mathfrak{F} = \{\mathfrak{T}_x : x \in \Delta\}$ is a type forest.

To construct runs through $\mathfrak{F}$, we require the following inductive extension of the notion of a trace. Let u be a trace of u' in $\mathfrak{T}_x$ and $v' > u'$. Then we pick one $v > u$ such that $v \in T_x$, $v \sim_x v'$ and declare it to be the *trace* of v' in $\mathfrak{T}_x$.

Now, with every world $v \in W$ we associate the pair

$$r_v = \langle \Delta_{r_v}, \{R_{r_v} : R \in rol\varphi\} \rangle$$

by taking $r_v(x) = \tau(x, u_x)$, u_x the trace of v in $\mathfrak{T}_x$, and $r_v(x) R_{r_v} r_v(y)$ iff $x R^{I,v} y$. It should be clear that r_v is a run through $\mathfrak{G}$.

Finally, we put $\mathfrak{R} = \{r_w : w \in W\}$ and $r_u \lhd r_v$ iff $u < v$. It is a matter of routine to check that $\mathfrak{m} = \langle \mathfrak{F}, \mathfrak{R}, \lhd \rangle$ is a quasimodel satisfying φ. ❑

Suppose $\mathfrak{m} = \langle \mathfrak{F}, \mathfrak{R}, \lhd \rangle$ is a quasimodel (for φ) over Δ, $x \in \Delta$, $R \in rol\varphi$ and $R = \mathsf{M} R_i$ for some (possibly empty) string M of $\Diamond$ and $\Box$, R_i a role name. Consider the type tree $\mathfrak{T}_x = \langle T_x, <_x \rangle$ as a usual Kripke frame. If we have $(\mathfrak{T}_x, r(x)) \models \mathsf{M}\bot$, for $r \in \mathfrak{R}$, then let us say that R is *r-universal*. This name is explained by the fact that if R is r-universal then $R_r = \Delta_r \times \Delta_r$, which can be easily established by induction on the length of the string M.

We also say that objects $y, z \in \Delta$ are *twins* relative to $x \in \Delta$ if (i) $\mathfrak{T}_y$ and $\mathfrak{T}_z$ are isomorphic, and (ii) for all $r \in \mathfrak{R}$ and $R \in rol\varphi$, we have $r(y) = r(z)$ and $r(x) R_r r(y)$ iff $r(x) R_r r(z)$.

Lemma 15. *Every satisfiable $\mathcal{ALC}_{\mathcal{M}}$-formula φ is satisfied in a quasimodel $\mathfrak{m}^* = \langle \mathfrak{F}^*, \mathfrak{R}^*, \lhd^* \rangle$ for φ over some Δ^* such that the following conditions hold:*

- *for any distinct $x, y \in \Delta^*$, the object y has infinitely many twins relative to x;*
- $\{\langle x,y\rangle : x,y \notin ob\varphi,\ \exists R \exists r \in \mathfrak{R}^*(r(x)R_r r(y)$ *& R is not r-universal*$)\}$ *is an intransitive forest order on the set $\Delta - ob\varphi$.*

Proof Suppose φ is satisfied in a quasimodel $\mathfrak{m} = \langle \mathfrak{F}, \mathfrak{R}, \lhd\rangle$ for φ over Δ. For each $x \in \Delta$ we take an infinite set X_x containing x so that $X_y \cap X_z = \emptyset$ whenever $y \neq z$. For every $y \in X_x$ let $\mathfrak{T}_y$ be an isomorphic copy of $\mathfrak{T}_x$, and let $\Delta' = \bigcup\{X_x : x \in \Delta\}$. Thus we have got a type forest $\mathfrak{F}'$ over Δ'. Now we extend every run $r \in \mathfrak{R}$ to a run r' through $\mathfrak{F}'$ simply by taking $r'(y) = r(x)$ for all $y \in X_x$, and $r'(y')R_{r'}r'(z')$ iff $r(y)R_r r(z)$, for all $y' \in X_y$, $z' \in X_z$. The resulting set of runs is denoted by $\mathfrak{R}'$; we put $r_1' \lhd' r_2'$ iff $r_1 \lhd r_2$, for all $r_1, r_2 \in \mathfrak{R}$. It is readily seen that $\mathfrak{m}' = \langle \mathfrak{F}', \mathfrak{R}', \lhd'\rangle$ is a quasimodel satisfying φ and the former condition of the lemma. To satisfy the latter, we apply to $\mathfrak{m}'$ the unraveling technique.

Denote by Δ^* the set of all finite n-tuples $\langle x_1, \dots, x_n\rangle$ of objects in Δ', $n < \omega$, such that $x_i \notin ob\varphi$ for $i \neq 1$, and let $\mathfrak{T}_{\langle x_1,\dots,x_n\rangle} = \mathfrak{T}_{x_n}$, which yields us a type forest $\mathfrak{F}^*$ over Δ^*. Given a run $r \in \mathfrak{R}'$, we construct r^* by taking

$$r^*(\langle x_1, \dots, x_n\rangle) = r(x_n)$$

and, for every $R \in rol\varphi$, $r^*(\langle x_1, \dots, x_n\rangle)R_{r^*}r^*(\langle y_1, \dots, y_m\rangle)$ iff either R is r-universal or $\langle x_1, \dots, x_n\rangle = \langle y_1, \dots, y_{m-1}\rangle$ and $r(x_n)R_r r(y_m)$. It is not hard to check that r^* is a run through $\mathfrak{F}^*$. Finally, we put $r_1^* \lhd^* r_2^*$ iff $r_1 \lhd' r_2$, for all $r_1, r_2 \in \mathfrak{R}'$. The structure $\mathfrak{m}^* = \langle \mathfrak{F}^*, \mathfrak{R}^*, \lhd^*\rangle$ is then a quasimodel satisfying φ and both conditions of the lemma as well. ❑

4 Constructing mosaics

We are in a position now to show that a formula φ is satisfiable iff one can construct a (possibly infinite) quasimodel satisfying φ out of a finite set of finite pattern blocks.

Definition 16 (block). Let $\mathfrak{F}$ be a type forest for φ of depth m over a finite Δ which is disjoint from $ob\varphi$, $x \in \Delta$, $\mathfrak{R}$ a set of weak x-saturated runs through $\mathfrak{F}$ such that

$$\{\langle x,y\rangle : \exists R \in rol\varphi \exists r \in \mathfrak{R}\ (r(x)R_r r(y) \ \&\ R \text{ is not } r\text{-universal}\}$$

is an intransitive tree order on Δ with root x, and let $\lhd$ be an intransitive tree order on $\mathfrak{R}$. We say $\mathfrak{b} = \langle \mathfrak{F}, \mathfrak{R}, \lhd\rangle$ is a $\mathfrak{T}_x$-*block* for φ if it satisfies conditions (j)–(n). The tree $\mathfrak{T}_x$ is called then the *root* of $\mathfrak{b}$.

Definition 17 (kernel block). A *kernel block* over $ob\varphi \neq \emptyset$ is a structure of the form $\mathfrak{b}_o = \langle \mathfrak{F}_o, \mathfrak{R}_o, \lhd_o\rangle$ in which $\mathfrak{F}_o$ is a type forest over $ob\varphi$ of depth m (it contains only type trees named by elements in $ob\varphi$), $\mathfrak{R}_o$ a set of weak runs through $\mathfrak{F}_o$ and $\lhd_o$ an intransitive tree order on $\mathfrak{R}_o$ satisfying (j)–(n).

Definition 18 (satisfying set). A non-empty set of blocks $\mathcal{S}$ for φ is called a *satisfying set* for φ if

(o) $\mathcal{S}$ contains one kernel block for φ whenever $ob\varphi \neq \emptyset$;

(p) in every block $\langle \mathfrak{F}, \mathfrak{R}, \lhd \rangle \in \mathcal{S}$ there is $r \in \mathfrak{R}$ such that $r \models \varphi$;

(q) for every $\langle \mathfrak{F}, \mathfrak{R}, \lhd \rangle$ in $\mathcal{S}$ and every $\mathfrak{T}_x \in \mathfrak{F}$, there is precisely one $\mathfrak{T}_x$-block in $\mathcal{S}$;

(r) if $ob\varphi \neq \emptyset$ then, for every $\mathfrak{T}_a \in \mathfrak{F}_o$, there is precisely one $\mathfrak{T}_x$-block in $\mathcal{S}$ such that $\mathfrak{T}_a$ is isomorphic to $\mathfrak{T}_x$ (but types in $\mathfrak{T}_x$ are not named).

Theorem 19. *An $\mathcal{ALC}_{\mathcal{M}}$-formula φ is satisfiable iff there is a satisfying set for φ the domain of each (non-kernel) block in which contains at most*

$$\sharp(\varphi) \cdot |con\varphi| \cdot (md(\varphi) + 1) + 1$$

objects.

Proof ($\Rightarrow$) Suppose φ is satisfiable. Then there is a quasimodel $\langle \mathfrak{F}, \mathfrak{R}, \lhd \rangle$ satisfying φ and meeting the conditions of Lemma 15. Let $\mathfrak{R} = \bigcup_{i=1}^{m} \mathfrak{R}^i$ for some $m \leq md\varphi$. With every type tree $\mathfrak{T}_x = \langle T_x, <_x \rangle$, for $x \in \Delta - ob\varphi$, we are going to associate a $\mathfrak{T}_x$-block $\mathfrak{b}_x = \langle \mathfrak{F}_x, \mathfrak{R}_x, \lhd_x \rangle$.

We begin the construction by defining auxiliary sets of runs $\mathfrak{Q}^i$, for $i \leq m$. Let $\mathfrak{Q}^1$ consist of the unique run in $\mathfrak{R}^1$. Now suppose $\mathfrak{Q}^k$ has been constructed. Then for every run $r \in \mathfrak{Q}^k$ and every $\tau >_x r(x)$, we select (in accordance with (l)) one run $r' \in \mathfrak{R}^{k+1}$ such that $r \lhd r'$, $r'(x) = \tau$ and add it to $\mathfrak{Q}^{k+1}$. Thus

$$\left| \bigcup_{i=1}^{m} \mathfrak{Q}^i \right| \leq \sharp(\varphi).$$

To construct $\mathfrak{b}_x$, we first define its domain Δ^x. For every $r \in \mathfrak{Q}^k$, $k \leq m$, and every $R \in rol\varphi$ with $\exists R.C \in r(x)$ we select (by Lemma 15 and (f)) $n = m + 1$ twins $y_1, \dots, y_n \in \Delta$ relative to x such that $C \in r(y_i)$ and $r(x) R_r r(y_i)$, $i \in [1, n]$.[4] Without loss of generality we may assume that, for every pair $r \in \mathfrak{Q}^k$ and $R \in rol\varphi$ with $\exists R.C \in r(x)$, we choose a new n-tuple of twins. All these objects together with x form Δ^x. And then we take $\mathfrak{F}_x = \{\mathfrak{T}_z : z \in \Delta^x\}$. So

$$|\Delta^x| = |\mathfrak{F}_x| \leq \sharp(\varphi) \cdot |con\varphi| \cdot (md(\varphi) + 1) + 1.$$

According to Lemma 15 we may assume that for every run $r \in \mathfrak{R}$ and every $R \in rol(\varphi)$, we have:

- if $(\mathfrak{T}_x, r(x)) \not\models \mathsf{M}\bot$ then, for all $y, z \in \Delta^x$, $r(z) R_r r(y)$ implies $z = x$ and $x \neq y$;
- if $(\mathfrak{T}_x, r(x)) \models \mathsf{M}\bot$ then $R_r = \Delta_r \times \Delta_r$.

Given runs $r, r' \in \mathfrak{R}^d$ such that $r(x) = r'(x)$ and given an object $y \in \Delta^x$ different from x, construct a run $s = r(y) + r'$ by taking, for every $z \in \Delta^x$,

$$s(z) = \begin{cases} r(y) & \text{if } z = y, \\ r'(z) & \text{otherwise} \end{cases}$$

and by defining $R_s \subseteq \Delta_s^x \times \Delta_s^x$, for every $R \in rol\varphi$, as follows:

[4]Here and below $[i, j] = \{i, i + 1, \dots, j\}$, $(i, j] = \{i + 1, \dots, j\}$.

- $R_s = \Delta^x_s \times \Delta^x_s$ whenever R is r-universal (and so r'-universal as well),
- $s(x)R_s s(z)$ iff $z = y$ and $r(x)R_r r(z)$, or $z \neq y$ and $r'(x)R_{r'}r'(z)$, otherwise.

It should be clear that

$$s = \langle \Delta^x_s, \{R_s : R \in rol\varphi\}\rangle$$

is a weak run of co-depth d through $\mathfrak{F}_x$ which behaves like r on the pair x, y and like r' on other pairs.

Now let us construct sets $\mathfrak{Q}^1_x, \dots, \mathfrak{Q}^m_x$. To begin with, we put $\mathfrak{Q}^i \subseteq \mathfrak{Q}^i_x$, for $i \in [1, m]$, and $\mathfrak{Q}^1_x = \mathfrak{Q}^1$.

Suppose now that $d \in (1, m]$, $y_1, \dots, y_l$ are distinct objects in Δ^x for some $l \in [0, d]$, $x \neq y_i$, $r_1, \dots, r_l \in \mathfrak{R}^d$, $r \in \mathfrak{Q}^d$, and $r_i(x) = r(x)$, for all $i \in [1, l]$. Then we form the weak run

$$s = r_1(y_1) + (r_2(y_2) + (\dots + (r_l(y_l) + r)\dots)) \tag{1}$$

and add it to $\mathfrak{Q}^d_x$. Since $r \in \mathfrak{Q}^d$ and $l < n$, the restriction of s to Δ^x is an x-saturated weak run of co-depth d through $\mathfrak{F}_x$.

Let $\mathfrak{R}^i_x$, $i \in [1, m]$, consist of the restrictions of runs in $\mathfrak{Q}^i_x$ to Δ^x, and let

$$\mathfrak{R}_x = \bigcup_{i=1}^{m} \mathfrak{R}^i_x.$$

Finally, given two weak x-saturated runs s and s' of, respectively, the form (1) and

$$s' = r'_1(y'_1) + (r'_2(y'_2) + (\dots + (r'_k(y'_k) + r')\dots)) \tag{2}$$

such that $s \in \mathfrak{R}^d_x$, $s' \in \mathfrak{R}^{d+1}_x$, we put $s \lhd_x s'$ iff $l \leq k$, $y_i = y'_i$ and $r_i \lhd r'_i$ for $i \in [1, l]$, $r \lhd r'_j$ for $j \in [l+1, k]$, and $r \lhd r'$.

Let us prove that the constructed triple $\mathfrak{b}_x = \langle \mathfrak{F}_x, \mathfrak{R}_x, \lhd_x\rangle$ is a $\mathfrak{T}_x$-block. Clearly, it satisfies (j) and (k).

(l) Suppose s of the form (1) is in $\mathfrak{R}^d_x$, $z \in \Delta^x$, $\tau \in T_z$ and $s(z) <_z \tau$.

Case 1: $z = y_i$ for some $i \in [1, l]$. By (l) we have $r'_i \in \mathfrak{R}^{d+1}$ such that $r_i \lhd r'_i$ and $r'_i(y_i) = \tau$. For $j \neq i$ we take, again by (l), $r'_j \rhd r_j$ in $\mathfrak{R}^{d+1}$ such that $r'_j(x) = r'_i(x)$. And by the definition of $\mathfrak{Q}^{d+1}$, in this set we have some $r' \rhd r$ with $r'(x) = r'_i(x)$. Then (the restriction of)

$$s' = r'_1(y_1) + (r'_2(y_2) + (\dots + (r'_l(y_l) + r')\dots))$$

is in $\mathfrak{R}^{d+1}_x$, $s'(z) = r'_i(y_i) = \tau$ and $s \lhd_x s'$.

Case 2: $z \neq y_i$ for any $i \in [1, l]$. By (l) we select a run $r'_{l+1} \rhd r$ in $\mathfrak{R}^{d+1}$ such that $r'_{l+1}(z) = \tau$. Then we choose runs $r'_i \rhd r_i$ in $\mathfrak{R}^{d+1}$, $i \in [1, l]$, with $r'_i(x) = r'_{l+1}(x)$, and finally, we take in $\mathfrak{Q}^{d+1}$ a run $r' \rhd r$ with $r'(x) = r'_{l+1}(x)$. Then

$$s' = r'_1(y_1) + (r'_2(y_2) + (\dots + (r'_{l+1}(y_{l+1}) + r')\dots))$$

is in $\mathfrak{R}^{d+1}_x$, $s'(z) = r'_{l+1}(z) = \tau$ and $s \lhd_x s'$.

(m) Suppose $s \in \mathfrak{R}^d_x$ is of the form (1) and $s(x)(\Diamond R)_s s(z)$, for some $z \in \Delta^x$. Let us assume first that $\Diamond R$ is not s-universal.

Case 1: $z = y_i$ for some $i \in [1,l]$. By (m) we have a run $r'_i \rhd r_i$ in $\mathfrak{R}^{d+1}$ such that $r'_i(x) R_{r'_i} r'_i(y_i)$. For $j \neq i$ we select by (l) a run $r'_j \rhd r_j$ in $\mathfrak{R}^{d+1}$ such that $r'_j(x) = r'_i(x)$. And we also take in $\mathfrak{Q}^{d+1}$ a run $r' \rhd r$ with $r'(x) = r'_i(x)$. Then

$$s' = r'_1(y_1) + (r'_2(y_2) + (\cdots + (r'_l(y_l) + r') \ldots))$$

is in $\mathfrak{R}^{d+1}_x$, $s \lhd_x s'$ and $s'(x) R_{s'} s'(z)$.

Case 2: $z \neq y_i$ for any $i \in [1,l]$. By (m) we have a run $r'_{l+1} \rhd r$ in $\mathfrak{R}^{d+1}$ such that $r'_{l+1}(x) R_{r'_{l+1}} r'_{l+1}(z)$. Then we choose runs $r'_i \rhd r_i$ in $\mathfrak{R}^{d+1}$, $i \in [1,l]$ with $r'_i(x) = r'_{l+1}(x)$, and also we take in $\mathfrak{Q}^{d+1}$ a run $r' \rhd r$ with $r'(x) = r'_{l+1}(x)$. And then

$$s' = r'_1(y_1) + (r'_2(y_2) + (\cdots + (r'_{l+1}(y_{l+1}) + r') \ldots))$$

is in $\mathfrak{R}^{d+1}_x$, $s \lhd_x s'$ and $s'(x) R_{s'} s'(z)$.

Assume now that $\Diamond R$ is s-universal and $s(y)(\Diamond R)_s s(z)$, $y, z \in \Delta^x$, s being of the form (1). Then $\Diamond R$ is r_i- and r-universal too. Suppose $R = \mathsf{M} R_j$, R_j a role name. Then there exists $\tau \in T_x$ such that $r(x) <_x \tau$ and $(\mathfrak{T}_x, \tau) \models \mathsf{M}\bot$. Take $r' \rhd r$ in $\mathfrak{Q}^{d+1}$ such that $r'(x) = r(x)$ and also $r'_i \rhd r_i$ in $\mathfrak{R}^{d+1}$ with $r'_i(x) = r_i(x)$. Then

$$s' = r'_1(y_1) + (r'_2(y_2) + (\cdots + (r'_l(y_l) + r') \ldots))$$

is in $\mathfrak{R}^{d+1}_x$, $s \lhd_x s'$ and R is s-universal.

Thus we have proved the ($\Rightarrow$)-part of (m). To show the converse, suppose s and s' are of the form (1) and (2), respectively, $s \lhd_x s'$ and $s'(x) R_{s'} s'(z)$, for some $\Diamond R \in rol\varphi$ that is not s-universal.

If $z = y_i$, for some $i \in [1,l]$, then we have $r_i \lhd r'_i$ and so, in view of (m), $r_i(x)(\Diamond R)_{r_i} r_i(z)$, from which $s(x)(\Diamond R)_s s(z)$. Let $z = y_j$ for $j \in [l+1,k]$. Then $r \lhd r_j$, $r(x)(\Diamond R)_r r(z)$ and so $s(x)(\Diamond R)_s s(z)$. Finally, if $z \neq y_i$ for any $i \in [1,k]$ then $r \lhd r'$, and we again have $r(x)(\Diamond R)_r r(z)$, from which $s(x)(\Diamond R)_s s(z)$.

The case of s-universal $\Diamond R$ is trivial.

(n) is treated similarly to (m).

It remains to construct a kernel block $\mathfrak{b}_o = \langle \mathfrak{F}_o, \mathfrak{R}_o, \lhd_o \rangle$ for φ provided that $ob\varphi \neq \emptyset$. Let $\mathfrak{F}_o = \{\mathfrak{T}_a : a \in ob\varphi\}$, let $\mathfrak{R}_o$ contain the restrictions of runs in $\mathfrak{R}$ to $\mathfrak{F}_o$ (now they become weak runs), and let $r_1 \restriction \mathfrak{F}_o \lhd_o r_2 \restriction \mathfrak{F}_o$ iff $r_1 \lhd r_2$, for all $r_1, r_2 \in \mathfrak{R}$. It is readily seen that $\mathfrak{b}_o$ is a kernel block for φ.

($\Rightarrow$) Let $\mathcal{S}$ be a satisfying set for φ. We are going to construct a quasimodel $\mathfrak{m} = \langle \mathfrak{F}, \mathfrak{R}, \lhd \rangle$ satisfying φ as a limit of a sequence of structures $\mathfrak{m}_i = \langle \mathfrak{F}_i, \mathfrak{R}_i, \lhd_i \rangle$ over some domains Δ_i, $i \geq 0$.

Step 1. Let $\Delta_0 = \emptyset$, $\mathfrak{m}_1 = \mathfrak{b}_o$ if $ob\varphi \neq \emptyset$ and let $\mathfrak{m}_1$ be some other block in $\mathcal{S}$ if $ob\varphi = \emptyset$, Δ_1 being the domain of $\mathfrak{m}_1$.

Step $n \geq 1$. For every $x \in \Delta_n - \Delta_{n-1}$, $x \notin ob\varphi$, we take the $\mathfrak{T}_x$-block $\mathfrak{b}_x = \langle \mathfrak{F}_x, \mathfrak{R}_x, \lhd_x \rangle \in \mathcal{S}$ over Δ^x. And if $x \in ob\varphi$, we take a $\mathfrak{T}_y$-block $\mathfrak{b}_y$ in $\mathcal{S}$ such that $\mathfrak{T}_y$ is an isomorphic copy of $\mathfrak{T}_a$ and replace $\mathfrak{T}_y$ in it with $\mathfrak{T}_a$. Without loss of generality we may assume that all Δ^x, the domains of the selected blocks, are disjoint from each other and $\Delta^x \cap \Delta_n = \{x\}$. We then define Δ_{n+1} to be the union of Δ_n and the domains Δ^x of the selected blocks. $\mathfrak{F}_{n+1}$ is the set of the corresponding type trees.

Now, suppose that we have weak runs $r \in \mathfrak{R}_n$ and $r_x \in \mathfrak{R}_x$, for every $x \in \Delta_n - \Delta_{n-1}$, such that $r(x) = r_x(x)$. Then we construct a weak run

$s = [r, \cup r_x]$, called an *extension* of r, by taking $s = \langle \Delta_s, \{R_s : R \in rol\varphi\}\rangle$, where

$$s(z) = \begin{cases} r(z) & \text{if } z \in \Delta_n, \\ r_x(z) & \text{if } z \in \Delta^x, \end{cases}$$

$R_s = \Delta_s \times \Delta_s$ if R is r- or r_x-universal for some $x \in \Delta_n - \Delta_{n-1}$ and

$$s(y) R_s s(z) \text{ iff } \begin{cases} r(y) R_r r(z) & \text{if } y, z \in \Delta_n, \\ r_x(y) R_{r_x} r_x(z) & \text{if } y, z \in \Delta^x \end{cases}$$

otherwise. The set of all weak runs of this sort is denoted by $\mathfrak{R}_{n+1}$. And finally, let us put $[r, \cup r_x] \lhd_{n+1} [r', \cup r'_x]$ iff $r \lhd r'$ and $r_x \lhd_x r'_x$ for all $x \in \Delta_n - \Delta_{n-1}$. This defines $\mathfrak{m}_{n+1}$. It is easily seen that it satisfies conditions (g)–(n) and (e).

Step ω. Finally, we define Δ and $\mathfrak{F}$ to be the unions of all Δ_i and $\mathfrak{F}_i$ for $i < \omega$, respectively. For every sequence of weak runs $\sigma = r_1, r_2, \ldots$ such that $r_{i+1} \in \mathfrak{R}_{i+1}$ is an extension of $r_i \in \mathfrak{R}_i$, we define a run r_σ by taking

$$\Delta_{r_\sigma} = \bigcup \{\Delta_{r_i} : r_i \in \sigma\},$$

$R_{r_\sigma} = \Delta_{r_\sigma} \times \Delta_{r_\sigma}$ if R_{r_i} is r_i-universal for some $i < \omega$ and

$$R_{r_\sigma} = \bigcup \{R_{r_i} : r_i \in \sigma\}$$

otherwise. Clearly, r_σ is a run through $\mathfrak{F}$. Let $\mathfrak{R}$ be the set of all runs of that sort. And for runs r_σ, $r_{\sigma'}$ ($\sigma' = r'_1, r'_2, \ldots$) in $\mathfrak{R}$ we put $r_\sigma \lhd r_{\sigma'}$ iff $r_i \lhd_i r'_i$ for all $i < \omega$.

It is not hard to see that $\mathfrak{m}$ is a quasimodel for φ. ❑

5 Decidability

Since one can effectively check, given an $\mathcal{ALC}_\mathcal{M}$-formula φ, whether there exists a satisfying set for φ, as an immediate consequence of Theorem 19 we obtain:

Theorem 20. *The satisfaction problem for $\mathcal{ALC}_\mathcal{M}$-formulas is decidable.*

It should be noted, however, that the obvious "brute-force" algorithm is extremely complex, it is even non-elementary.

So far we were considering satisfiability in *arbitrary* $\mathcal{ALC}_\mathcal{M}$-models. However, various specializations of the modal operators may impose different restrictions on the structure of the underlying frames in our models. For instance, if we understand $\Box$ as "it is known", we may need frames that are transitive, reflexive and symmetrical, i.e, **S5**-frames in the modal logic terminology, and if $\Box$ is intended to stand for "it is believed", then we may need **KD45**-frames which have the form of a non-degenerate cluster (i.e., an **S5**-frame) possibly with one irreflexive predecessor (for details consult [Fagin *et al.*, 1995]).

It is not hard to adopt the developed technique for proving the following:

Theorem 21. *There is an algorithm which is capable of deciding, given an arbitrary $\mathcal{ALC}_\mathcal{M}$-formula φ, whether φ is satisfiable in an $\mathcal{ALC}_\mathcal{M}$-model based upon* (i) *an* **S5**-*frame or* (ii) *a* **KD45**-*frame.*

Proof We sketch here only the most important modifications of the proof above for the case of **S5**-frames. First of all, instead of type trees we use now *type clusters* $\mathfrak{C}$ which are simply sets of distinct types for a given formula φ such that

- $\forall \tau \in \mathfrak{C} \forall \Diamond C \in con\varphi \ (\Diamond C \in \tau \Leftrightarrow \exists \tau' \in \mathfrak{C} \ C \in \tau')$;
- $\forall \tau \in \mathfrak{C} \forall \Diamond \psi \in sub\varphi \ (\Diamond \psi \in \tau \Leftrightarrow \exists \tau' \in \mathfrak{C} \ \psi \in \tau')$.

A type forest consists now of type clusters. And quasimodels have the form $\mathfrak{m} = \langle \mathfrak{F}, \mathfrak{R} \rangle$, where $\mathfrak{F}$ is a type forest over some $\Delta \supseteq ob\varphi$, and $\mathfrak{R}$ a set of runs through $\mathfrak{F}$ such that:

- $\forall x \in \Delta \forall \tau \in \mathfrak{C}_x \exists r \in \mathfrak{R} \ r(x) = \tau$;
- for all $x, y \in \Delta$, $\Diamond R \in rol\varphi$, and $r \in \mathfrak{R}$, we have $r(x)(\Diamond R)_r r(y)$ iff $\exists r' \in \mathfrak{R} \ r'(x) R_{r'} r'(y)$;
- for all $x, y \in \Delta$, $\Box R \in rol\varphi$, and $r \in \mathfrak{R}$, we have $r(x)(\Diamond R)_r r(y)$ iff $\forall r' \in \mathfrak{R} \ r'(x) R_{r'} r'(y)$.

The remaining part of the proof is similar to that of Theorem 19. It may be worth noting that now to construct Δ^x only two twins y_1, y_2 are enough. ❑

In some applications we may need $\mathcal{ALC}_\mathcal{M}$-models with *global roles*, i.e., roles R which are interpreted by the same binary relation in every world of a model. In quasimodels, we can reflect this by requiring that $r(x) R_r r(y)$ implies $r'(x) R_{r'} r'(y)$ for all $x, y \in \Delta$, and $r, r' \in \mathfrak{R}$; in other words, global roles correspond to binary relations between type trees. By a straightforward modification of the proof of Theorem 20 one can show the following:

Theorem 22. *There is an algorithm which is capable of deciding, given an arbitrary $\mathcal{ALC}_\mathcal{M}$-formula φ with global roles, whether φ is satisfiable in an $\mathcal{ALC}_\mathcal{M}$-model based upon* (i) *an arbitrary frame,* (ii) *an* **S5***-frame or* (iii) *a* **KD45***-frame.*

When dealing with intensional knowledge, one usually needs one modal operator $\Box_i$ for each agent i (meaning that "agent i knows" or "agent i believes"); see e.g. [Fagin *et al.*, 1995] for a discussion of propositional multimodal epistemic logics. Let $\mathcal{ALC}_{\mathcal{M}_n}$ denote the modal description language with n modal operators, so that $\mathcal{ALC}_{\mathcal{M}_n}$-models are based on Kripke frames with n accessibility relations $\lhd_1, \ldots, \lhd_n$. These frames are called *n-frames.* **S5**$_n$*-frames* and **KD45**$_n$*-frames* are those n-frames all monomodal fragments of which are **S5**-frames and **KD45**-frames, respectively. The developed technique provides us with a satisfiability checking algorithm for this multimodal case as well.

Theorem 23. *For every $n \geq 1$, there is an algorithm which is capable of deciding, for an arbitrary $\mathcal{ALC}_{\mathcal{M}_n}$-formula φ with global roles, whether φ is satisfiable in an $\mathcal{ALC}_{\mathcal{M}_n}$-model based upon* (i) *an arbitrary n-frame,* (ii) *an* **S5**$_n$*-frame or* (iii) *a* **KD45**$_n$*-frame.*

If all roles in the language $\mathcal{ALC}_{\mathcal{M}_n}$ are global then every $\mathcal{ALC}_{\mathcal{M}_n}$-model can be regarded as the Cartesian product of two multimodal frames of the form $\langle W, R_1, \ldots, R_n \rangle$ and $\langle V, S_1, \ldots, S_m \rangle$. Thus we obtain one more decision algorithm for the logics $\mathbf{K}_n \times \mathbf{K}_m$, $\mathbf{S5}_n \times \mathbf{K}_m$ and $\mathbf{KD45}_n \times \mathbf{K}_m$ (cf. [Gabbay and Shehtman, 1998]). Moreover, it follows from the proofs that all these logics have the product finite model property. In the case of $\mathbf{K} \times \mathbf{K}$, for instance, it suffices to make $md(\varphi)$ steps in the proof of Theorem 19 ($\Rightarrow$). Although the resulting structure is not in general a quasimodel, the corresponding model constructed as in the proof of Theorem 14 will satisfy φ (because the truth-values of modally deep subformulas of φ at the worlds located far from the root have no influence on the truth-value of φ at the root).

6 Undecidability

It would also be of interest to extend the constructed epistemic description language $\mathcal{ALC}_{\mathcal{M}_n}$ with the *common knowledge operator* $\mathbf{C}$ which is interpreted by the transitive and reflexive closure of the union $\lhd_1 \cup \ldots \cup \lhd_n$. (For various applications of $\mathbf{C}$ in the analysis of multi-agent systems see [Fagin *et al.*, 1995].) Another important kind of modality often used in applications is the temporal operator "always in the future" (or the operators "Since" and "Until") interpreted in linearly ordered sets of worlds (see e.g. [Gabbay *et al.*, 1994]). The satisfaction problem for these languages without global and modalized roles is known to be decidable (see [Wolter and Zakharyaschev, 1998a; Wolter and Zakharyaschev, 1998b; Wolter and Zakharyaschev, 1998c]). However, this is not the case for the more expressive language constructed in this paper.

Theorem 24. *The satisfaction problem for the epistemic description formulas with $n \geq 2$ agents and the common knowledge operator, and having global or modalized roles is undecidable in the class of $\mathbf{S5}_n$-frames.*

Proof Let $\mathbf{K} \times \mathbf{K}$ be the bi-modal logic determined by the class of all frames of the form $\mathfrak{F} \times \mathfrak{G}$, where $\mathfrak{F} = \langle W, R\rangle$ and $\mathfrak{G} = \langle V, S\rangle$ are usual Kripke frames for $\mathbf{K}$.[5] We will assume that $\mathbf{K} \times \mathbf{K}$ is formulated in the language $\mathcal{L}_1$ with the necessity operators $\boxminus$ and $\boxminus$ interpreted by R and S, respectively. Denote by $\vdash_1$ the *global consequence relation* for $\mathbf{K} \times \mathbf{K}$. That is, for any two $\mathcal{L}_1$-formulas φ and ψ, we have $\varphi \vdash_1 \psi$ iff ψ is true in every model for $\mathbf{K} \times \mathbf{K}$ in which φ is true. As was shown in [Marx, 1997], the global consequence problem "$\varphi \vdash_1 \psi$?" is undecidable.

Now, let $\mathbf{S5}_2^C \times \mathbf{K}$ be the 4-modal logic determined by the class of all frames of the form $\mathfrak{F} \times \mathfrak{G}$, where $\mathfrak{F}$ is an $\mathbf{S5}_2$-frame with accessibility relations R_1 and R_2, and $\mathfrak{G}$ is an arbitrary frame with a relation S. We assume that this logic is formulated in the language $\mathcal{L}_2$ with necessity operators $\Box_1$, $\Box_2$ (interpreted by R_1 and R_2, respectively), the common knowledge operator $\mathbf{C}$ (interpreted by $R_1 \cup R_2$), and $\boxminus$ (interpreted by S). Denote by $\vdash_2$ the global consequence relation for $\mathbf{S5}_2^C \times \mathbf{K}$.

Our first aim is to show that $\vdash_2$ is undecidable. We will do this by embedding $\mathcal{L}_1$ into $\mathcal{L}_2$ using the following translation T. Fix some propositional variable p

[5] We remind the reader that $\mathfrak{F} \times \mathfrak{G} = \langle W \times V, \overline{R}, \overline{S}\rangle$, where $\langle x, y\rangle\, \overline{R}\, \langle z, u\rangle$ iff xRz and $y = u$, $\langle x, y\rangle\, \overline{S}\, \langle z, u\rangle$ iff $x = z$ and ySu.

and put

$$
\begin{array}{rcl}
q^T & = & p \wedge q, \ \ q \text{ a variable;} \\
(\varphi \wedge \psi)^T & = & \varphi^T \wedge \psi^T; \\
(\neg\varphi)^T & = & p \wedge \neg\varphi^T; \\
(\boxminus\varphi)^T & = & p \wedge \boxminus\varphi^T; \\
(\boxplus\varphi)^T & = & p \wedge \Box_1(\neg p \to \Box_2(p \to \varphi^T)).
\end{array}
$$

Let φ and ψ be two arbitrary $\mathcal{L}_1$-formulas containing no occurrences of p. We will show that

$$\varphi \vdash_1 \psi \text{ iff } \chi \wedge (p \to \varphi^T) \vdash_2 p \to \psi^T,$$

where

$$\chi = \mathbf{C}((p \to \boxminus p) \wedge (\neg p \to \boxminus\neg p)).$$

Suppose that $\chi \wedge (p \to \varphi^T) \not\vdash_2 p \to \psi^T$. Then there is a model $\mathfrak{M}$ for $\mathbf{S5}_2^{\mathbf{C}} \times \mathbf{K}$ based on a frame $\langle W, R_1, R_2\rangle \times \langle V, S\rangle$ and having a valuation $\mathfrak{V}$ such that $\mathfrak{M} \models \chi \wedge (p \to \varphi^T)$ and $(\mathfrak{M}, \langle x, y\rangle) \not\models p \to \psi^T$, for some $\langle x, y\rangle \in W \times V$. Since $\mathfrak{M} \models \chi$, without loss of generality we may assume that $\langle x, y\rangle \models p$ iff $\langle x, z\rangle \models p$, for all $x \in W$ and $y, z \in V$. Fix a point $v \in V$ and define a relation R on W by taking, for $x_1, x_2 \in W$,

$$x_1 R x_2 \Leftrightarrow \exists z \in W (x_1 R_1 z R_2 x_2 \wedge \langle x_1, v\rangle \models p \wedge \langle z, v\rangle \not\models p \wedge \langle x_2, v\rangle \models p).$$

Define $\mathfrak{N}$ to be the $\mathcal{L}_1$-model based on the frame $\langle W', R\rangle \times \langle V, S\rangle$, where

$$W' = W \cap \{x \in W : \exists y \in V \ \langle x, y\rangle \models p\},$$

and having the restriction of $\mathfrak{V}$ to this frame as its valuation. Now, for all $\langle x_1, y_1\rangle \in W' \times V$ and all $\alpha \in \mathcal{L}_1$,

$$(\mathfrak{N}, \langle x_1, y_1\rangle) \models \alpha \text{ iff } (\mathfrak{M}, \langle x_1, y_1\rangle) \models \alpha^T.$$

The prove is by induction. We will consider here only the case of $\boxplus\alpha$. Let $(\mathfrak{N}, \langle x_1, y_1\rangle) \not\models \boxplus\alpha$, i.e., there is $x_2 \in W'$ such that $x_1 R x_2$ and $(\mathfrak{N}, \langle x_2, y_1\rangle) \not\models \alpha$. By the definition of R, we then have $z \in W$ for which $x_1 R_1 z R_2 x_2$, $(\mathfrak{M}, \langle x_1, y_1\rangle) \models p$, $(\mathfrak{M}, \langle z, y_1\rangle) \not\models p$, and $(\mathfrak{M}, \langle x_2, y_1\rangle) \models p$. And by the induction hypothesis we have $(\mathfrak{M}, \langle x_2, y_1\rangle) \not\models \alpha^T$. It follows that $(\mathfrak{M}, \langle x_1, y_1\rangle) \not\models (\boxplus\alpha)^T$.

Conversely, if $(\mathfrak{M}, \langle x_1, y_1\rangle) \models p$ and $(\mathfrak{M}, \langle x_1, y_1\rangle) \not\models (\boxplus\alpha)^T$ then we have some $x_2 \in W$ such that $x_1 R x_2$, $(\mathfrak{M}, \langle x_2, y_1\rangle) \models p$ and $(\mathfrak{M}, \langle x_2, y_1\rangle) \not\models \alpha^T$. By the induction hypothesis, it follows that $(\mathfrak{N}, \langle x_2, y_1\rangle) \not\models \alpha$ and so $(\mathfrak{N}, \langle x_1, y_1\rangle) \not\models \boxplus\alpha$. It follows that $\mathfrak{N} \models \varphi$ and $(\mathfrak{N}, \langle x, y\rangle) \not\models \psi$.

Now let us suppose that $\varphi \not\vdash_1 \psi$. Then there is a model $\mathfrak{N}$ based on a frame $\langle W, R\rangle \times \langle V, S\rangle$ and having a valuation $\mathfrak{V}$ such that $\mathfrak{N} \models \varphi$ and $(\mathfrak{N}, \langle x, y\rangle) \not\models \psi$, for some $\langle x, y\rangle \in W \times V$. Put $W' = W \cup R$ and let R_1, R_2 be the transitive, reflexive and symmetrical closures of the relations S_1 and S_2, respectively, where

$$x S_1 \langle y_1, y_2\rangle \text{ iff } y_1 = x \text{ and } x R y_2,$$

and

$$\langle y_1, y_2\rangle S_2 x \text{ iff } y_2 = x \text{ and } y_1 R x.$$

Define a valuation $\mathfrak{V}'$ on $W' \times V$ by taking $\mathfrak{V}'(q) = \mathfrak{V}(q)$, for every variable q different from p, and $\mathfrak{V}'(p) = W \times V$. Let $\mathfrak{M} = \langle \langle W', R_1, R_2 \rangle \times \langle V, S \rangle, \mathfrak{V}' \rangle$. It is not hard to show by induction that $\mathfrak{M} \models \chi \wedge (p \rightarrow \varphi^T)$ and $\mathfrak{M} \not\models p \rightarrow \psi^T$.

It follows that the global consequence relation $\vdash_2$ is undecidable. Using this we can easily prove the claim of our theorem.

Let us first show that the satisfaction problem for the epistemic description formulas with $n \geq 2$ agents and the common knowledge operator is undecidable in the class of $\mathbf{S5}_n$-frames with global roles. Fix a global role R. For an $\mathcal{L}_2$-formula φ, denote by φ^* the concept that results from φ by replacing every occurrence of $\boxminus$ in it with $\forall R$ and every variable p_i with the concept name C_i. Then we clearly have

$$\varphi \not\vdash_2 \psi \text{ iff } \mathbf{C}(\varphi^* = \top) \wedge \neg(\neg\psi^* = \bot) \text{ is satisfiable.}$$

To prove the undecidability of the satisfaction problem for the language with modalized, but without global roles, it suffices to observe that $\mathbf{C}R$ behaves itself like a global role, for any local role R. Thus, in presence of the common knowledge operator $\mathbf{C}$ global roles are definable by means of local ones in the language with modalized roles, and so this case reduces to the previous one. ❑

Theorem 25. *The satisfaction problem for $\mathcal{ALC}_\mathcal{M}$-formulas (with global or modalized roles) in strictly linearly ordered frames is undecidable; it is undecidable in $\langle \mathbb{N}, < \rangle$ as well.*

Proof The idea of the proof is borrowed from [Spaan, 1993] (Section 6.4). The undecidable problem we are going to reduce to the satisfaction problem for $\mathcal{ALC}_\mathcal{M}$-formulas in strictly linearly ordered frames is as follows: given a Turing machine, to determine whether it comes to a stop having started from the empty tape. To make the proof shorter, we will confine ourselves to considering only models based on the frame $\mathfrak{N} = \langle \mathbb{N}, < \rangle$.

Let $\mathfrak{A}$ be a single-tape right-infinite deterministic Turing machine with state space S, initial state s_0, tape alphabet A ($b \in A$ stands for blank) and transition function δ. The configurations of $\mathfrak{A}$ will be represented by infinite words of the form $\sharp a_0 \dots a_i \dots a_n b^\omega$, where $\sharp$ marks the left side of the tape, all $a_0, \dots, a_n$ save one, say a_i, are in A, while a_i belongs to $S \times A$ and represents the active cell and the current state. The start configuration, for instance, looks as follows: $\sharp(s_0, b)b^\omega$. Let $A' = A \cup \{\sharp\} \cup (S \times A)$.

Our aim is to construct an $\mathcal{ALC}_\mathcal{M}$-formula $\varphi_\mathfrak{A}$ which is satisfiable in a model based on $\mathfrak{N}$ iff, started from $\sharp(s_0, b)b^\omega$, $\mathfrak{A}$ never comes to a stop.

Let $\mathfrak{M} = \langle \mathfrak{N}, I \rangle$ be an $\mathcal{ALC}_\mathcal{M}$-model with domain Δ. To begin with, we partition the worlds in $\mathfrak{N}$ into an infinite number of disjoint intervals. This can be done using the concept name In and the formulas

$$\Box^+(In = \top \vee In = \bot), \tag{3}$$

$$\Box^+((In \rightarrow \Diamond\neg In) \wedge (\neg In \rightarrow \Diamond In) = \top). \tag{4}$$

(Here and below $\Box^+ E = E \wedge \Box E$, $\Diamond^+ E = E \vee \Diamond E$, E a formula or a concept.) Without loss of generality we may assume that if the conjunction of these two formulas is true in $\mathfrak{M}$ then the time line $0, 1, \dots$ is partitioned into an infinite sequence of intervals $i_0, i_1, \dots$ such that $i_0 = \{0, \dots, k\}$, $i_1 = \{k+1, \dots, l\}$, etc.,

$In^{I,m} = \top$, for all $m \in i_n$ with even n, and $In^{I,m} = \bot$ otherwise. To simplify notation, we will say that $x \in C$ (somewhere) in the interval i_j whenever $x \in C^n$ for all (some) $n \in i_j$.

Suppose now that C and D are concept names and $In_1(C, D)$ is the conjunction of the concepts:

$$\begin{gathered}
\Diamond^+ C, \\
\Box^+(C \wedge In \to \Box(\neg In \to \Box^+ \neg C)), \\
\Box^+(C \wedge \neg In \to \Box(In \to \Box^+ \neg C)), \\
\Box^+(In \wedge \Diamond C \wedge \neg\Diamond(\neg In \wedge \Diamond C) \to C), \\
\Box^+(\neg In \wedge \Diamond C \wedge \neg\Diamond(In \wedge \Diamond C) \to C), \\
\Box^+(C \wedge In \to \Diamond(D \wedge \neg In)), \\
\Box^+(C \wedge \neg In \to \Diamond(D \wedge In)), \\
\Box^+(D \wedge In \to \Box^+(\neg In \to \Box^+ \neg D)), \\
\Box^+(D \wedge \neg In \to \Box^+(In \to \Box^+ \neg D)), \\
\Box^+(C \to \Box^+(\neg D \wedge \Diamond D \to C)).
\end{gathered}$$

It is easy to see that if $x \in In_1(C, D)$ in the world 0 then there is an interval i_j such that $x \in C$ in i_j, $x \notin C$ outside i_j, $x \in D$ somewhere in i_{j+1}, and $x \notin D$ outside i_{j+1}. It follows in particular that if x belongs to the concept

$$In_n(C_1, \dots, C_{n+1}) = In_1(C_1, C_2) \wedge In_1(C_2, C_3) \wedge \cdots \wedge In_1(C_n, C_{n+1})$$

in the world 0 then there are n consecutive intervals $i_{j_1}, \dots, i_{j_n}$ such that $x \in C_k$, for $k \in [1, n]$, only in i_{j_k}, while $x \in C_{n+1}$ only somewhere in $i_{j_{n+1}}$.

With every $\alpha \in A'$ we associate a concept name C_α. The formula

$$\Box^+(\bigvee_{\alpha \in A'} (C_\alpha \wedge \neg \bigvee_{\alpha \neq \beta \in A'} C_\beta) = \top) \tag{5}$$

true in $(\mathfrak{M}, 0)$ means that every object in every world belongs precisely to one of the concepts C_α, for $\alpha \in A'$.

To encode the initial configuration we use an object name e and the formula

$$e : C_\sharp \wedge In_2(C_\sharp, C_{(s_0,b)}, C') \wedge \Box^+(C_\sharp \vee C_{(s_0,b)} \vee C_b). \tag{6}$$

It says that $e \in C_\sharp$ in i_0, $e \in C_{(s_0,b)}$ in i_1, and $e \in C_b$ in all other intervals.

To simulate the transition function we "mark" every object x in some intervals by one of the three concept names L, S, or R by means of the following formulas

$$In_3(L, S, R, D) = \top, \tag{7}$$

$$\Box^+((\bigvee_{(s,a) \in A'} C_{(s,a)} \leftrightarrow S) = \top). \tag{8}$$

Thus, if $x \in C_{(s,a)}$ in i_j then $x \in L$ in i_{j-1}, $x \in S$ in i_j, and $x \in R$ in i_{j+1}.

The transition from one configuration to another is simulated by a global role name T with the help of the formulas:

$$\begin{aligned}
&(\Diamond^+(L \wedge C_\alpha) \wedge \Diamond^+(S \wedge C_\beta) \wedge \Diamond^+(R \wedge C_\gamma) \to \\
&\quad \exists T.\top \wedge \Box^+(L \to \forall T.C_{\alpha'}) \wedge \Box^+(S \to \forall T.C_{\beta'}) \wedge \Box^+(R \to \forall T.C_{\gamma'})) = \top,
\end{aligned} \tag{9}$$

for all $\delta(\alpha, \beta, \gamma) = (\alpha', \beta', \gamma')$,

$$(\Diamond^+(L \wedge C_\alpha) \wedge \Diamond^+(S \wedge C_\beta) \wedge \Diamond^+(R \wedge C_\gamma) \to \neg\exists T.\top) = \top, \tag{10}$$

for all triples $\alpha, \beta, \gamma \in A'$, $\beta \in S \times A$, such that $\delta(\alpha, \beta, \gamma)$ is not defined, and

$$\bigwedge_{\alpha \in A'} \Box^+(\neg L \wedge \neg S \wedge \neg R \wedge C_\alpha \to \forall T.C_\alpha) = \top. \tag{11}$$

It remains only to ensure that there is an infinite chain of T-arrows starting from e. This can be done using the formulas:

$$e : C, \quad \Box^+((C \to \Box C) = \top), \tag{12}$$

$$\Box^+(C \to \exists T.\neg C = \top), \quad \Box^+((\neg C \to \Diamond C) = \top). \tag{13}$$

Let $\varphi_{\mathfrak{A}}$ be the conjunction of (3)–(13). We leave it to the reader to check that $\varphi_{\mathfrak{A}}$ is satisfied in a model based on $\mathfrak{N}$ iff $\mathfrak{A}$ has an infinite computation which starts from the empty tape. If we do not have global roles then we can easily modify the formulas above using $\Box T$ instead of T. ❑

Thus, for "strong" modal description logics (like epistemic logics with common knowledge operators or temporal ones) the language without modalized and global roles seems to be an optimal compromise between the expressive power and decidability. It is worth mentioning, however, that [Wolter, 1998] used another approach to construct decidable modal description logics with an expressive modal component. It is based on the observation that quite often it is complex formulas rather than complex concepts that cause undecidability. Decidability can be recovered by considering languages with only concepts. For example, it is shown in [Wolter, 1998] that the satisfaction problem for concepts (i.e., the satisfaction problem for formulas of the form $\neg(C = \bot)$) is decidable for the epistemic languages with the common knowledge operator and global roles. The same concerns various temporal logics.

7 Discussion and open problems

This paper makes one more step in the study of concept description languages of high expressive power that are located near the border between decidable and undecidable. We have designed a "full" multi-dimensional modal description language which imposes no restrictions whatsoever on the use of modal operators (they can be applied to all types of syntactic terms: concepts, roles and formulas) and contains both local and global object, concept and role names. Using the mosaic technique we have proved that the satisfaction problem for the formulas of this language (and so many other reasoning tasks as well) is decidable in some important classes of models. (Actually, this gives a solution to a problem raised in [Baader and Ohlbach, 1993].) On the other hand, it was shown that the language becomes undecidable when interpreted on temporal structures or augmented with the common knowledge operator.

The obtained results demonstrate a principle possibility of using this highly expressive language in knowledge representation systems. Further investigations are required to make it really applicable. In particular, it would be of interest to answer the following questions:

(1) Do the logics considered above have the finite model property?

Our conjecture is that they do have this property, and so the finite model reasoning in those logics is effective.

(2) What is the complexity of satisfiability checking in these logics?

We only know that the satisfaction problem in all of them is NEXPTIME-hard.

(3) Is it possible to extend the developed technique to transitive frames?

Our method is heavily based on the fact that models of depth $\leq \varphi$ are always enough to satisfy a given formula φ. This is not the case when the accessibility relations are transitive, i.e., satisfy the natural epistemic axiom $\Box\varphi \rightarrow \Box\Box\varphi$. In particular, a challenging open problem is

(4) to find out whether the logics $\mathbf{K4} \times \mathbf{K4}$ and $\mathbf{S4} \times \mathbf{S4}$ are decidable.

To increase the language's capacity of expressing the dynamics of relations between individual objects in application domains it would be desirable also

(5) to extend $\mathcal{ALC}_{\mathcal{M}_n}$ with (some of) the booleans operating on roles,

(6) to extend the underlying description logic with new constructs

and, of course, retain decidability.

References

[Baader and Hanschke, 1991] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, 1991.

[Baader and Hollunder, 1991] F. Baader and B. Hollunder. KRIS: Knowledge representation and inference system. *SIGART Bulletin*, 2:8–14, 1991.

[Baader and Laux, 1995] F. Baader and A. Laux. Terminological logics with modal operators. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 808–814, Montreal, Canada, 1995. Morgan Kaufman.

[Baader and Ohlbach, 1993] F. Baader and H. Ohlbach. A multi-dimensional terminological knowledge representation language. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 690–695, 1993.

[Baader and Ohlbach, 1995] F. Baader and H.J. Ohlbach. A multi-dimensional terminological knowledge representation language. *Journal of Applied Non-Classical Logic*, 5:153–197, 1995.

[Borgida *et al.*, 1989] A. Borgida, R. Brachman, D. Mc Guinness, and L. Resnick. CLASSIC: A structural data model for objects. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 59–67, 1989.

[Borgida, 1995] A. Borgida. Description logics in data management. *IEEE Trans. on Knowledge and Data Engineering*, 7:671–682, 1995.

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

[Brachman *et al.*, 1991] R. Brachman, D. McGuiness, P. Patel-Schneider, L. Resnick, and A. Borgida. Living with CLASSIC: when and how to use a KL-ONE like language. In *Principles of Semantic Networks*, pages 401–456, 1991.

[Calvanese *et al.*, 1999] D. Calvanese, G. De Giacomo, D. Nardi, and M. Lenzerini. Reasoning in expressive description logics. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*. Elsevier Science Publishers B.V., 1999. In print.

[Catarci and Lenzerini, 1993] T. Catarci and M. Lenzerini. Representing and using interschema knowledge in cooperative information systems. *J. of Intelligent and Cooperative Information Systems*, 2:375–398, 1993.

[Chagrov and Zakharyaschev, 1997] A.V. Chagrov and M.V. Zakharyaschev. *Modal Logic*. Oxford Logic Guides 35. Clarendon Press, Oxford, 1997.

[Donini *et al.*, 1992] F. Donini, M. Lenzerini, D. Nardi, A. Schaerf, and W. Nutt. Adding epistemic operators to concept languages. In *Principles of Knowledge Representation and Reasoning*, pages 342–353, 1992.

[Donini *et al.*, 1996] F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in description logics. In G. Brewka, editor, *Principles of Knowledge Representation*, pages 191–236. CSLI Publications, 1996.

[Fagin *et al.*, 1995] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[Finger and Gabbay, 1992] M. Finger and D. Gabbay. Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information*, 2:203–233, 1992.

[Gabbay and Shehtman, 1993] D. Gabbay and V. Shehtman. Undecidability of modal and intermediate first-order logics with two individual variables. *Journal of Symbolic Logic*, 58:800–823, 1993.

[Gabbay and Shehtman, 1998] D. Gabbay and V. Shehtman. Products of modal logics. Part I. *Journal of the IGPL*, 6:73–146, 1998.

[Gabbay *et al.*, 1994] D. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic*. Oxford University Press, 1994.

[Giacomo and Lenzerini, 1996] G. De Giacomo and M. Lenzerini. TBox and ABox reasoning in expressive description logics. In *Proceedings of the fifth Conference on Principles of Knowledge Representation and Reasoning*, Montreal, Canada, 1996. Morgan Kaufman.

[Gräber *et al.*, 1995] A. Gräber, H. Bürckert, and A. Laux. Terminological reasoning with knowledge and belief. In A. Laux and H. Wansing, editors, *Knowledge and Belief in Philosophy and Artificial Intelligence*, pages 29–61. Akademie Verlag, 1995.

[Horrocks, 1998] I. Horrocks. Using an expressive description logic: FACT or fiction? In *Proceedings of the ninth International Conference on Principles of Knowledge Representation and Reasoning*, pages 636–647, 1998.

[Hughes and Cresswell, 1996] G.E. Hughes and M.J. Cresswell. *A New Introduction to Modal Logic*. Methuen, London, 1996.

[Kobsa, 1989] A. Kobsa. The SB-ONE knowledge representation workbench. In *Proceedings of the Workshop on Formal Aspects of Semantic Networks*, 1989.

[Laux, 1994] A. Laux. Beliefs in multi-agent worlds: a terminological approach. In *Proceedings of the 11th European Conference on Artificial Intelligence*, pages 299–303, Amsterdam, 1994.

[Marx, 1997] M. Marx. Complexity of products of modal logics. Submitted, 1997.

[Patel-Schneider, 1998] P. Patel-Schneider. DLP system description. In *Proceedings of the 1998 International Workshop on Description Logioc*, pages 87–89, 1998. ITC Report 9805-03.

[Peltason, 1991] C. Peltason. The BACK-system — an overview. *SIGART Bulletin*, 2:114–119, 1991.

[Quantz and Kindermann, 1990] J. Quantz and C. Kindermann. Implementation of the BACK-system version 4. Technical report, FB Informatik, Technische Universität Berlin, 1990.

[Schild, 1991] K. Schild. A correspondence theory for terminological logics: preliminary report. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)*, pages 466–471, Sydney, 1991.

[Schild, 1993] K. Schild. Combining terminological logics with tense logic. In *Proceedings of the 6th Portuguese Conference on Artificial Intelligence*, pages 105–120, Porto, 1993.

[Schmidt-Schauß and Smolka, 1991] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.

[Schmiedel, 1990] A. Schmiedel. A temporal terminological logic. In *Proceedings of the 9th National Conference of the American Association for Artificial Intelligence*, pages 640–645, Boston, 1990.

[Spaan, 1993] E. Spaan. *Complexity of Modal Logics*. PhD thesis, Department of Mathematics and Computer Science, University of Amsterdam, 1993.

[Wolter and Zakharyaschev, 1998a] F. Wolter and M. Zakharyaschev. Dynamic description logic. In *Proceedings of AiML'98*, Uppsala, 1998. To appear in "Advances in Modal Logic. Volume II", CSLI Publications, Stanford, 1999. See `http://www.informatik.uni-leipzig.de/~wolter`.

[Wolter and Zakharyaschev, 1998b] F. Wolter and M. Zakharyaschev. Satisfiability problem in description logics with modal operators. In *Proceedings of the sixth Conference on Principles of Knowledge Representation and Reasoning*, pages 512–523, Montreal, Canada, 1998. Morgan Kaufman.

[Wolter and Zakharyaschev, 1998c] F. Wolter and M. Zakharyaschev. Temporalizing description logics. In *Proceedings of FroCoS'98*, Amsterdam, 1998. To appear in "Frontiers of Combining Systems", Kluwer Academic Publishers, 1999. See `http://www.informatik.uni-leipzig.de/~wolter`.

[Wolter, 1998] F. Wolter. The product of converse PDL and polymodal K. Manuscript. See `http://www.informatik.uni-leipzig.de/~wolter`, 1998.

[Wright *et al.*, 1993] G.T. Wright, E.S. Weixelbaum, G.T. Vesonder, K.E. Brown, S.R. Palmer, J.I. Berman, and H.H. Moore. A knowledge-based configurator that supports sales, engineering, and manufacturing at AT&T network systems. *AI Magazine*, 14:69–80, 1993.