



HAL
open science

Two decision problems in Contact Logics

Philippe Balbiani, Cigdem Gencer, Zafer Özdemir

► **To cite this version:**

Philippe Balbiani, Cigdem Gencer, Zafer Özdemir. Two decision problems in Contact Logics. Logic Journal of the IGPL, 2019, 27 (1), pp.8-32. 10.1093/jigpal/jzy016 . hal-02378372

HAL Id: hal-02378372

<https://hal.science/hal-02378372>

Submitted on 25 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in:

<http://oatao.univ-toulouse.fr/24760>

Official URL

DOI : <https://doi.org/10.1093/jigpal/jzy016>

To cite this version: Balbiani, Philippe and Gencer, Cigdem and Özdemir, Zafer *Two decision problems in Contact Logics*. (2019) *Logic Journal of the IGPL*, 27 (1). 8-32. ISSN 1367-0751

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Two decision problems in Contact Logics

PHILIPPE BALBIANI*, *Institut de recherche en informatique de Toulouse, CNRS — Toulouse University, 118 route de Narbonne, 31062 Toulouse Cedex 9, France*

ÇİĞDEM GENCER, *Faculty of Arts and Sciences, Istanbul Aydın University.*

ZAFER ÖZDEMİR, *Institut de Recherche en Informatique de Toulouse, Toulouse University.*

Abstract

Contact Logics provide a natural framework for representing and reasoning about regions in several areas of computer science. In this paper, we focus our attention on reasoning methods for Contact Logics and address the satisfiability problem and the unifiability problem. Firstly, we give sound and complete tableaux-based decision procedures in Contact Logics and we obtain new results about the decidability/complexity of the satisfiability problem in these logics. Secondly, we address the computability of the unifiability problem in Contact Logics and we obtain new results about the unification type of the unifiability problem in these logics.

Keywords: Contact Logics, satisfiability problem, unifiability problem.

1 Introduction

In order to model situations arising in domains such as geographical information systems, scene modelling information systems or robot navigation, the need for building discrete region-based theories of space arises. Such theories have been proposed by Galton [13, 14] who considered the so called adjacency spaces, i.e. pairs (W, R) where W is a nonempty set of *indivisible regions* or *cells* and R is a binary relation between regions. If two cells in W are in the relation R then they are said to be adjacent. A well-known example of an adjacency space is the chess-board. Its cells are the black and white squares of the chess-board and its adjacency relation is the binary relation that holds for two squares if and only if they have a common point. In this case, the adjacency relation between cells is reflexive and symmetric.

Following this idea, Balbiani *et al.* [7] and Vakarelov [27] define *regions* in a frame (W, R) to be arbitrary subsets of W . In their setting, the operations over regions are the Boolean operations of join and complement whereas the relations between regions are the *contact* relation and the *equality* relation. The binary relation $C_{(W,R)}$ of contact between two regions A, B of W is defined as follows: $C_{(W,R)}(A, B)$ iff A contains a point x and B contains a point y such that xRy . As shown in [7, 27], the language of Contact Logics can be seen as a *first-order* language without quantifiers. See also [10, 11]. It can also be seen as a linguistic restriction of the ordinary language of modal logics with the universal box $[U]$ and the universal diamond $\langle U \rangle$, as we will see later the basic formulas $C(a, b)$ and $a \equiv b$ being, respectively, equivalent to the modal formulas $\langle U \rangle(a \wedge \diamond b)$ and $[U](a \leftrightarrow b)$. As a result, most concepts, tools and techniques typical of ordinary modal logics can be applied to Contact Logics. In this paper, we will address the satisfiability problem and the unifiability problem in Contact Logics.

*E-mail: philippe.balbiani@irit.fr

As for any semantically-based logical formalism, it is of course of the utmost importance to be able to decide the satisfiability problem in Contact Logics. But we also believe that it is essential to be able to decide the unifiability problem in Contact Logics. The unifiability problem has been introduced in modal logics as a special case of the admissibility problem [16, 19, 24]. In description logics, it has been introduced as a tool for detecting redundancies in knowledge bases [2, 3]. In a semantically-presented logic, it can be defined as follows [1]: determine whether a given formula becomes valid after replacing its variables by appropriate expressions. An important question in unification theory is as follows [12]: if a formula is unifiable, does it have a minimal complete set of unifiers? If the answer is ‘yes’, how large is this set?

Our motivation for considering the unifiability problem in Contact Logics comes from the fact that there is a wide variety of situations where it arises. In geographical information systems, such a situation arises when one has to synthesize geographical regions validating a given set of geographical constraints like ‘at the border of’, ‘to the North of’, etc. In scene modelling information systems, such a situation arises when one has to synthesize complex objects validating spatial constraints like ‘overlaps’, ‘contains’, etc. In robot navigation, such a situation arises when one has to synthesize trajectories of mobile agents validating a given set of positional constraints like ‘to the left of’, ‘far from’, etc. Such situations involving variables p_1, \dots, p_n representing unknown regions in some real space can be formalized by a formula $\varphi(p_1, \dots, p_n)$. By determining whether there exists terms a_1, \dots, a_n such that the formula $\varphi(a_1, \dots, a_n)$ is valid, one becomes able to synthesize complex regions a_1, \dots, a_n validating the formalized situations.

Concerning the satisfiability problem, an interesting result for Contact Logics is the following: the satisfiability problem with respect to the class of all models or with respect to the class of all reflexive and symmetric models is **NP**-complete (recall that the satisfiability problem with respect to these classes of models is **PSPACE**-complete if one considers the ordinary language of modal logics and **EXPTIME**-complete if this language is extended with the universal diamond). See [7] for details. Several computability results have also been obtained when Contact Logics are interpreted in topological spaces and when their languages are extended with connectedness predicates [20–22, 28]. For several classes of frames, we do not precisely know how complex is the satisfiability problem for contact formulas. For instance, for the class of all serial models or the class of all dense models. It is the purpose of this paper to give answer to these open questions. See Sections 5 and 6.

Concerning the unifiability problem, we are interested in supporting a new inference capability: unifiability of formulas. The unifiability problem consists, given a formula $\varphi(p_1, \dots, p_n)$, in determining whether there exist terms a_1, \dots, a_n such that $\varphi(a_1, \dots, a_n)$ is valid. And if one is able to find such appropriate substitutions, then one is interested to find the maximal ones. If a set of formulas is unifiable then an important question is as follows: does it have a minimal complete set of unifiers? If the answer is ‘yes’, how large is this set? It is the purpose of this paper to give answer to these open questions. See Sections 7–11.

In this paper, we focus our attention on reasoning methods for Contact Logics and address the two following decision problems: the satisfiability problem and the unifiability problem. Sections 2 and 3 introduce the syntax and semantics of Contact Logics whereas Section 4 presents some mathematical aspects of Contact Logics. In Sections 5 and 6, we give sound and complete tableaux-based decision procedures in Contact Logics and we obtain new results about the decidability/complexity of the satisfiability problem in these logics. In Sections 7–11, we address the computability of the unifiability problem in Contact Logics and we obtain new results about the unification type of the unifiability problem in these logics.

2 Syntax of Contact Logics

It is now time to meet the language of Contact Logics we will be working with. Let \mathbf{BV} be a countably infinite set of *Boolean variables* (with members denoted p, q , etc). Let (p_1, p_2, \dots) be an enumeration of \mathbf{BV} without repetitions. As usual, we will follow the standard rules for omission of the parentheses. The set of all *Boolean terms* based on \mathbf{BV} (with members denoted a, b , etc) is defined as follows:

- $a := p \mid 0 \mid \neg a \mid (a \cup b)$.

The other Boolean constructs for terms ($1, \cap$, etc) are defined as usual: 1 is $\neg 0$, $(a \cap b)$ is $\neg(\neg a \cup \neg b)$, etc. We use the following notations for terms: a^0 for $\neg a$ and a^1 for a . Reading terms as regions, the constructs $0, \neg$ and \cup should be regarded as the empty region, the complement operation and the union operation, respectively. As a result, the constructs 1 and \cap should be regarded as the full region and the intersection operation, respectively. Examples of terms are $(p \cap \neg q) \cup (\neg p \cap q)$ and $\neg p \cup \neg q$. For all $\mathbf{BV}' \subseteq \mathbf{BV}$, let $TER(\mathbf{BV}')$ be the set of all terms whose variables form a subset of \mathbf{BV}' . In the sequel, we use $a(p_1, \dots, p_n)$ to denote a term a whose variables form a subset of $\{p_1, \dots, p_n\}$. Let $a(p_1, \dots, p_n)$ be such a term. The result of the *uniform replacement* in each of their occurrences of the variables p_1, \dots, p_n by the terms b_1, \dots, b_n is denoted $a(b_1, \dots, b_n)$.

The set of all *formulas* based on \mathbf{BV} (with members denoted φ, ψ , etc) is defined as follows:

- $\varphi := C(a, b) \mid a \equiv b \mid \perp \mid \neg \varphi \mid (\varphi \vee \psi)$.

The other Boolean constructs for formulas (\top, \wedge , etc) are defined as usual: \top is $\neg \perp$, $(\varphi \wedge \psi)$ is $\neg(\neg \varphi \vee \neg \psi)$, etc. Given terms a and b , let $a \not\equiv b$ be $\neg a \equiv b$, $a \leq b$ be $(a \cup b) \equiv b$ and $a \not\leq b$ be $\neg(a \cup b) \equiv b$. The intuitive readings of the formulas $C(a, b)$ and $a \equiv b$ are ‘region a is in contact with region b ’ and ‘region a is equal to region b ’. When a and b are arbitrary terms, examples of formulas are $a \not\equiv 0 \rightarrow C(a, a)$ (‘if a is nonempty then a is in contact with itself’) and $C(a, b) \rightarrow C(b, a)$ (‘if a is in contact with b then b is in contact with a ’). For all $\mathbf{BV}' \subseteq \mathbf{BV}$, let $FOR(\mathbf{BV}')$ be the set of all formulas whose variables form a subset of \mathbf{BV}' . In the sequel, we use $\varphi(p_1, \dots, p_n)$ to denote a formula φ whose variables form a subset of $\{p_1, \dots, p_n\}$. Let $\varphi(p_1, \dots, p_n)$ be such a formula. The result of the *uniform replacement* in each of their occurrences of the variables p_1, \dots, p_n by the terms a_1, \dots, a_n is denoted $\varphi(a_1, \dots, a_n)$.

As the reader can see, formulas are quantifier-free first-order formulas in a language based on the function symbols 0 (arity 0), \neg (arity 1) and \cup (arity 2) and the binary predicate C of contact and the binary predicate \equiv of equality. As we will soon see, the contact language we are working with is a variant of the **BRCC8** language elaborated by Wolter and Zakharyashev [28]. See also [20–22]. **BRCC8** is the result of the combination of **RCC8** with Boolean reasoning. Within the context of **RCC8**, formulas would just be quantifier-free first-order formulas in a function-free language based on the 8 binary predicates DC (‘disconnected’), EC (‘external contact’), PO (‘partial overlap’), TPP (‘tangential proper part’), $TPPI$ (‘inverse of TPP ’), $NTPP$ (‘nontangential proper part’), $NTPPI$ (‘inverse of $NTPP$ ’) and EQ (‘equal’) of **RCC8**. For instance, $TPP(p, q) \wedge TPP(p, r) \rightarrow TPP(q, r) \vee TPP(r, q)$ (‘if p is a tangential proper part both of q and r then either q is a tangential proper part of r , or r is a tangential proper part of q ’). By allowing to apply the 8 binary predicates of **RCC8** not only to variables but also to terms, Wolter and Zakharyashev [28] have strictly extended their expressive capacity. For instance, in the class of all topological spaces, the **BRCC8** formula $EQ(p \cup q, r)$ (‘the union of p and q is equal to r ’) has no equivalent formula in a pure **RCC8**-based language. As well, with this enriched language, one becomes able by using the **BRCC8** formula $DC(p, \neg p) \rightarrow EQ(p, 0) \vee EQ(\neg p, 0)$ (‘if p is disconnected with the complement of p then either p is equal to the

empty region, or the complement of p is equal to the empty region') to distinguish between connected and non-connected topological spaces. Using the binary predicates of contact, it is possible to define the 8 above-mentioned binary predicates by letting $DC(a, b)$ be $\neg C(a, b)$, $EC(a, b)$ be $C(a, b) \wedge (a \cap b) \equiv 0$, $PO(a, b)$ be $(a \cap b) \neq 0 \wedge a \not\leq b \wedge b \not\leq a$, $TPP(a, b)$ be $a \leq b \wedge b \not\leq a \wedge C(a, -b)$, $TPPI(a, b)$ be $a \not\leq b \wedge b \leq a \wedge C(-a, b)$, $NTPP(a, b)$ be $a \neq b \wedge \neg C(a, -b)$, $NTPPI(a, b)$ be $a \neq b \wedge \neg C(-a, b)$ and $EQ(a, b)$ be $a \equiv b$.

3 Semantics of Contact Logics

It is now time to present the semantics of Contact Logics we will be working with. The best way to understand the meaning of the binary predicate of contact is by interpreting terms and formulas in topological spaces [20–22, 28]. Nevertheless, a relational perspective is suggested by Galton [14] who introduces the notion of adjacency space. Galton's spaces are frames (W, R) where W is a nonempty set of cells and R is an adjacency relation between cells. Galton defines regions to be sets of cells. He also defines two regions A and B to be connected iff some cell in A is adjacent to some cell in B . This definition relates Galton's adjacency spaces to the relational semantics of modal logics which makes it possible to use methods from modal logics for studying region-based theories of space. The truth is that the above-mentioned topological semantics and the relational perspective suggested by Galton are equivalent [27].

A *frame* is a pair $\mathcal{F} = (W, R)$ where W is a nonempty set and $R \subseteq W \times W$. The elements of W can be seen as points in some geometrical structure and the binary relation R between them can be seen as a proximity relation. The elements of W are called *points* and, naturally, *regions* are sets of points, i.e. elements of 2^W . To the binary predicate C of contact, Galton associates the binary relation $C^{(W,R)}$ defined on the set of all subsets of W as follows: $C^{(W,R)}(A, B)$ iff $(A \times B) \cap R \neq \emptyset$. Hence, when regions are subsets of some frame as in the relational semantics of Contact Logics, two regions are in contact iff their Cartesian product and the frame's proximity relation intersect. Obviously, when R is reflexive and symmetric, the following conditions hold: if $A \neq \emptyset$ then $C^{(W,R)}(A, A)$ and if $C^{(W,R)}(A, B)$ then $C^{(W,R)}(B, A)$. If $R = W \times W$ then the following condition holds: if $A \neq \emptyset$ and $B \neq \emptyset$ then $C^{(W,R)}(A, B)$. In that case, we say that the frame (W, R) is *indiscrete*. If $R^+ = W \times W$ then the following condition holds: if $A \neq \emptyset$ and $W \setminus A \neq \emptyset$ then $C^{(W,R)}(A, W \setminus A)$. In that case, we say that the frame (W, R) is *connected*. Let \mathcal{C}_{all} , \mathcal{C}_{ind} and \mathcal{C}_{con} , respectively, be the class of all frames, the class of all indiscrete frames and the class of all connected frames.

A *model* based on \mathcal{F} is a 3-tuple $\mathcal{M} = (W, R, V)$ where $V: \mathbf{BV} \rightarrow 2^W$. The function V is the *valuation function* of the model. It associates a region of \mathcal{M} to each variable of the language. It can be extended into the function \bar{V} defined as follows on the set of all terms:

- $\bar{V}(p) = V(p)$,
- $\bar{V}(0) = \emptyset$,
- $\bar{V}(\neg a) = W \setminus \bar{V}(a)$,
- $\bar{V}(a \cup b) = \bar{V}(a) \cup \bar{V}(b)$.

Thus, every term is interpreted as a subset of W . Now, we have everything that is needed to define the *satisfiability relation* between a model $\mathcal{M} = (W, R, V)$ and a formula φ , in symbols $\mathcal{M} \models \varphi$, as follows:

- $\mathcal{M} \models C(a, b)$ iff there exists $x, y \in W$ such that $x \in \bar{V}(a)$, $y \in \bar{V}(b)$ and xRy ,
- $\mathcal{M} \models a \equiv b$ iff $\bar{V}(a) = \bar{V}(b)$,
- $\mathcal{M} \not\models \perp$,

- $\mathcal{M} \models \neg\varphi$ iff not $\mathcal{M} \models \varphi$,
- $\mathcal{M} \models \varphi \vee \psi$ iff $\mathcal{M} \models \varphi$ or $\mathcal{M} \models \psi$.

As the reader can see, Contact Logics can be seen as a linguistic restriction of the ordinary language of modal logics with the universal box $[U]$ and the universal diamond $\langle U \rangle$, its basic formulas $C(a, b)$ and $a \equiv b$ being, respectively, equivalent to the modal formulas $\langle U \rangle(a \wedge \langle U \rangle b)$ and $[U](a \leftrightarrow b)$.

We shall say that φ is *satisfiable* in a frame \mathcal{F} iff $\mathcal{M} \models \varphi$ for some \mathcal{M} based on \mathcal{F} . φ is said to be *valid* in \mathcal{F} iff $\mathcal{M} \models \varphi$ for every \mathcal{M} based on \mathcal{F} . Let \mathcal{C} be a class of frames. We shall say that φ is *satisfiable* in \mathcal{C} iff $\mathcal{M} \models \varphi$ for some \mathcal{M} based on a frame in \mathcal{C} . φ is said to be *valid* in \mathcal{C} iff $\mathcal{M} \models \varphi$ for every \mathcal{M} based on a frame in \mathcal{C} . Let $\text{sat}(\mathcal{C})$ be the set of all formulas that are satisfiable in \mathcal{C} and $\text{val}(\mathcal{C})$ be the set of all formulas that are valid in \mathcal{C} . Examples of formulas valid in the class of all frames are $C(a, b) \rightarrow a \neq 0 \wedge b \neq 0$, $C(a, b \cup c) \leftrightarrow C(a, b) \vee C(a, c)$ and $C(a \cup b, c) \leftrightarrow C(a, c) \vee C(b, c)$. The formulas $a \neq 0 \rightarrow C(a, a)$ and $C(a, b) \rightarrow C(b, a)$ are valid in, respectively, the class of all reflexive frames and the class of all symmetric frames.

4 Mathematical aspects of Contact Logics

We say that a class \mathcal{C} of frames is *determined* iff there exists a set of formulas such that \mathcal{C} is the class of all frames validating each formula in that set. At first sight, the Contact Logics introduced above seem to indicate that the capacity of their language to determine a particular class of frames is weak. In spite of their simplicity, Contact Logics turn out to be a useful tool for describing relational structures. Let $\mathcal{F} = (W, R)$ be a frame. We shall say that

- \mathcal{F} is *reflexive* iff for all $x \in W$, xRx ,
- \mathcal{F} is *serial* iff for all $x \in W$, there exists $y \in W$ such that xRy ,
- \mathcal{F} is *dense* iff for all $x, y \in W$, if xRy then there exists $z \in W$ such that xRz and zRy ,
- \mathcal{F} is *connected* iff for all $x, y \in W$, if $x \neq y$ then there exists a positive integer N and there exists a sequence (z_0, \dots, z_N) in W such that $z_0 = x$, $z_N = y$ and for all positive integers k , if $k \leq N$ then $z_{k-1}Rz_k$,
- \mathcal{F} is *non-2-colourable* iff points in W cannot be coloured by colours from a given set of 2 colours such that each two points connected by R have different colours,
- \mathcal{F} is *looping* iff for all $x \in W$, there exists a positive integer N and there exists a sequence (y_0, \dots, y_N) in W such that $y_0 = x$, $y_N = x$ and for all positive integers k , if $k \leq N$ then $y_{k-1}Ry_k$.

Remark that the properties of reflexivity, seriality and density are first-order definable whereas the properties of connectedness, non-2-colourability and looping are not first-order definable. Note also that the properties of reflexivity, seriality and density are determined in the ordinary language of modal logics by, respectively, the modal formulas $\Box p \rightarrow p$, $\langle \Box \rangle \top$ and $\Box \Box p \rightarrow \Box p$ whereas the properties of connectedness, non-2-colourability and looping are not determined in the ordinary language of modal logics.

PROPOSITION 4.1 ([6])

The following classes of frames are determined by the associated formulas: the class of all reflexive frames ($p \neq 0 \rightarrow C(p, p)$), the class of all serial frames ($p \neq 0 \rightarrow C(p, 1)$), the class of all dense frames ($C(p, q) \rightarrow C(p, r) \vee C(-r, q)$), the class of all connected frames ($p \neq 0 \wedge -p \neq 0 \rightarrow C(p, -p)$), the class of all non-2-colourable frames ($(p \cup q) \equiv 1 \wedge (p \cap q) \equiv 0 \rightarrow C(p, p) \vee C(q, q)$), the class of all looping frames ($(p \cap -q) \neq 0 \rightarrow C(p, -q) \vee C(q, -q)$).

In Proposition 4.1, we have seen that the language of Contact Logics is able to determine simple classes of frames as well as classes of frames that are not modally definable in the ordinary language of modal logics. Nevertheless, the language of Contact Logics is not able to determine some other simple classes of frames. Let $\mathcal{F} = (W, R)$ be a frame. We shall say that

- \mathcal{F} is *transitive* iff for all $x, y \in W$, if there exists $z \in W$ such that xRz and zRy then xRy ,
- \mathcal{F} is *Church–Rosser* iff for all $x, y, z \in W$, if xRy and xRz then there exists $t \in W$ such that yRt and zRt .

Remark that the properties of transitivity and Church–Rosser are first-order definable. Note also that the properties of transitivity and Church–Rosser are determined in the ordinary language of modal logics by, respectively, the modal formulas $\Box p \rightarrow \Box\Box p$ and $\Diamond\Box p \rightarrow \Box\Diamond p$.

PROPOSITION 4.2 ([6])

The following classes of frames are not determined by a set of formulas: the class of all transitive frames and the class of all Church–Rosser frames.

What do we learn from Propositions 4.1 and 4.2? We learn that the capacities of the language of Contact Logics and the ordinary language of modal logics are not comparable.

There are many reasons to study a logical formalism. For any semantically-based logical formalism, perhaps one of the most important reasons is to evaluate our capacity to decide the satisfiability problem. Now, we examine a technique that can be used as a decision procedure for the satisfiability problem in Contact Logics. This technique is based on a variant of the standard translation of the ordinary language of modal logics into a first-order language. By now, the reader should have noticed an important difference between the semantics of the language of Contact Logics and the semantics of the basic language of modal logics: in the semantics of Contact Logics, satisfaction is a binary relation between models and formulas whereas in the semantics of modal logics, satisfaction is a ternary relation between models, points and formulas [9, Definition 1.20]. Such a difference relates to the way we have defined the satisfiability of the formulas $C(a, b)$ and $a \equiv b$ in models. This way implies that in every model, the operators $[U]$ and $\langle U \rangle$ being interpreted by the universal binary relation on the set of all points and the operators \Box and \Diamond being interpreted by the binary relation R on the set of all points, $C(a, b)$ corresponds to $\langle U \rangle(a \wedge \Diamond b)$ and $a \equiv b$ corresponds to $[U](a \leftrightarrow b)$. The following translation of the contact language into a first-order language illustrates this correspondence. Let $\mathcal{L}^1(\mathbf{BV})$ be the first-order language with equality which has the unary predicates P_1, P_2, \dots corresponding to the variables p_1, p_2, \dots in \mathbf{BV} and the binary predicate R_C corresponding to the C construct. If u is a first-order variable and a is a term then the corresponding first-order formula $ST(u, a)$ in $\mathcal{L}^1(\mathbf{BV})$ is inductively defined as follows:

- $ST(u, p_n) = P_n(u)$,
- $ST(u, 0) = \perp$,
- $ST(u, \neg a) = \neg ST(u, a)$,
- $ST(u, a \cup b) = ST(u, a) \vee ST(u, b)$.

If φ is a formula then the corresponding first-order sentence $ST(\varphi)$ in $\mathcal{L}^1(\mathbf{BV})$ is inductively defined as follows:

- $ST(C(a, b)) = \exists u \exists v (ST(u, a) \wedge ST(v, b) \wedge R_C(u, v))$,
- $ST(a \equiv b) = \forall u (ST(u, a) \leftrightarrow ST(u, b))$,
- $ST(\perp) = \perp$,
- $ST(\neg \varphi) = \neg ST(\varphi)$,
- $ST(\varphi \vee \psi) = ST(\varphi) \vee ST(\psi)$.

PROPOSITION 4.3

Let $\mathcal{M} = (W, R, V)$ be a model. For all terms a , for all $x \in W$ and for all formulas φ , $x \in \bar{V}(a)$ iff $\mathcal{M} \models ST(u, a)[x]$ and $\mathcal{M} \models \varphi$ iff $\mathcal{M} \models ST(\varphi)$.

PROOF. By induction on a and φ . □

Remark that for all contact formulas φ , the first order sentence $ST(\varphi)$ belongs to the 2-variable fragment of $\mathcal{L}^1(\mathbf{BV})$. The decidability of the 2-variable fragment of any first-order language with equality has been obtained by Mortimer [23]. The membership in **NEXPTIME** of its satisfiability problem has been established by Grädel *et al.* [18]. Hence, the embedding of our language into $\mathcal{L}^1(\mathbf{BV})$ considered in Proposition 4.3 implies that

PROPOSITION 4.4

If \mathcal{C} is a class of frames definable by a first-order sentence with at most 2 variables then the following decision problem is decidable in nondeterministic exponential time:

input: a formula φ ,
output: determine whether φ is satisfiable in \mathcal{C} .

PROOF. Suppose \mathcal{C} is a class of frames definable by a first-order sentence with at most 2 variables. Let F be a first-order sentence with at most 2 variables defining \mathcal{C} . By Proposition 4.3, φ is satisfiable in \mathcal{C} iff the first-order sentence $F \wedge ST(\varphi)$ is satisfiable. Remark that the first-order sentence $F \wedge ST(\varphi)$ belongs to the 2-variable fragment of $\mathcal{L}^1(\mathbf{BV})$. Moreover, the size of $F \wedge ST(\varphi)$ is linear in the size of φ . Since the satisfiability problem of the 2-variable fragment is in **NEXPTIME**, therefore the above decision problem is in **NEXPTIME** too. □

Apart from the class of all frames, the following classes of frames are definable by a first-order sentence with at most 2 variables: the class of all serial frames, the class of all reflexive frames and the class of all symmetric frames. Nevertheless, there exist simple classes of frames that are not definable by a first-order sentence with at most 2 variables: the class of all transitive frames and the class of all dense frames. The class of all transitive frames may be not definable by a first-order sentence with at most 2 variables, it can still be proved that the above decision problem is in **NEXPTIME**. How? Simply by recalling the fact that the class of all transitive frames and the class of all frames validate the same formulas [7]. As for the class of all serial frames or the class of all dense frames, they do not validate the same formulas as the class of all frames, witness the formula $p \neq 0 \rightarrow C(p, 1)$ which is valid in the class of all serial frames but not valid in the class of all frames and the formula $C(p, q) \rightarrow C(p, r) \vee C(-r, q)$ which is valid in the class of all dense frames but not valid in the class of all frames. Nevertheless, by using a tableaux-based approach as described below, one can show that the above decision problem is in **PSPACE** when one considers the class of all serial frames and decidable when one considers the class of all dense frames.

5 Tableaux-based decision procedures

In this section we present tableaux-based decision procedures for the satisfiability problem in Contact Logics. In the relational semantics of Contact Logics, satisfiability is a binary relation defined between models and formulas. Moreover, the language of Contact Logics is based on two types of expressions: terms and formulas. For these reasons, using a countably infinite set of symbols

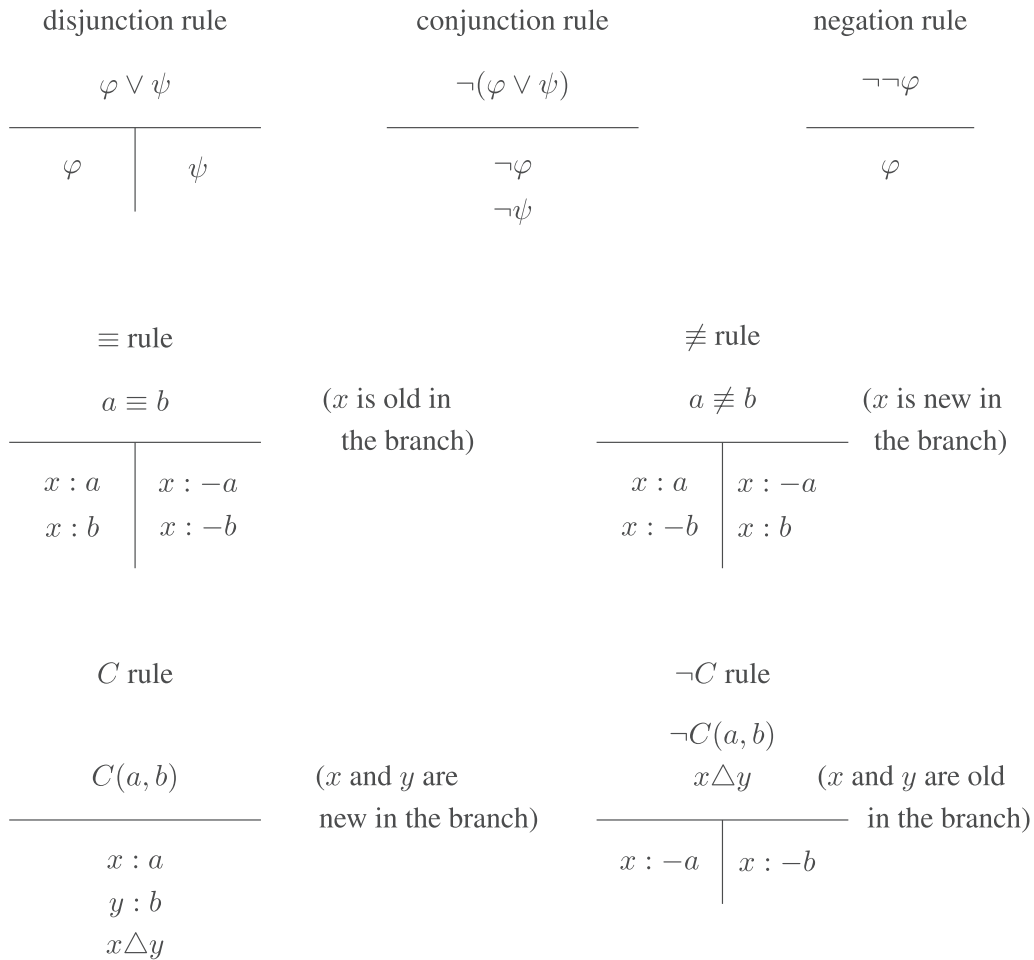


FIGURE 1. Formula rules.

(with members denoted x, y , etc), *tableaux* will be trees with nodes labeled by the following types of expressions:

- formulas,
- expressions of the form $x : a$ where x is a symbol and a is a term,
- expressions of the form $x \Delta y$ where x, y are symbols.

Given a formula φ , its *initial tableau* is the labeled tree consisting of exactly one node (called root) labeled with φ . Since tableaux are trees, therefore they have branches. A branch is said to be *closed* iff one of the following conditions holds:

- it contains a node labeled with $x : 0$,
- it contains two nodes respectively labeled with $x : a, x : -a$,
- it contains a node labeled with \perp .

We will say that a branch is *open* iff it is not closed. A tableau is *closed* when all its branches are closed. We will say that a tableau is *open* iff it is not closed. The *formula rules* and the *Boolean rules* are given in Figure 1 and Figure 2, respectively. Rules are applied in a standard way by extending

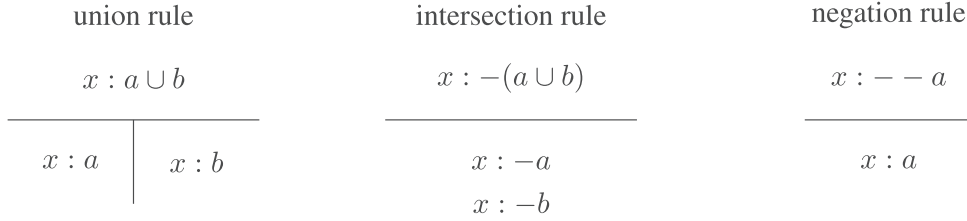


FIGURE 2. Boolean rules.

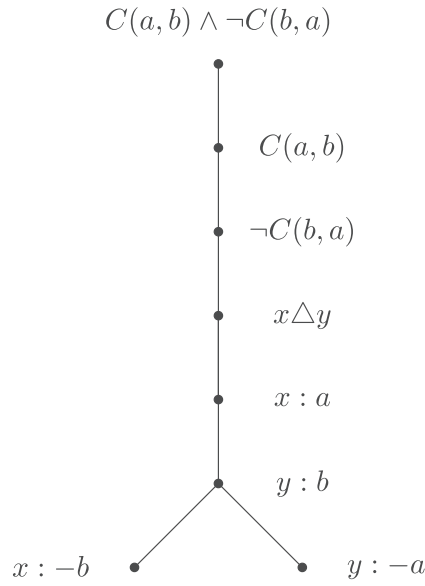


FIGURE 3. Example of an open tableau.

branches of constructed trees. For example, given a current tree t , a branch β in t and a node n in β labeled with the formula $C(a, b)$, applying the C rule to n consists in successively adding to the end of β three new nodes respectively labeled with $x : a$, $y : b$ and $x\Delta y$ where x, y are new symbols. Let us consider an example: the contact formula $C(a, b) \wedge \neg C(b, a)$ (Figure 3). The tableau obtained for this formula by applying our rules has two open branches.

In order to prove that the tableaux of satisfiable formulas cannot be closed, we introduce the concept of interpretability of a branch in a model. Let $\mathcal{M} = (W, R, V)$ be a model. Let β be a branch in a tableau and W' be the set of all variables occurring in β . The branch β is said to be *interpretable* in \mathcal{M} if there exists a function $f : W' \rightarrow W$ such that:

- for all φ occurring in β , $\mathcal{M} \models \varphi$,
- for all $x\Delta y$ occurring in β , $f(x)Rf(y)$,
- for all $x : a$ occurring in β , $f(x) \in \bar{V}(a)$.

These conditions are called compatibility conditions for f . Let t be a tableau and \mathcal{M} be a model. The tableau t is said to be *interpretable* in \mathcal{M} if and only if there exists a branch in t which is interpretable in \mathcal{M} . Obviously, interpretable branches and, then, interpretable tableaux are open.

Now, we show the soundness of the tableau rules for Contact Logics, i.e. we show that if a formula is satisfiable then its initial tableau cannot be transformed into a closed tableau by means of the rules.

We will also show that starting from the initial tableau of a given formula, after a finite number of steps, no tableau rule can be applied. Firstly, about the soundness.

PROPOSITION 5.1

Let $\mathcal{M} = (W, R, V)$ be a model and φ be a formula. If $\mathcal{M} \models \varphi$, then every tableau computed from the initial tableau of φ is interpretable in \mathcal{M} and is therefore open.

PROOF. Suppose $\mathcal{M} \models \varphi$. Since the initial tableau of φ consists of a single node labeled with φ , therefore the initial tableau of φ is interpretable in \mathcal{M} . The fact that the tableau rules preserve the interpretability property in \mathcal{M} follows from the strict similarity between the relational semantics of Contact Logics and the tableau rules presented in page 5. \square

Secondly, about the termination.

PROPOSITION 5.2

Let φ be a contact formula. After a finite number of steps from the initial tableau of φ , no tableau rule can be applied.

PROOF. Suppose there exists an infinite sequence t_0, t_1, \dots of tableaux such that t_0 is the initial tableau of φ and for all $n \in \mathbb{N}$, t_{n+1} is obtained from t_n by applying a rule. Remark that every node occurring in these tableaux has at most two successors. As a result, from the sequence t_0, t_1, \dots , we can extract an infinite branch β . The branch β contains information of the form $\psi, x\Delta y$ and $x : a$, where ψ is a sub-formula or the negation of a sub-formula of φ and a is a sub-term or the complement of a sub-term of φ . Note that the symbol x, y, \dots occurring in β has been introduced by two specific rules: the C rule and the \neq rule. Remark also that the application of these two rules is triggered by the occurrence of atomic formula of the form $C(a, b)$ and $a \neq b$. These atomic formulas being sub-formulas or negations of sub-formulas of φ , the above two specific rules will be applied finitely many times only. This means that, all in all, the infinite sequence t_0, t_1, \dots cannot exist. \square

We will say that a tableau is *saturated* when no tableau rule can be applied to it.

Now, we prove the completeness of the tableau method, i.e. for all valid formulas φ , after finitely many steps one can obtain a closed tableau from the initial tableau of $\neg\varphi$. Let t be a tableau and β be a branch in t . The *model for β* is the triple $\mathcal{M} = (W, R, V)$ where

- W is the set of all symbols occurring in β ,
- R is the binary relation on W defined by xRy iff β contains the information $x\Delta y$,
- $V(p)$ is the set of all $x \in W$ such that β contains the information $x : p$.

The following lemma is crucial for proving the completeness of our method.

LEMMA 5.3 (Truth Lemma)

Let t be a saturated tableau and β be an open branch in t . Let $\mathcal{M} = (W, R, V)$ be the model for β . We have the following:

- If β contains $x : a$, then $x \in \bar{V}(a)$,
- If β contains a contact formula φ , then $\mathcal{M} \models \varphi$.

PROOF. For the terms a , the proof is done by induction on a . The base case follows from the definition of V . The induction steps are left to the reader.

For the formulas φ , the proof is done by induction on φ . We only consider the case $C(a, b)$, the other cases being left to the reader. Suppose β contains the contact formula $C(a, b)$. Since t is saturated, β contains nodes labeled with $x\Delta y$, $x : a$ and $y : b$. By the first item of the Truth

Lemma, we have $x \in \bar{V}(a)$ and $y \in \bar{V}(b)$. Since β contains the information $x\Delta y$, then xRy . Therefore $M \models C(a, b)$. \square

Now, we are ready to prove the completeness of our method.

PROPOSITION 5.4

Let φ be a formula and t a saturated tableau obtained from the initial tableau of $\neg\varphi$. If φ is valid in the class of all models then t is closed.

PROOF. Suppose t is open. Thus, t contains an open branch β . Let $M = (W, R, V)$ be the model for β . By the Truth Lemma, we have $\mathcal{M} \models \neg\varphi$, contradicting the validity of φ . \square

Obviously, the formula $a \neq 0$ is satisfiable iff the term a is consistent in Boolean Logic. For this reason, the problem of determining if a given formula is satisfiable is **NP**-hard. A careful analysis of the tableau rules immediately leads us to the conclusion that the depth of a tableau computed from a given formula φ is linear in the number of symbols in φ . Since tableaux are finitely branching, for this reason, together with Propositions 5.1 and 5.2, Proposition 5.4 allows us to conclude that the problem of determining if a given formula is satisfiable in the class of all models is **NP**-complete. This complexity result has already been discussed in [7] where it was obtained by means of a more complicated argument based on the filtration method.

6 Variants

Let \mathcal{C}_{sym} , \mathcal{C}_{ref} , \mathcal{C}_{ser} and \mathcal{C}_{den} , respectively, be the class of all symmetric frames, the class of all reflexive frames, the class of all serial frames and the class of all dense frames. We extend our systems by adding new tableau rules which are Sym (for symmetric models), Ref (for reflexive models), Ser (for serial models) and Den (for dense models). Let us consider the class of all symmetric models. In order to decide satisfiability with respect to this class, we should add the following rule to our system:

$$(Sym) \quad \frac{x\Delta y}{y\Delta x} \quad (x \text{ and } y \text{ already occur in the branch}).$$

Obviously, (Sym) preserves the property of interpretability in symmetric models. Thus, if φ is satisfiable in a symmetric model, then all tableaux computed from the initial tableau of φ are open. Seeing that there is no need to apply (Sym) twice to the same pair (x, y) , termination of our extended tableau system easily follows. Thus, after a finite number of steps from an initial tableau, no rules can be applied. Finally, within the context of our extended system, the model defined in the previous section is clearly symmetric. The proof of the Truth Lemma being repeated as such, we therefore obtain the following:

PROPOSITION 6.1

Let φ be a formula and t a tableau obtained from the initial tableau of $\neg\varphi$ by applying the tableau rules augmented with (Sym). If φ is valid in the class of all symmetric models then t is closed.

Again, a careful analysis of the tableau rules augmented with (Sym) immediately leads us to the conclusion that the depth of a tableau computed from a given formula φ is linear in the number of symbols in φ . Consequently, the problem of determining if a given formula is satisfiable in the class of all symmetric models is **NP**-complete.

If we now consider the class of all reflexive models, the following rules should be used:

$$\text{(Ref)} \quad \frac{\bullet}{x\Delta x} \quad (x \text{ already occurs in the branch}).$$

The same line of reasoning as the one considered in the case of the class of all models would easily allow us to obtain the soundness, termination and completeness of the extended tableau system.

Let us now consider the class of all serial models. Apparently, the seriality property is quite innocent. Nevertheless, the satisfiability problem for the restriction of Contact Logic to the class of all serial models seems to be more difficult than expected. Since we are interested in the class of all serial models, the following rule is of some importance:

$$\text{(Ser)} \quad \frac{\bullet}{x\Delta y} \quad (x \text{ already occurs in the branch} \\ \text{and } y \text{ is new in the branch}).$$

Obviously, the application of (Ser) preserves the property of interpretability in serial models. The main problem with (Ser) clearly concerns the termination property. Nevertheless, by means of a strategy, we can obtain a terminating tableaux-based decision procedure. To define our strategy, we need the following preliminary definitions. Let β be a branch in some tableau and x be a symbol occurring in β . We will say that x is *successor-free*, iff there is no symbol y such that $x\Delta y$ occurs in β . Let $term(x, \beta)$ be the set of all terms which is associated with x . x is said to be *twin-free*, if there is no symbol y occurring in β , $x \neq y$, such that $term(x, \beta) = term(y, \beta)$ and y has a successor. Recall from the example presented in page 6 that the contact formula $C(a, b) \wedge \neg C(b, a)$ has an open tableau. According to our definition, the symbols x and y are twin-free in both branches. On the other hand, x is not successor-free: x has a successor, namely y , in both branches. Besides, y has no successor in both branches. In other respect, y is a successor-free symbol. In the general case, our strategy is the following:

- Apply the formula rules and term rules as much as possible,
- Choose a successor-free and twin-free symbol x already existing in the branch. Apply the (Ser) rule to x and go to the first item otherwise, go to the last item,
- Halt.

Now, let us show how our strategy will terminate. Remark first that in any branch β of a tableau constructed from φ 's initial tableau, $term(x, \beta)$ contains, for each symbol x occurring in β , only subterms or complements of subterms of φ . There exists finitely many subterms of φ . Consequently, at some point of the computation, in each branch β of the constructed tree, each β successor-free symbol is not β twin-free. Thus, our strategy terminates. Finally, we now prove the completeness of our strategy. Suppose β is an open branch (if there is at least one) in the tableau obtained by means of our strategy from the initial tableau of φ . Let $\mathcal{M} = (W, R, V)$ be the structure defined as follows:

- W is the set of all symbols x in β that are not successor-free,
- for all $x, y \in W$, xRy if and only if $x\Delta y$ occurs in β or there is a successor-free symbol z in β such that $term(y, \beta) = term(z, \beta)$ and $x\Delta z$ occurs in β (we will say that y is a *twin* of z).
- $V(p)$ is the set of all $x \in W$ such that β contains the information $x : p$.

Obviously, \mathcal{M} is serial. Moreover,

LEMMA 6.2

Let t be an open saturated tableau obtained by applying the tableau rules augmented with (Ser),

β be an open branch in t and $\mathcal{M} = (W, R, V)$ be the model defined as above. The following conditions hold:

- If β contains the expression $x : a$, then $x \in \bar{V}(a)$,
- If β contains the contact formula φ , then $\mathcal{M} \models \varphi$.

PROOF. The first condition is proved by induction on a . As for the second condition, we just consider the case of formula $C(a, b)$. Suppose β contains $C(a, b)$, we show that $\mathcal{M} \models C(a, b)$. Since t rule is saturated, β contains the following expressions: $x\Delta y, x : a$ and $y : b$. Then, $x \in W$. Since $x : a$ occurs in β , therefore by the first condition, $x \in \bar{V}(a)$. As for y , there are two cases to consider:

Case 1: y has a successor. Then, $y \in W$ and xRy . Since $y : b$ occurs in β , therefore by the first condition, $y \in \bar{V}(b)$. It follows that $\mathcal{M} \models C(a, b)$.

Case 2: y has no successor. Hence, $y \notin W$. Let z be a twin of y in β . By definition of the twin-free property, z has a successor, $z \in W$ and xRz . Moreover, $z : b$ belongs to the branch β . By the first condition, $z \in \bar{V}(b)$. Consequently, $\mathcal{M} \models C(a, b)$. \square

From all this, it follows that

PROPOSITION 6.3

Let φ be a formula and t a saturated tableau obtained from the initial tableau of $\neg\varphi$. If φ is valid in the class of all serial models then t is closed.

Concerning the computational complexity of deciding the satisfiability of formulas with respect to the class of all serial models, obviously, the number N of pairwise distinct sets $term(\cdot, \beta)$ associated to symbols in the branch β of a tableau computed from φ is exponential in the size of φ . Thus, we immediately obtain from the tableau approach for the class of all serial models the following result: the computational complexity of deciding the satisfiability of formulas is in **NEXPTIME**. Nevertheless, in a branch β , during a computation, there is no need, when all formulas of the form $C(a, b)$ and $a \not\equiv b$ have been taken into account, and we know that the number of such formulas is linearly bounded in the size of φ , to keep in memory all expressions of the form $x : a$ occurring in β . All we have to do is as follows: (1) when all formulas of the form $C(a, b)$ and $a \not\equiv b$ have been taken into account, each time (Ser) is applied, apply the $\neg C$ and the \equiv rules as much as possible, then eliminate from the branch the symbol x that has triggered the execution of (Ser); (2) count the number of times the rule (Ser) have been applied; (3) once this number is greater than N , stop. This would give us an improved strategy that can be implemented in polynomial space. Consequently, the computational complexity of deciding the satisfiability of formulas with respect to the class of all serial models is in **PSPACE**. This complexity result is new; it cannot be easily obtained by means of an argument based on the filtration method.

A model $\mathcal{M} = (W, R, V)$ is said to be dense if for all $x, y \in W$, if xRy then there exists $z \in W$ such that xRz and zRy . If we are interested in the class of all dense models, the following rule can be used:

$$\begin{array}{c}
 \text{(Den)} \quad \frac{x\Delta y}{\begin{array}{c} x\Delta z \\ z\Delta y \end{array}} \quad \begin{array}{l} (x \text{ and } y \text{ already occur in the branch} \\ \text{and } z \text{ is new in the branch}). \end{array}
 \end{array}$$

Obviously, (Den) is sound with respect to dense models, i.e. it preserves the interpretability property of tableaux. Nevertheless, as in the case of (Ser), it may lead to infinite computations. Thus,

we have to define a strategy that will guarantee the soundness, the completeness and the termination of our tableaux-based system. In a branch β of some tableau, we shall say that the pair (x, y) of symbols occurring in β is intermediate-free iff $x\Delta y$ occurs in β and for all symbols z in β , either $x\Delta z$ does not occur in β or $z\Delta y$ does not occur in β . Given a symbol x in β , as in the case of serial models, let $term(x, \beta) = \{a : x : a \text{ occurs in } \beta\}$. A pair (x, y) of symbols occurring in β is said to be twin-free if $x\Delta y$ occurs in β and for all symbols z_1, z_2, z_3 occurring in β , if $term(x, \beta) = term(z_1, \beta)$ and $term(y, \beta) = term(z_3, \beta)$ then either $z_1\Delta z_2$ does not occur in β , or $z_1\Delta z_3$ does not occur in β , or $z_2\Delta z_3$ does not occur in β . Our strategy is the following:

- Apply the formula rules and the term rules as much as possible,
- Choose an intermediate-free twin-free pair (x, y) of symbols occurring in some branch β ; apply the rule (Den) to (x, y) and go to (i) otherwise go to (iii),
- Halt.

In order to show that our strategy terminates, it suffices to follow an argument similar to the one developed in the case of serial models. Let us be more precise. Firstly, remark that in any branch β of a tableau constructed from the initial formula φ and for any x occurring in β , $term(x, \beta)$ only contains subterms or negation of subterms from φ . Seeing that there exist finitely many subterms from φ , at some point of the computation, each intermediate-free pair (x, y) of symbols occurring in a branch β is not twin-free. Therefore, our strategy terminates. Secondly, let us prove the soundness and the completeness of our tableau system extended with (Den) and following the above strategy. Obviously, every tableau constructed, by following the above strategy, from the initial tableau of a formula φ satisfiable in a dense model will be open. Conversely, suppose β is an open branch obtained, by means of our strategy, at the end of the tableau computation from an initial formula φ . Let W be the set of all x, y , etc occurring in β . As expected, we define on W the valuation V such that for all variables p , $V(p) = \{x \in W : x : p \text{ occurs in } \beta\}$. Now, for the accessibility relation R on W , it is defined as follows: for all $x, y \in W$, xRy iff $x\Delta y$ occurs in β . Defining $\mathcal{M} = (W, R, V)$, as in the cases of the previous classes of models that we have considered, we obtain the following lemma which proof is similar to the proof of Lemma 6.2.

LEMMA 6.4

- If $x : a$ occurs in β then $x \in \bar{V}(a)$.
- If φ occurs in β then $\mathcal{M} \models \varphi$.

Since the considered tableau has been computed from the initial tableau of φ , the branch β contains the formula φ and, by item (ii) of the above Lemma 2.3, $\mathcal{M} \models \varphi$. The main drawback is that R might be not dense. Suppose R is not dense. In order to prove that φ can be satisfied in a dense model, we have to transform \mathcal{M} into a modally equivalent dense model \mathcal{M}' . Clearly, the accessibility relation R can be seen as the set of all pairs (x, y) in $W \times W$ such that $x\Delta y$ occurs in β . Since R is not dense, therefore R , as a subset of $W \times W$, is nonempty. A pair (x, y) in R is said to be an R -defect if there is no $z \in W$ such that xRz and zRy . By definition of R , seeing that β is a branch in a tableau that has been constructed by following the strategy described in the previous page, each R -defect (x, y) can be associated to a triple $\delta(x, y) = (z_1, z_2, z_3)$ of elements of W such that $term(x, \beta) = term(z_1, \beta)$, $term(y, \beta) = term(z_3, \beta)$ and $z_1\Delta z_2, z_1\Delta z_3$ and $z_2\Delta z_3$ occur in β . Let R_0, R_1, \dots be the sequence of binary relation on W and $\delta_0, \delta_1, \dots$ be the sequence of functions defined as follows.

1. Let $R_0 = R$ and $\delta_0 = \delta$. By definition, for each R_0 -defect (x, y) , the triple $\delta_0(x, y) = (z_1, z_2, z_3)$ of elements of W is such that $term(x, \beta) = term(z_1, \beta)$, $term(y, \beta) = term(z_3, \beta)$ and $z_1R_0z_2$, $z_1R_0z_3$ and $z_2R_0z_3$.

2. Suppose for some $i \in \mathbb{N}$ we have already defined a binary relation R_i on W and a function δ_i associating, for each R_i -defect (x, y) , a triple $\delta_i(x, y) = (z_1, z_2, z_3)$ of elements of W such that $term(x, \beta) = term(z_1, \beta)$, $term(y, \beta) = term(z_3, \beta)$ and $z_1 R_i z_2$, $z_1 R_i z_3$ and $z_2 R_i z_3$.
3. If R_i has no defect, we define $R_{i+1} = R_i$. Otherwise, let (x, y) be an R_i -defect. Let z_1, z_2, z_3 be elements of W such that $\delta_i(x, y) = (z_1, z_2, z_3)$. We define $R_{i+1} = R_i \cup \{(x, z_2), (z_2, y)\}$.

For all $i \in \mathbb{N}$, let $\mathcal{M}_i = (W, R_i, V)$. Remark that, according to step 3 above, the only differences between \mathcal{M}_{i+1} and \mathcal{M}_i are the possibly new links (x, z_2) and (z_2, y) . Since, according to step 3, the elements z_1, z_2, z_3 of W are such that $\delta_i(x, y) = (z_1, z_2, z_3)$, then $term(x, \beta) = term(z_1, \beta)$, $term(y, \beta) = term(z_3, \beta)$, $z_1 R_i z_2$ and $z_2 R_i z_3$. Thus, for all formula ψ based on variables occurring in φ , $\mathcal{M}_{i+1} \models \psi$ iff $\mathcal{M}_i \models \psi$. Since $\mathcal{M} \models \varphi$, therefore for all $i \in \mathbb{N}$, $\mathcal{M}_i \models \varphi$. Since W is finite, there exists $i \in \mathbb{N}$ such that R_i has no defect. Hence, R_i is a dense relation on W . Let $\mathcal{M}' = (W, R_i, V)$. By the above discussion, $\mathcal{M}' \models \varphi$. Hence, φ can be satisfied in a dense model and we obtain the following result.

PROPOSITION 6.5

Let φ be a formula and t a saturated tableau obtained from the initial tableau of $\neg\varphi$. If φ is valid in the class of all dense models then t is closed.

Thus, the problem of the satisfiability of formulas with respect to the class of all dense models is decidable. Unfortunately, we do not know its exact computational complexity. Note that this decidability result is new; it seems that it cannot be easily obtained by means of an argument based on the filtration method, seeing that the filtration construction does not preserve the elementary property of density.

7 Unifiability

A *substitution* is a function $\sigma : \mathbf{BV} \longrightarrow TER(\mathbf{BV})$ which moves at most finitely many variables. Given a substitution σ , let $\bar{\sigma} : TER(\mathbf{BV}) \cup FOR(\mathbf{BV}) \longrightarrow TER(\mathbf{BV}) \cup FOR(\mathbf{BV})$ be the endomorphism such that for all variables p , $\bar{\sigma}(p) = \sigma(p)$. The *composition* of the substitutions σ and τ is the substitution $\sigma \circ \tau$ such that for all $p \in \mathbf{BV}$, $(\sigma \circ \tau)(p) = \bar{\sigma}(\tau(p))$. Let \mathcal{C} be a class of frames. We say that a substitution σ is \mathcal{C} -*equivalent* to a substitution τ (in symbols $\sigma \simeq_{\mathcal{C}} \tau$) iff for all variables p , $\sigma(p) \equiv \tau(p)$ is \mathcal{C} -valid. Obviously,

PROPOSITION 7.1

The binary relation $\simeq_{\mathcal{C}}$ is reflexive, symmetric and transitive on the set of all substitutions.

We say that a substitution σ is *more \mathcal{C} -general than* a substitution τ (in symbols $\sigma \preceq_{\mathcal{C}} \tau$) iff there exists a substitution ν such that $\sigma \circ \nu \simeq_{\mathcal{C}} \tau$. Obviously,

PROPOSITION 7.2

The binary relation $\preceq_{\mathcal{C}}$ is reflexive and transitive on the set of all substitutions. Moreover, it contains $\simeq_{\mathcal{C}}$.

We say that a finite set $\{(\varphi_1, \psi_1), \dots, (\varphi_n, \psi_n)\}$ of pairs of formulas is \mathcal{C} -*unifiable* iff there exists a substitution σ such that $\bar{\sigma}(\varphi_1) \leftrightarrow \bar{\sigma}(\psi_1), \dots, \bar{\sigma}(\varphi_n) \leftrightarrow \bar{\sigma}(\psi_n)$ are \mathcal{C} -valid. As a consequence of the classical interpretation of the constructs for formulas, this is equivalent to $\bar{\sigma}((\varphi_1 \leftrightarrow \psi_1) \wedge \dots \wedge (\varphi_n \leftrightarrow \psi_n))$ is \mathcal{C} -valid. This means that we can restrict our attention to a simpler kind of unifiability problems consisting of exactly one formula. We say that a formula φ is \mathcal{C} -*unifiable* iff there exists a substitution σ such that $\bar{\sigma}(\varphi)$ is \mathcal{C} -valid. In that case, we say that σ is a \mathcal{C} -*unifier* of φ . For instance,

$0 \equiv p \vee 1 \equiv p$ is unifiable in \mathcal{C}_{all} , \mathcal{C}_{ind} and \mathcal{C}_{con} . As we will prove it with Proposition 11.3, its unifiers are the substitutions σ such that considered as a formula in Classical Propositional Logic (**CPL**), $\sigma(p)$ is either equivalent to 0 or equivalent to 1. The *elementary \mathcal{C} -unifiability problem* consists in determining whether a given formula is \mathcal{C} -unifiable. See [1, 15, 16] for an introduction to the unifiability problem in modal and description logics. We say that a set of \mathcal{C} -unifiers of a formula φ is *complete* iff for all \mathcal{C} -unifiers σ of φ , there exists a \mathcal{C} -unifier τ of φ in that set such that $\tau \preceq_{\mathcal{C}} \sigma$. As we will prove it with Proposition 11.3, the substitutions σ_0 and σ_1 such that $\sigma_0(p) = 0$, $\sigma_1(p) = 1$ and for all variables q , if $p \neq q$ then $\sigma_0(q) = q$ and $\sigma_1(q) = q$ constitute a complete set of \mathcal{C} -unifiers of $0 \equiv p \vee 1 \equiv p$. If a formula is \mathcal{C} -unifiable then an important question is as follows: does it have a minimal complete set of \mathcal{C} -unifiers? If the answer is ‘yes’, how large is this set? We say that

- a \mathcal{C} -unifiable formula φ is *\mathcal{C} -finitary* iff there exists a finite minimal complete set of \mathcal{C} -unifiers of φ but there exists no with cardinality 1,
- a \mathcal{C} -unifiable formula φ is *\mathcal{C} -unitary* iff there exists a minimal complete set of \mathcal{C} -unifiers of φ with cardinality 1.

We say that

- unification in \mathcal{C} is *finitary* iff every \mathcal{C} -unifiable formula is either \mathcal{C} -finitary or \mathcal{C} -unitary and there exists a \mathcal{C} -finitary formula,
- unification in \mathcal{C} is *unitary* iff every \mathcal{C} -unifiable formula is \mathcal{C} -unitary.

See [12] for an introduction to the unification types in logics.

8 Monomials and polynomials

Before we provide, in Section 10, computability results about unifiability in Contact Logics and prove, in Section 11, that elementary unification is finitary, we introduce the notions of monomial and polynomial (this section) and define some equivalence relations (next section). Let k, n be a nonnegative integer. An *n -monomial* is a term of the form

$$\bullet p_1^{\beta_1} \cap \dots \cap p_n^{\beta_n}$$

where $(\beta_1, \dots, \beta_n) \in \{0, 1\}^n$. Considering the terms $p_1^{\beta_1}, \dots, p_n^{\beta_n}$ as literals in **CPL**, n -monomials are just conjunctions of literals. Considering a term a in $TER(p_1, \dots, p_n)$ as a formula in **CPL**, let $mon(n, a)$ be the set of all n -monomials $p_1^{\beta_1} \cap \dots \cap p_n^{\beta_n}$ such that a is a tautological consequence of $p_1^{\beta_1} \cap \dots \cap p_n^{\beta_n}$. An *n -polynomial* is a term of the form

$$\bullet (p_1^{\beta_{11}} \cap \dots \cap p_n^{\beta_{1n}}) \cup \dots \cup (p_1^{\beta_{m1}} \cap \dots \cap p_n^{\beta_{mn}})$$

where m is a nonnegative integer and $(\beta_{11}, \dots, \beta_{1n}), \dots, (\beta_{m1}, \dots, \beta_{mn}) \in \{0, 1\}^n$. Considering the terms $p_1^{\beta_{11}} \cap \dots \cap p_n^{\beta_{1n}}, \dots, p_1^{\beta_{m1}} \cap \dots \cap p_n^{\beta_{mn}}$ as conjunctions of literals in **CPL**, n -polynomials are just disjunctive normal forms. Note that for all terms a in $TER(p_1, \dots, p_n)$, $\bigcup mon(n, a)$ is an n -polynomial. For all positive integers i , if $i \leq n$ then let $\pi_i : \{0, 1\}^n \rightarrow \{0, 1\}$ be the function such that for all $(\beta_1, \dots, \beta_n) \in \{0, 1\}^n$, $\pi_i(\beta_1, \dots, \beta_n) = \beta_i$. Let $f : \{0, 1\}^k \rightarrow \{0, 1\}^n$ be a function. For all $(\beta_1, \dots, \beta_n) \in \{0, 1\}^n$, we define

$$\bullet f^{-1}(\beta_1, \dots, \beta_n) = \{(\alpha_1, \dots, \alpha_k) \in \{0, 1\}^k : f(\alpha_1, \dots, \alpha_k) = (\beta_1, \dots, \beta_n)\}.$$

Obviously, for all $(\beta_1, \dots, \beta_n) \in \{0, 1\}^n$, $f^{-1}(\beta_1, \dots, \beta_n) \subseteq \{0, 1\}^k$. For all positive integers i , if $i \leq n$ then we define

- $\Delta_i = \{(\alpha_1, \dots, \alpha_k) \in \{0, 1\}^k : \pi_i(f(\alpha_1, \dots, \alpha_k)) = 1\}$,
- $c_i = \bigcup \{p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} : (\alpha_1, \dots, \alpha_k) \in \Delta_i\}$.

Obviously, for all positive integers i , if $i \leq n$ then $\Delta_i \subseteq \{0, 1\}^k$ and c_i is a k -polynomial. Note that Δ_i and c_i depend on f too. Lemma 8.1 is an immediate consequence of the definition of $mon(n, a)$.

LEMMA 8.1

Let $a(p_1, \dots, p_n) \in TER(p_1, \dots, p_n)$. Considered as formulas in **CPL**, the terms a and $\bigcup mon(n, a)$ are equivalent.

More difficult is the proof of Proposition 8.2.

PROPOSITION 8.2

For all $(\beta_1, \dots, \beta_n) \in \{0, 1\}^n$, considered as formulas in **CPL**, the terms $\bigcup \{p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} : (\alpha_1, \dots, \alpha_k) \in f^{-1}(\beta_1, \dots, \beta_n)\}$ and $c_1^{\beta_1} \cap \dots \cap c_n^{\beta_n}$ are equivalent.

PROOF. Let $(\beta_1, \dots, \beta_n) \in \{0, 1\}^n$. It suffices to show that considered as formulas in **CPL**, for all $\theta_1, \dots, \theta_k \in \{0, 1\}$, if p_1 is interpreted by θ_1, \dots, p_k is interpreted by θ_k then $\bigcup \{p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} : (\alpha_1, \dots, \alpha_k) \in f^{-1}(\beta_1, \dots, \beta_n)\}$ is equivalent to 1 iff $c_1^{\beta_1} \cap \dots \cap c_n^{\beta_n}$ is equivalent to 1. Let $\theta_1, \dots, \theta_k \in \{0, 1\}$. Let p_1 be interpreted by θ_1, \dots, p_k be interpreted by θ_k .

Suppose $\bigcup \{p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} : (\alpha_1, \dots, \alpha_k) \in f^{-1}(\beta_1, \dots, \beta_n)\}$ is equivalent to 1. Hence, $(\theta_1, \dots, \theta_k) \in f^{-1}(\beta_1, \dots, \beta_n)$. Thus, $f(\theta_1, \dots, \theta_k) = (\beta_1, \dots, \beta_n)$. For the sake of the contradiction, suppose $c_1^{\beta_1} \cap \dots \cap c_n^{\beta_n}$ is equivalent to 0. Let i be a positive integer such that $i \leq n$ and $c_i^{\beta_i}$ is equivalent to 0. Since either $\beta_i = 0$ or $\beta_i = 1$, therefore we have to consider two cases. In the former case, $\beta_i = 0$ and therefore $\bigcup \{p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} : (\alpha_1, \dots, \alpha_k) \in \Delta_i\}$ is equivalent to 1. Consequently, $(\theta_1, \dots, \theta_k) \in \Delta_i$. Hence, $\pi_i(f(\theta_1, \dots, \theta_k)) = 1$. Since $f(\theta_1, \dots, \theta_k) = (\beta_1, \dots, \beta_n)$, therefore $\beta_i = 1$: a contradiction. In the latter case, $\beta_i = 1$ and therefore $\bigcup \{p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} : (\alpha_1, \dots, \alpha_k) \in \Delta_i\}$ is equivalent to 0. Thus, $(\theta_1, \dots, \theta_k) \notin \Delta_i$. Hence, $\pi_i(f(\theta_1, \dots, \theta_k)) = 0$. Since $f(\theta_1, \dots, \theta_k) = (\beta_1, \dots, \beta_n)$, therefore $\beta_i = 0$: a contradiction.

Suppose $c_1^{\beta_1} \cap \dots \cap c_n^{\beta_n}$ is equivalent to 1. Let i be an arbitrary positive integer such that $i \leq n$. Since $c_1^{\beta_1} \cap \dots \cap c_n^{\beta_n}$ is equivalent to 1, therefore $c_i^{\beta_i}$ is equivalent to 1. Since either $\beta_i = 0$ or $\beta_i = 1$, therefore we have to consider two cases. In the former case, $\beta_i = 0$ and therefore c_i is equivalent to 0. Hence, $(\theta_1, \dots, \theta_k) \notin \Delta_i$. Thus, $\pi_i(f(\theta_1, \dots, \theta_k)) = 0$. Since $\beta_i = 0$, therefore $\pi_i(f(\theta_1, \dots, \theta_k)) = \beta_i$. In the latter case, $\beta_i = 1$ and therefore c_i is equivalent to 1. Consequently, $(\theta_1, \dots, \theta_k) \in \Delta_i$. Hence, $\pi_i(f(\theta_1, \dots, \theta_k)) = 1$. Since $\beta_i = 1$, therefore $\pi_i(f(\theta_1, \dots, \theta_k)) = \beta_i$. In both cases, $\pi_i(f(\theta_1, \dots, \theta_k)) = \beta_i$. Since i was arbitrary, therefore $f(\theta_1, \dots, \theta_k) = (\beta_1, \dots, \beta_n)$. Thus, $(\theta_1, \dots, \theta_k) \in f^{-1}(\beta_1, \dots, \beta_n)$. Consequently, $\bigcup \{p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} : (\alpha_1, \dots, \alpha_k) \in f^{-1}(\beta_1, \dots, \beta_n)\}$ is equivalent to 1. \square

9 Some equivalence relations

Let k, n be nonnegative integers and \mathcal{C} be a class of frames. Given $(a_1, \dots, a_n) \in TER(p_1, \dots, p_k)^n$, we define on $\{0, 1\}^k$ the equivalence relation $\sim_{(a_1, \dots, a_n)}^k$ as follows:

- $(\alpha_1, \dots, \alpha_k) \sim_{(a_1, \dots, a_n)}^k (\alpha'_1, \dots, \alpha'_k)$ iff for all positive integers i , if $i \leq n$, then $p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} \in mon(k, a_i)$ iff $p_1^{\alpha'_1} \cap \dots \cap p_k^{\alpha'_k} \in mon(k, a_i)$.

Lemma 9.1 is a consequence of its definition.

LEMMA 9.1

For all $(a_1, \dots, a_n) \in TER(p_1, \dots, p_k)^n$, $\sim_{(a_1, \dots, a_n)}^k$ has at most 2^n equivalence classes on $\{0, 1\}^k$.

From now on, we will always assume that $k \geq n$. Let $f : \{0, 1\}^k \rightarrow \{0, 1\}^n$ be a surjective function such that for all $(\alpha_1, \dots, \alpha_k), (\alpha'_1, \dots, \alpha'_k) \in \{0, 1\}^k$, if $f(\alpha_1, \dots, \alpha_k) = f(\alpha'_1, \dots, \alpha'_k)$ then $(\alpha_1, \dots, \alpha_k) \sim_{(a_1, \dots, a_n)}^k (\alpha'_1, \dots, \alpha'_k)$. By means of the function f , we define the n -tuple (b_1, \dots, b_n) of n -polynomials as follows:

- $b_i = \bigcup \{p_1^{\beta_1} \cap \dots \cap p_n^{\beta_n} : p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} \in \text{mon}(k, a_i) \text{ and } f(\alpha_1, \dots, \alpha_k) = (\beta_1, \dots, \beta_n)\}$.

We say that (b_1, \dots, b_n) is the n -tuple of n -polynomials *properly obtained* from the given n -tuple (a_1, \dots, a_n) in $TER(p_1, \dots, p_k)^n$ with respect to (k, n) . Lemma 9.2 is a consequence of its definition.

LEMMA 9.2

Let (a_1, \dots, a_n) be an n -tuple in $TER(p_1, \dots, p_k)^n$ and (b_1, \dots, b_n) be an n -tuple of n -polynomials. Let W be a nonempty set. If (b_1, \dots, b_n) is properly obtained from (a_1, \dots, a_n) with respect to (k, n) then

- for all valuations V on W , there exists a valuation V' on W such that for all positive integers i , if $i \leq n$, then $\bar{V}(a_i) = \bar{V}'(b_i)$,
- for all valuations V on W , there exists a valuation V' on W such that for all positive integers i , if $i \leq n$, then $\bar{V}'(b_i) = \bar{V}(a_i)$.

For all $(\beta_1, \dots, \beta_n) \in \{0, 1\}^n$, let $f^{-1}(\beta_1, \dots, \beta_n)$ be as in Section 8. For all positive integers i , if $i \leq n$ then let Δ_i and c_i be as in Section 8. Let ν be the substitution such that

- for all positive integers i , if $i \leq n$ then $\nu(p_i) = c_i$,
- for all variables q , if $q \notin \{p_1, \dots, p_n\}$ then $\nu(q) = q$.

PROPOSITION 9.3

For all positive integers i , if $i \leq n$ then considered as formulas in **CPL**, the terms a_i and $\bar{\nu}(b_i)$ are equivalent.

PROOF. Let i be a positive integer such that $i \leq n$. Considered as formulas in **CPL**, the following terms are equivalent:

1. $\bar{\nu}(b_i)$.
2. $\bigcup \{c_1^{\beta_1} \cap \dots \cap c_n^{\beta_n} : p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} \in \text{mon}(k, a_i) \text{ and } f(\alpha_1, \dots, \alpha_k) = (\beta_1, \dots, \beta_n)\}$.
3. $\bigcup \{\bigcup \{p_1^{\alpha'_1} \cap \dots \cap p_k^{\alpha'_k} : (\alpha'_1, \dots, \alpha'_k) \in f^{-1}(\beta_1, \dots, \beta_n)\} : p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} \in \text{mon}(k, a_i) \text{ and } f(\alpha_1, \dots, \alpha_k) = (\beta_1, \dots, \beta_n)\}$.
4. $\bigcup \{p_1^{\alpha'_1} \cap \dots \cap p_k^{\alpha'_k} : (\alpha'_1, \dots, \alpha'_k) \in f^{-1}(\beta_1, \dots, \beta_n), p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} \in \text{mon}(k, a_i) \text{ and } f(\alpha_1, \dots, \alpha_k) = (\beta_1, \dots, \beta_n)\}$.
5. $\bigcup \{p_1^{\alpha'_1} \cap \dots \cap p_k^{\alpha'_k} : p_1^{\alpha_1} \cap \dots \cap p_k^{\alpha_k} \in \text{mon}(k, a_i) \text{ and } f(\alpha'_1, \dots, \alpha'_k) = f(\alpha_1, \dots, \alpha_k)\}$.
6. $\bigcup \text{mon}(k, a_i)$.
7. a_i .

The equivalence between 1 and 2 is a consequence of the definition of ν ; the equivalence between 2 and 3 is a consequence of Proposition 8.2; the equivalences between 3, 4 and 5 are consequences of simple set-theoretic properties; the equivalence between 5 and 6 is a consequence of the definition

of $\sim_{(a_1, \dots, a_n)}^k$ and the fact that for all $(\alpha_1, \dots, \alpha_k), (\alpha'_1, \dots, \alpha'_k) \in \{0, 1\}^k$, if $f(\alpha_1, \dots, \alpha_k) = f(\alpha'_1, \dots, \alpha'_k)$ then $(\alpha_1, \dots, \alpha_k) \sim_{(a_1, \dots, a_n)}^k (\alpha'_1, \dots, \alpha'_k)$; the equivalence between 6 and 7 is a consequence of Lemma 8.1. \square

We define on $FOR(p_1, \dots, p_n)$ the equivalence relation $\equiv_{\mathcal{C}}^n$ as follows:

- $\varphi \equiv_{\mathcal{C}}^n \psi$ iff $\varphi \leftrightarrow \psi$ is \mathcal{C} -valid.

PROPOSITION 9.4

$\equiv_{\mathcal{C}}^n$ has finitely many equivalence classes on $FOR(p_1, \dots, p_n)$.

PROOF. Each formula φ in $FOR(p_1, \dots, p_n)$ is a combination of formulas of the form $P(a, b)$ where a and b are terms in $TER(p_1, \dots, p_n)$ and P is one of the 8 binary predicates of **RCC8**. Hence, $\equiv_{\mathcal{C}}^n$ has finitely many equivalence classes on $FOR(p_1, \dots, p_n)$. \square

Let A_n be the set of all n -tuples of terms. Note that n -tuples of terms in A_n may contain occurrences of variables outside $\{p_1, \dots, p_n\}$. Given a model (W, R, V) on a frame in \mathcal{C} and $(a_1, \dots, a_n) \in A_n$, let $\Phi_{(a_1, \dots, a_n)}^{(W, R, V)}$ be the set of all equational formulas $\varphi(p_1, \dots, p_n)$ in $FOR(p_1, \dots, p_n)$ such that $(W, R, V) \models \varphi(a_1, \dots, a_n)$. Consider a complete list of representatives for each equivalence class on $\Phi_{(a_1, \dots, a_n)}^{(W, R, V)}$ modulo $\equiv_{\mathcal{C}}^n$ and let $\varphi_{(a_1, \dots, a_n)}^{(W, R, V)}(p_1, \dots, p_n)$ be their conjunction.

We define on A_n the equivalence relation $\cong_{\mathcal{C}}^n$ as follows:

- $(a_1, \dots, a_n) \cong_{\mathcal{C}}^n (b_1, \dots, b_n)$ iff for all formulas $\varphi(p_1, \dots, p_n)$ in $FOR(p_1, \dots, p_n)$, $\varphi(a_1, \dots, a_n)$ is \mathcal{C} -valid iff $\varphi(b_1, \dots, b_n)$ is \mathcal{C} -valid.

Now, we define on A_n the equivalence relation $\simeq_{\mathcal{C}}^n$ as follows:

- $(a_1, \dots, a_n) \simeq_{\mathcal{C}}^n (b_1, \dots, b_n)$ iff for all equational formulas $\varphi(p_1, \dots, p_n)$ in $FOR(p_1, \dots, p_n)$, $\varphi(a_1, \dots, a_n)$ is \mathcal{C} -valid iff $\varphi(b_1, \dots, b_n)$ is \mathcal{C} -valid.

Obviously, $\cong_{\mathcal{C}}^n$ is finer than $\simeq_{\mathcal{C}}^n$. Below, Lemma 9.5 is a consequence of Proposition 9.4, Lemma 9.6 is a consequence of Lemma 9.5, Lemma 9.7 is a consequence of the definition of $\simeq_{\mathcal{C}}^n$ and Lemma 9.2, Lemma 9.8 is a consequence of Lemma 9.7 and Lemma 9.9 is a consequence of the definition of $\varphi_{(a_1, \dots, a_n)}^{(W, R, V)}(p_1, \dots, p_n)$.

LEMMA 9.5

$\cong_{\mathcal{C}}^n$ has finitely many equivalence classes on A_n .

LEMMA 9.6

$\simeq_{\mathcal{C}}^n$ has finitely many equivalence classes on A_n .

LEMMA 9.7

Let (a_1, \dots, a_n) be an n -tuple in $TER(p_1, \dots, p_k)^n$ and (b_1, \dots, b_n) be an n -tuple of n -polynomials. If (b_1, \dots, b_n) is properly obtained from (a_1, \dots, a_n) with respect to (k, n) then $(a_1, \dots, a_n) \simeq_{\mathcal{C}}^n (b_1, \dots, b_n)$.

LEMMA 9.8

$TER(p_1, \dots, p_n)^n$ constitutes a complete set of representatives for each equivalence class on A_n modulo $\simeq_{\mathcal{C}}^n$.

LEMMA 9.9

Let (W, R, V) be a model on a frame in \mathcal{C} and $(a_1, \dots, a_n) \in A_n$. $(W, R, V) \models \varphi_{(a_1, \dots, a_n)}^{(W, R, V)}(a_1, \dots, a_n)$.

For the last results of this section, we have to consider a special kind of models: the balanced ones. A model $\mathcal{M} = (W, R, V)$ is *balanced* iff for all terms a , $\bar{V}(a) = \emptyset$ or $\bar{V}(a) = W$ or $\bar{V}(a)$ is infinite and coinfinite. We say that the class \mathcal{C} of frames is *balanced* iff for all formulas φ , if φ is \mathcal{C} -satisfiable then there exists a countable frame (W, R) in \mathcal{C} and there exists a balanced valuation V on W such that $(W, R, V) \models \varphi$. We should stress here that many natural classes of frames are balanced.

PROPOSITION 9.10

The following classes of frames are balanced: \mathcal{C}_{all} , \mathcal{C}_{sym} , \mathcal{C}_{ref} , \mathcal{C}_{ser} , \mathcal{C}_{den} , \mathcal{C}_{ind} and \mathcal{C}_{con} .

PROOF. Let \mathcal{C} be one of the above-mentioned classes of frames. In order to demonstrate that \mathcal{C} is balanced, let φ be a \mathcal{C} -satisfiable formula. By [7, Theorem 4.2], φ is satisfiable in a finite frame of \mathcal{C} . Let (W, R) be a finite frame of \mathcal{C} and V be a valuation on it such that $(W, R, V) \models \varphi$. Let $W' = W \times \mathbb{N}$ and R' be the binary relation on W' defined by $(x, i)R'(y, j)$ iff xRy . Since the frame (W, R) is finite, therefore the frame (W', R') is countable. Let V' be the valuation on (W', R') defined by $V'(p) = V(p) \times \mathbb{N}$ for each Boolean variable p . As the reader may easily verify by induction, $\bar{V}'(a) = \bar{V}(a) \times \mathbb{N}$ for each term a . It follows that the model (W', R', V') is balanced. As the reader may easily verify by induction, $(W, R, V) \models \psi$ iff $(W', R', V') \models \psi$ for each formula ψ . Since $(W, R, V) \models \varphi$, therefore $(W', R', V') \models \varphi$. Finally, it suffices now to remark that if (W, R) is symmetric (respectively, reflexive, serial, dense, indiscrete, connected) then (W', R') is symmetric (respectively, reflexive, serial, dense, indiscrete, connected) too. \square

The interest to consider balanced classes of frames lies in the following:

PROPOSITION 9.11

If \mathcal{C} is balanced then for all $(a_1, \dots, a_n), (b_1, \dots, b_n) \in A_n$, if $(a_1, \dots, a_n) \simeq_{\mathcal{C}}^n (b_1, \dots, b_n)$ then $(a_1, \dots, a_n) \cong_{\mathcal{C}}^n (b_1, \dots, b_n)$.

PROOF. Suppose \mathcal{C} is balanced. Let $(a_1, \dots, a_n), (b_1, \dots, b_n) \in A_n$ be such that $(a_1, \dots, a_n) \simeq_{\mathcal{C}}^n (b_1, \dots, b_n)$ and $(a_1, \dots, a_n) \not\cong_{\mathcal{C}}^n (b_1, \dots, b_n)$. Let $\varphi(p_1, \dots, p_n)$ be a formula in $FOR(p_1, \dots, p_n)$ such that $\varphi(a_1, \dots, a_n)$ is \mathcal{C} -valid not-iff $\varphi(b_1, \dots, b_n)$ is \mathcal{C} -valid. Without loss of generality, let us assume that $\varphi(a_1, \dots, a_n)$ is \mathcal{C} -valid and $\varphi(b_1, \dots, b_n)$ is not \mathcal{C} -valid. Since \mathcal{C} is balanced, therefore let (W, R, V) be a balanced model on a countable frame in \mathcal{C} such that $(W, R, V) \not\models \varphi(b_1, \dots, b_n)$. By Lemma 9.9, $(W, R, V) \models \varphi_{(b_1, \dots, b_n)}^{(W, R, V)}(b_1, \dots, b_n)$. Hence, $\neg \varphi_{(b_1, \dots, b_n)}^{(W, R, V)}(b_1, \dots, b_n)$ is not \mathcal{C} -valid. Since $(a_1, \dots, a_n) \simeq_{\mathcal{C}}^n (b_1, \dots, b_n)$, therefore $\neg \varphi_{(b_1, \dots, b_n)}^{(W, R, V)}(a_1, \dots, a_n)$ is not \mathcal{C} -valid. Since \mathcal{C} is balanced, therefore let (W', R', V') be a balanced model on a countable frame in \mathcal{C} such that $(W', R', V') \models \varphi_{(b_1, \dots, b_n)}^{(W, R, V)}(a_1, \dots, a_n)$. Now, consider $(\beta_1, \dots, \beta_n) \in \{0, 1\}^n$. If $\bar{V}(b_1^{\beta_1} \cap \dots \cap b_n^{\beta_n}) = \emptyset$ then $(W, R, V) \models b_1^{\beta_1} \cap \dots \cap b_n^{\beta_n} \equiv 0$. Thus, $\varphi_{(b_1, \dots, b_n)}^{(W, R, V)}(p_1, \dots, p_n) \rightarrow p_1^{\beta_1} \cap \dots \cap p_n^{\beta_n} \equiv 0$ is \mathcal{C} -valid. Since $(W', R', V') \models \varphi_{(b_1, \dots, b_n)}^{(W, R, V)}(a_1, \dots, a_n)$, therefore $(W', R', V') \models a_1^{\beta_1} \cap \dots \cap a_n^{\beta_n} \equiv 0$. Consequently, $\bar{V}'(a_1^{\beta_1} \cap \dots \cap a_n^{\beta_n}) = \emptyset$. Similarly, the reader may easily verify that if $\bar{V}(b_1^{\beta_1} \cap \dots \cap b_n^{\beta_n}) = W$ then $\bar{V}'(a_1^{\beta_1} \cap \dots \cap a_n^{\beta_n}) = W'$ and if $\bar{V}(b_1^{\beta_1} \cap \dots \cap b_n^{\beta_n})$ is infinite and coinfinite then $\bar{V}'(a_1^{\beta_1} \cap \dots \cap a_n^{\beta_n})$ is infinite and coinfinite. In all cases, there exists a bijection $g_{(\beta_1, \dots, \beta_n)}$ from $\bar{V}(b_1^{\beta_1} \cap \dots \cap b_n^{\beta_n})$ to $\bar{V}'(a_1^{\beta_1} \cap \dots \cap a_n^{\beta_n})$. Let g be the union of all $g_{(\beta_1, \dots, \beta_n)}$ when $(\beta_1, \dots, \beta_n)$ describes $\{0, 1\}^n$. The reader may easily verify that g is a bijection from W to W' such that for all $u \in W$ and for all $(\beta_1, \dots, \beta_n) \in \{0, 1\}^n$, $u \in \bar{V}(b_1^{\beta_1} \cap \dots \cap b_n^{\beta_n})$ iff $g(u) \in \bar{V}'(a_1^{\beta_1} \cap \dots \cap a_n^{\beta_n})$. Let R'_g be the binary relation on W' defined by $u'R'_g v'$ iff $g^{-1}(u')Rg^{-1}(v')$. Obviously, g is an isomorphism from (W, R) to (W', R'_g) . Since $\varphi(a_1, \dots, a_n)$ is \mathcal{C} -valid, therefore $(W', R'_g, V') \models \varphi(a_1, \dots, a_n)$. Hence, $(W, R, V) \models \varphi(b_1, \dots, b_n)$: a contradiction. \square

Lemma 9.12 is a consequence of the definition of the equational formula $\psi_{(a_1, \dots, a_n)}(p_1, \dots, p_n)$ and Lemma 9.9.

LEMMA 9.12

Let $(a_1, \dots, a_n) \in A_n$. $\psi_{(a_1, \dots, a_n)}(a_1, \dots, a_n)$ is \mathcal{C} -valid.

Lemma 9.13 is a consequence of the definition of the equational formula $\psi_{(a_1, \dots, a_n)}(p_1, \dots, p_n)$ and Lemma 9.12.

LEMMA 9.13

Let $(a_1, \dots, a_n) \in A_n$. For all equational formulas $\varphi(p_1, \dots, p_n)$ in $FOR(p_1, \dots, p_n)$, $\varphi(a_1, \dots, a_n)$ is \mathcal{C} -valid iff $\psi_{(a_1, \dots, a_n)}(p_1, \dots, p_n) \rightarrow \varphi(p_1, \dots, p_n)$ is \mathcal{C} -valid.

Lemma 9.14 is a consequence of the definition of the equivalence relation $\simeq_{\mathcal{C}}^n$ and Lemmas 9.12 and 9.13.

LEMMA 9.14

Let $(a_1, \dots, a_n), (b_1, \dots, b_n) \in A_n$. $(a_1, \dots, a_n) \simeq_{\mathcal{C}}^n (b_1, \dots, b_n)$ iff $\psi_{(a_1, \dots, a_n)}(p_1, \dots, p_n) \leftrightarrow \psi_{(b_1, \dots, b_n)}(p_1, \dots, p_n)$ is \mathcal{C} -valid.

10 Computability of unifiability

Let \mathcal{C} be a class of frames. Lemma 10.1 is a consequence of the definitions in Section 3.

LEMMA 10.1

1. For all $a \in TER(\emptyset)$, either $a \equiv 0$ is \mathcal{C} -valid or $a \equiv 1$ is \mathcal{C} -valid. Moreover, the formula in $\{a \equiv 0, a \equiv 1\}$ that is \mathcal{C} -valid can be computed in linear time.
2. For all $a, b \in TER(\emptyset)$, either $DC(a, b)$ is \mathcal{C} -valid or $a \equiv b$ is \mathcal{C} -valid. Moreover, the formula in $\{DC(a, b), a \equiv b\}$ that is \mathcal{C} -valid can be computed in linear time.

Lemma 10.2 is a consequence of the definition of unifiability.

LEMMA 10.2

For all formulas $\varphi(p_1, \dots, p_n)$, φ is \mathcal{C} -unifiable iff there exists $a_1, \dots, a_n \in TER(\emptyset)$ such that $\varphi(a_1, \dots, a_n)$ is \mathcal{C} -valid.

PROPOSITION 10.3

The elementary \mathcal{C} -unifiability problem is in **NP**.

PROOF. By Lemmas 10.1 and 10.2, for all formulas $\varphi(p_1, \dots, p_n)$, $\varphi(p_1, \dots, p_n)$ is \mathcal{C} -unifiable iff there exists $a_1, \dots, a_n \in \{0, 1\}$ such that $\varphi(a_1, \dots, a_n)$ is \mathcal{C} -valid. Obviously, this can be decided in polynomial time. \square

PROPOSITION 10.4

Let $a(p_1, \dots, p_n)$ be a term. $a(p_1, \dots, p_n) \equiv 1$ is \mathcal{C} -unifiable iff considered as a formula in **CPL**, $a(p_1, \dots, p_n)$ is satisfiable.

PROOF. Suppose considered as a formula in **CPL**, $a(p_1, \dots, p_n)$ is satisfiable. Let b_1, \dots, b_n in $\{0, 1\}$ be such that $a(b_1, \dots, b_n)$ is a tautology. Hence, $a(b_1, \dots, b_n) \equiv 1$ is \mathcal{C} -valid. Thus, $a(p_1, \dots, p_n) \equiv 1$ is \mathcal{C} -unifiable.

Suppose $a(p_1, \dots, p_n) \equiv 1$ is \mathcal{C} -unifiable. By Lemmas 10.1 and 10.2, let b_1, \dots, b_n in $\{0, 1\}$ be such that $a(b_1, \dots, b_n) \equiv 1$ is \mathcal{C} -valid. Consequently, $a(b_1, \dots, b_n)$ is a tautology. Hence, considered as a formula in **CPL**, $a(p_1, \dots, p_n)$ is satisfiable. \square

PROPOSITION 10.5

The elementary \mathcal{C} -unifiability problem is **NP**-hard.

PROOF. By Proposition 10.4 and the **NP**-hardness of the satisfiability problem of formulas in **CPL**. \square

It follows from Propositions 10.3 and 10.5 that

PROPOSITION 10.6

The elementary unifiability problem in $\mathcal{C}_{all}, \mathcal{C}_{sym}, \mathcal{C}_{ref}, \mathcal{C}_{ser}, \mathcal{C}_{den}, \mathcal{C}_{ind}$ and \mathcal{C}_{con} is **NP**-complete.

11 Unification type

Let \mathcal{C} be a class of frames. We say that \mathcal{C} *agrees with unions* iff for all disjoint frames $(W, R), (W', R')$ in \mathcal{C} , there exists a frame (W'', R'') in \mathcal{C} such that $W \cup W' = W''$. We should stress here that many natural classes of frames agree with unions.

PROPOSITION 11.1

The following classes of frames agree with unions: $\mathcal{C}_{all}, \mathcal{C}_{sym}, \mathcal{C}_{ref}, \mathcal{C}_{ser}, \mathcal{C}_{den}, \mathcal{C}_{ind}$ and \mathcal{C}_{con} .

PROOF. Left to the reader. \square

The interest to consider classes of frames that agree with unions lies in the following:

PROPOSITION 11.2

If \mathcal{C} agrees with unions then $0 \equiv p \vee 1 \equiv p$ is not \mathcal{C} -unitary.

PROOF. Suppose \mathcal{C} agrees with unions and $0 \equiv p \vee 1 \equiv p$ is \mathcal{C} -unitary. Let σ_0 and σ_1 be substitutions such that $\sigma_0(p) = 0$ and $\sigma_1(p) = 1$. Obviously, σ_0 and σ_1 are \mathcal{C} -unifiers of $0 \equiv p \vee 1 \equiv p$. Since $0 \equiv p \vee 1 \equiv p$ is \mathcal{C} -unitary, therefore let τ be a \mathcal{C} -unifier of $0 \equiv p \vee 1 \equiv p$ such that $\tau \preceq_{\mathcal{C}} \sigma_0$ and $\tau \preceq_{\mathcal{C}} \sigma_1$. Let μ, ν be substitutions such that $\tau \circ \mu \simeq_{\mathcal{C}} \sigma_0$ and $\tau \circ \nu \simeq_{\mathcal{C}} \sigma_1$. Hence, $\bar{\mu}(\tau(p)) \equiv 0$ is \mathcal{C} -valid and $\bar{\nu}(\tau(p)) \equiv 1$ is \mathcal{C} -valid. Thus, neither $0 \equiv \tau(p)$ is \mathcal{C} -valid nor $1 \equiv \tau(p)$ is \mathcal{C} -valid. Let (W, R) and (W', R') be disjoint frames in \mathcal{C} , V be a valuation on W and V' be a valuation on W' such that neither $\bar{V}(\tau(p)) = \emptyset$ nor $\bar{V}'(\tau(p)) = W'$. Since \mathcal{C} agrees with unions, therefore let (W'', R'') be a frame in \mathcal{C} such that $W \cup W' = W''$. Let V'' be the valuation on W'' such that for all variables q , $V''(q) = V(q) \cup V'(q)$. Obviously, for all terms a , $\bar{V}''(a) = \bar{V}(a) \cup \bar{V}'(a)$. Since neither $\bar{V}(\tau(p)) = \emptyset$ nor $\bar{V}'(\tau(p)) = W'$, therefore neither $\bar{V}''(\tau(p)) = \emptyset$ nor $\bar{V}''(\tau(p)) = W''$. Consequently, τ is not a \mathcal{C} -unifier of $0 \equiv p \vee 1 \equiv p$: a contradiction. \square

PROPOSITION 11.3

If \mathcal{C} agrees with unions then the substitutions σ_0 and σ_1 such that $\sigma_0(p) = 0$, $\sigma_1(p) = 1$ and for all variables q , if $p \neq q$ then $\sigma_0(q) = q$ and $\sigma_1(q) = q$ constitute a complete set of \mathcal{C} -unifiers of $0 \equiv p \vee 1 \equiv p$. Moreover, $0 \equiv p \vee 1 \equiv p$ is \mathcal{C} -finitary.

PROOF. Suppose \mathcal{C} agrees with unions. Hence, by Proposition 11.2, $0 \equiv p \vee 1 \equiv p$ is not \mathcal{C} -unitary. Obviously, σ_0 and σ_1 are \mathcal{C} -unifiers of $0 \equiv p \vee 1 \equiv p$. Let τ be an arbitrary \mathcal{C} -unifier of $0 \equiv p \vee 1 \equiv p$ such that neither $\sigma_0 \preceq_{\mathcal{C}} \tau$ nor $\sigma_1 \preceq_{\mathcal{C}} \tau$. Thus, neither $0 \equiv \tau(p)$ is \mathcal{C} -valid nor $1 \equiv \tau(p)$ is \mathcal{C} -valid. Following the same line of reasoning as in the proof of Proposition 11.2, we conclude τ is not a \mathcal{C} -unifier of $0 \equiv p \vee 1 \equiv p$: a contradiction. Since τ was arbitrary, therefore σ_0 and σ_1 constitute a

complete set of \mathcal{C} -unifiers of $0 \equiv p \vee 1 \equiv p$. Consequently, $0 \equiv p \vee 1 \equiv p$ is either \mathcal{C} -unitary or \mathcal{C} -finitary. Since $0 \equiv p \vee 1 \equiv p$ is not \mathcal{C} -unitary, therefore $0 \equiv p \vee 1 \equiv p$ is \mathcal{C} -finitary. \square

PROPOSITION 11.4

If \mathcal{C} is balanced then elementary unification in \mathcal{C} is either finitary or unitary. Moreover, if \mathcal{C} agrees with unions then elementary unification in \mathcal{C} is finitary.

PROOF. Suppose \mathcal{C} is balanced. Let $\varphi(p_1, \dots, p_n)$ be an arbitrary \mathcal{C} -unifiable formula. Let σ be an arbitrary substitution such that $\bar{\sigma}(\varphi)$ is \mathcal{C} -valid. Without loss of generality, we can assume that for all variables q , if $q \notin \{p_1, \dots, p_n\}$ then $\sigma(q) = q$. Let k be a nonnegative integer and $(a_1, \dots, a_n) \in TER(p_1, \dots, p_k)^n$ be such that for all positive integers i , if $i \leq n$ then $\sigma(p_i) = a_i$. Since $\bar{\sigma}(\varphi)$ is \mathcal{C} -valid, therefore $\varphi(a_1, \dots, a_n)$ is \mathcal{C} -valid. Let $\sim_{(a_1, \dots, a_n)}^k, f : \{0, 1\}^k \rightarrow \{0, 1\}^n$ and (b_1, \dots, b_n) be as in Section 9. By Lemma 9.7, $(a_1, \dots, a_n) \simeq_{\mathcal{C}}^n (b_1, \dots, b_n)$. Since \mathcal{C} is balanced, therefore by Proposition 9.11, $(a_1, \dots, a_n) \cong_{\mathcal{C}}^n (b_1, \dots, b_n)$. Let τ be the substitution such that for all positive integers i , if $i \leq n$ then $\tau(p_i) = b_i$ and for all variables q , if $q \notin \{p_1, \dots, p_n\}$ then $\tau(q) = q$. Note that $(\tau(p_1), \dots, \tau(p_n)) \in TER(p_1, \dots, p_n)^n$. Moreover, since $\varphi(a_1, \dots, a_n)$ is \mathcal{C} -valid and $(a_1, \dots, a_n) \cong_{\mathcal{C}}^n (b_1, \dots, b_n)$, therefore $\varphi(b_1, \dots, b_n)$ is \mathcal{C} -valid. Hence, τ is a \mathcal{C} -unifier of φ . For all positive integers i , if $i \leq n$ then let Δ_i and c_i be as in Section 8. Let ν be as in Section 9. By Proposition 9.3, for all positive integers i , if $i \leq n$ then considered as formulas in **CPL**, the terms a_i and $\bar{\nu}(b_i)$ are equivalent. Thus, for all positive integers i , if $i \leq n$ then $\bar{\nu}(\tau(p_i)) \equiv \sigma(p_i)$ is \mathcal{C} -valid. Consequently, $\tau \circ \nu \simeq_{\mathcal{C}} \sigma$. Hence, $\tau \preceq_{\mathcal{C}} \sigma$. Since σ was arbitrary and $(\tau(p_1), \dots, \tau(p_n)) \in TER(p_1, \dots, p_n)^n$, therefore φ is either \mathcal{C} -finitary or \mathcal{C} -unitary. Since φ was arbitrary, therefore elementary unification in \mathcal{C} is either finitary or unitary. Now, suppose \mathcal{C} agrees with unions. By Proposition 11.3, elementary unification in \mathcal{C} is not unitary. Since elementary unification in \mathcal{C} is either finitary or unitary, therefore elementary unification in \mathcal{C} is finitary. \square

It follows from the above discussion that elementary unification in $\mathcal{C}_{all}, \mathcal{C}_{sym}, \mathcal{C}_{ref}, \mathcal{C}_{ser}, \mathcal{C}_{den}, \mathcal{C}_{ind}$ and \mathcal{C}_{con} is finitary.

12 Conclusion

We anticipate a number of further investigations.

In Section 4, we provided classes of formulas defining first-order or second-order conditions on frames. For pointers to this line of work in the basic modal language, see Goldblatt and Thomason [17]. The characterisation of the first-order definable classes of frames that are, in the sense of the Goldblatt–Thomason theorem for the basic modal language, modally definable in our language is still open. A Goldblatt–Thomason theorem for our language is still to be obtained.

In modal logics, Sahlqvist formulas are modal formulas with remarkable properties [25, 26]: the Sahlqvist correspondence theorem says that every Sahlqvist formula corresponds to a first-order definable class of frames; the Sahlqvist completeness theorem says that when Sahlqvist formulas are used as axioms in a normal logic, the logic is complete with respect to the elementary class of frames the axioms define. Then, in the end, a natural question is to ask whether a Sahlqvist-like theory—i.e. a theory that identifies a set of formulas that correspond to first-order definable classes of frames and that define logics complete with respect to the elementary classes of frames they correspond to—can be elaborated for our language. A first answer to this question has been presented in [5].

Concerning decidability, we have proved in Section 4 that if \mathcal{C} is a class of frames determined by a finite set of formulas then the satisfiability problem in \mathcal{C} is **EXPSpace**. In [8], we have proved that there exist classes \mathcal{C} of frames such that the satisfiability problem in \mathcal{C} is **NP**-complete and

there exist classes \mathcal{C} of frames such that the satisfiability problem in \mathcal{C} is **PSPACE**-complete. Does there exist classes \mathcal{C} of frames such that the satisfiability problem in \mathcal{C} is **EXPTIME**-complete? **NEXPTIME**-complete?

Concerning unifiability, what becomes of the unification type when the language is extended by the connectedness predicate considered in [20, 21]? And when the language is interpreted in different Euclidean spaces as in [20, 22]? In other respect, it remains to see how decision procedures for unifiability can be used to improve the performance of algorithms that handle the satisfiability problem. Finally, one may as well consider these questions when the language is extended by a set of propositional constants.

Acknowledgements

We make a point of thanking Tinko Tinchev and Dimiter Vakarelov for many stimulating discussions in the field of Contact Logics. We also would like to thank the referees for the feedback we have obtained from them.

References

- [1] F. Baader and S. Ghilardi. Unification in modal and description logics. *Logic Journal of the IGPL*, **19**, 705–730, 2011.
- [2] F. Baader, S. Borgwardt, and B. Morawska. Extending unification in \mathcal{EL} towards general TBoxes. In *Principles of Knowledge Representation and Reasoning*, pp. 568–572. AAAI Press, 2012.
- [3] F. Baader and P. Narendran. Unification of concept terms in description logics. *Journal of Symbolic Computation*, **31**, 277–305, 2001.
- [4] P. Balbiani and S. Kikot. Sahlqvist theorems for precontact logics. In *Advances in Modal Logic*, vol. 9, pp. 55–70. College Publications, 2012.
- [5] P. Balbiani and T. Tinchev. Definability and canonicity for Boolean logic with a binary relation. *Fundamenta Informaticæ*, **129**, 301–327, 2014.
- [6] P. Balbiani, T. Tinchev, and D. Vakarelov. Modal logics for region-based theories of space. *Fundamenta Informaticæ*, **81**, 29–82, 2007.
- [7] P. Balbiani, T. Tinchev, and D. Vakarelov. Dynamic logics of the region-based theory of discrete spaces. *Journal of Applied Non-Classical Logics*, **17**, 39–61, 2007.
- [8] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
- [9] G. Dimov and D. Vakarelov. Contact algebras and region-based theory of space: a proximity approach — I. *Fundamenta Informaticæ*, **74**, 209–249, 2006.
- [10] G. Dimov and D. Vakarelov. Contact algebras and region-based theory of space: proximity approach — II. *Fundamenta Informaticæ*, **74**, 251–282, 2006.
- [11] W. Dzik. *Unification Types in Logic*. Wydawnictwo Uniwersytetu Śląskiego, 2007.
- [12] A. Galton. The mereotopology of g spaces. In *Spatial Information Theory, Proceedings of the International Conference COSIT'99*, pp. 251–266. Springer, 1999.
- [13] A. Galton. *Qualitative Spatial Change*. Oxford University Press, 2000.
- [14] Ç. Gencer and D. de Jongh. Unifiability in extensions of $K4$. *Logic Journal of the IGPL*, **17**, 159–172, 2009.
- [15] S. Ghilardi. Best solving modal equations. *Annals of Pure and Applied Logic*, **102**, 183–198, 2000.

- [16] R. Goldblatt and S. Thomason. Axiomatic classes in propositional modal logic. In *Algebra and Logic*, pp. 163–173. Springer, 1975.
- [17] E. Grädel, P. Kolaitis, and M. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, **3**, 53–69, 1997.
- [18] R. Iemhoff. On the admissible rules of intuitionistic propositional logic. *The Journal of Symbolic Logic*, **66**, 281–294, 2001.
- [19] R. Kontchakov, Y. Nenov, I. Pratt-Hartmann, and M. Zakharyashev. Topological logics with connectedness over Euclidean spaces. *ACM Transactions on Computational Logic*, **14**, DOI:10.1145/2480759.2480765, 2013.
- [20] R. Kontchakov, I. Pratt-Hartmann, F. Wolter, and M. Zakharyashev. Spatial logics with connectedness predicates. *Logical Methods in Computer Science*, **6**, 1–43, 2010.
- [21] R. Kontchakov, I. Pratt-Hartmann, and M. Zakharyashev. Interpreting topological logics over Euclidean spaces. In *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*, pp. 534–544. AAAI Press, 2010.
- [22] M. Mortimer. On languages with two variables. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, **21**, 135–140, 1975.
- [23] V. Rybakov. *Admissibility of Logical Inference Rules*. Elsevier, 1997.
- [24] H. Sahlqvist. Completeness and correspondence in the first and second order semantics for modal logic. In *Proceedings of the Third Scandinavian Logic Symposium*, pp. 110–143. North-Holland, 1975.
- [25] G. Sambin and V. Vaccaro. A new proof of Sahlqvist theorem on modal definability and completeness. *The Journal of Symbolic Logic*, **54**, 992–999, 1989.
- [26] D. Vakarelov. Region-based theory of space: algebras of regions, representation theory, and logics. In *Mathematical Problems from Applied Logic*, pp. 267–348. Logics for the XXIst Century. II. Springer, 2007.
- [27] F. Wolter and M. Zakharyashev. Spatial representation and reasoning in RCC-8 with Boolean region terms. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pp. 244–248. IOS Press, 2000.