A Topological Constraint Language with Component Counting

Ian Pratt-Hartmann

Department of Computer Science,
University of Manchester, U.K.
ipratt@cs.man.ac.uk

ABSTRACT. *A topological constraint language is a formal language whose variables range over certain subsets of topological spaces, and whose nonlogical primitives are interpreted as topological relations and functions taking these subsets as arguments. Thus, topological constraint languages typically allow us to make assertions such as "region V_1 touches the boundary of region V_2", "region V_3 is connected" or "region V_4 is a proper part of the closure of region V_5". A formula ϕ in a topological constraint language is said to be satisfiable if there exists an assignment to its variables of regions from some topological space under which ϕ is made true. This paper introduces a topological constraint language which, in addition to the usual mechanisms for expressing Boolean combinations of regions and their topological closures, includes primitives for bounding the number of components of a region. We call this language $\mathcal{TCC}$, a rough acronym for "topological constraint language with component counting". (Thus, $\mathcal{TCC}$ extends earlier topological constraint languages based on the so-called RCC-primitives.) Our main result is that the problem of determining the satisfiability of a $\mathcal{TCC}$-formula is NEXPTIME-complete.*

RÉSUMÉ. *To be provded*

KEYWORDS: *Topology, Constraints, Spatial Reasoning, Computational Complexity*

MOTS-CLÉS : *To be provided*

1. Introduction

A topological constraint language is a formal language whose variables range over certain subsets of topological spaces—henceforth called *regions*—and whose nonlogical primitives are interpreted as topological relations and functions taking regions as arguments. Thus, topological constraint languages typically allow us to make assertions such as "region V_1 touches the boundary of region V_2", "region V_3 is connected" or "region V_4 is a proper part of the closure of region V_5". A formula ϕ in a topological language is said to be satisfiable if there exists an assignment to its variables of regions from some topological space under which ϕ is made true. A central question regarding any topological constraint language is to determine the computational complexity of deciding whether a given formula in that language is satisfiable.

Region-based topological languages trace their origins to Laguna [LAG 22] and Whitehead [WHI 29], and were later revived by Clarke [CLA 85, CLA 81]. For our purposes however, the most convenient point of departure is the development of the so-called RCC theory of Randell, Cui and Cohn [RAN 92]. This theory introduced the collections of primitive topological relations now known as RCC-5 and RCC-8, which have attracted continued attention ever since. Thus, Renz and Nebel [REN 99], show that satisfiability for a topological constraint language based on the RCC-8 primitives is NP-complete, and Jonsson and Drakengren [JON 97] obtain corresponding results for a similar constraint language based on the RCC-5 primitives. More recently, work by Nutt [NUT 99] and Wolter and Zakharyaschev [WOL 00a, WOL 00b] has treated more expressive topological constraint languages. For example, Nutt's language contains statements about the equality or inequality of complex terms denoting arbitrary Boolean combinations of regions and their topological closures, yielding a satisfiability problem which is PSPACE-complete. The present paper continues this trend towards topological constraint languages of greater expressive power. Specifically, we define a language which allows us to place bounds on the number of components (maximal connected subsets) possessed by any region. We call this language $\mathcal{TCC}$, a rough acronym for "topological constraint language with component counting". Our main result is that the satisfiability problem for $\mathcal{TCC}$ is NEXPTIME-complete.

The plan of the paper is as follows. Section 2 reviews the the notational conventions and background material assumed in this paper. Section 3 defines the language $\mathcal{TCC}$ and states our main result. Sections 4 and 5 are devoted to the proof of this result. Section 6 relates the work reported here to the development of topological constraint languages more generally.

2. Technical preliminaries

This section reviews the the notational conventions and background material assumed in this paper. Basic facts about topology have been included to render the paper more self-contained; readers familiar with this material may skip to the next section.

If A is any set, we denote the power set of A by $\mathbb{P}(A)$. If $B \subseteq A$, we denote the complement of B in A, namely $A \setminus B$, by $-B$, provided that the embedding set A is clear from context. A *topological space* is a pair $\langle A, \mathcal{O} \rangle$ where $\mathcal{O} \subseteq \mathbb{P}(A)$ satisfying the following properties: (i) $\emptyset \in \mathcal{O}$, (ii) $A \in \mathcal{O}$, (iii) $\mathcal{O}$ is closed under arbitrary unions, (iv) $\mathcal{O}$ is closed under finite intersections. The elements of $\mathcal{O}$ are called the *open* subsets of X. Any set $B \subseteq A$ such that $-B \in \mathcal{O}$ is called a *closed* subset of A. It is usual to identify the topological space $\langle A, \mathcal{O} \rangle$ with its *carrier set* A, leaving the designation of the open subsets implicit.

Let A be a topological space. If $B \subseteq A$, we say that the *interior* of B (relative to A), denoted B^{0_A}, is the largest open subset of A included in B, and that the *closure* of B (relative to A), denoted B^{-_A}, is the smallest closed subset of A which includes B. Since the set of opens is closed under arbitrary unions, it follows that interiors and closures always exist. When the embedding topological space A is clear from context, we omit subscripts, writing B^0 instead of B^{0_A} and B^- instead of B^{-_A}. Note that a set $B \subseteq A$ is open if and only if $B = B^0$ and closed if and only if $B = B^-$. The following routine observation will be used below:

Observation 1. *Let A be a topological space and B, C subsets of A with C open. If $B^- \cap C \neq \emptyset$, then $B \cap C \neq \emptyset$.*

Of particular interest will be the topological space arising from a graph. Let G be a set and $\rightarrow$ a binary relation on G. Take the open subsets of G to be those sets H satisfying the property that $v \rightarrow v'$ and $v' \in H$ implies $v \in H$. It is easy to check that this is indeed a topological space, and that a subset H of G is closed in this topology if and only if it satisfies the property that $v \rightarrow v'$ and $v \in H$ implies $v' \in H$. If, in addition, the relation $\rightarrow$ is reflexive and transitive, then the closure operator $\cdot^{-_G}$ satisfies, for any $H \subseteq G$,

$$H^{-_G} \quad = \quad \{v' \in G \mid v \rightarrow v' \text{ for some } v \in H\}.$$

If $\langle A, \mathcal{O} \rangle$ is a topological space and $B \subseteq A$, we can regard B as a topological space with open sets $\{O \cap B | O \in \mathcal{O}\}$. In this case, we say that B (as a topological space) has the *subspace topology* induced by A. The space A is said to be *connected* if it is not the union of two nonempty, disjoint, closed (equivalently: open) subsets A_1 and A_2. A subset B of A is said to be *connected* (in the space A) if it is a connected space under the subspace topology. Equivalently, B is connected if and only if there do not exist $B_1 \subseteq A$ and $B_2 \subseteq A$ such that: (i) $B_1 \cap B \neq \emptyset$ and $B_2 \cap B \neq \emptyset$; (ii) $B \subseteq B_1 \cup B_2$; and (iii) ${B_1}^- \cap {B_2}^- \cap B = \emptyset$. Intuitively, connected sets should be thought of as consisting of 'one piece'. In particular, for a topological space arising from a graph, a subset H turns out to be connected if and only if any two points in H are connected by a finite path (ignoring the directions of the edges) which does not stray outside H. The following observation follows easily from the definitions just given.

Observation 2. *Let A be a topological space with $B \subseteq A$, $C \subseteq A$. If B and C are connected with $B \cap C \neq \emptyset$, then $B \cup C$ is connected.*

If A is a topological space and $B \subseteq A$, a *component* of B is a maximal connected subset of B. If $C \subseteq B$ is nonempty, then C is always included within a unique component of B. Every set has at least one component; the empty set is the only component of itself; and all components of a nonempty set are nonempty. A set is connected if and only if it has exactly one component. Of course, the notion of component is relative to the assumed topology. The following routine observations will be used below:

Observation 3. *Let A be a topological space, and suppose $C \subseteq B \subseteq A$. Then, taking B to have the subspace topology, we have $C^{-_B} = B \cap C^{-_A}$.*

Observation 4. *Let A be a topological space, and suppose $C \subseteq B \subseteq A$. Then, taking B to have the subspace topology, C has the same components in B as it does in A.*

Let A and X be topological spaces, and let $f : A \to X$ be any function. We say that f is *continuous* if, for any subset Y of X such that Y is open (in the space X), $f^{-1}(Y)$ is open (in the space A). This is equivalent to the condition that, for any subset Y of X such that Y is closed, $f^{-1}(Y)$ is closed. It is an easy exercise to show that, if $B \subseteq A$ is connected (in the space A) and $f : A \to X$ is continuous, then $f(A) \subseteq X$ is also connected (in the space X). More generally, if B has at most k components and f is continuous, then $f(B)$ has at most k components (though it may certainly have fewer).

3. The Main Result

We begin with the syntax of our topological constraint language $\mathcal{TCC}$.

Definition 1. Let $\mathbf{V}$ be some fixed countable set. We refer to the elements of $\mathbf{V}$ as *variables*. The set of *terms* is defined inductively as follows:

1) every variable is a term;

2) if T is a term, then so are $-T$ and T^-;

3) if T and T' are terms, then so is $T \cap T'$.

An *atomic formula* is an expression of either of the forms:

1) $T = T'$, where T, T' are terms

2) $c^{\leq k}(T)$, where T is a term and k is a binary numeral.

A *formula* is a Boolean combination of atomic formulas. The language $\mathcal{TCC}$ is the set of formulas. If $\phi \in \mathcal{TCC}$, we take the *size* of ϕ, denoted $|\phi|$, to be the number of symbols occurring in ϕ.

$\square$

Thus, the symbols $-$, $^-$ and $\cap$ do double duty: as operators on subsets of topological spaces and as term-constructors in $\mathcal{TCC}$. This overloading of notation makes the semantics of $\mathcal{TCC}$ more mnemonic, and should cause no confusion. We note in passing that the size of a binary numeral is the number of digits it contains (not the integer it represents). We allow binary numerals to have leading zeros.

Next, we provide a semantics for $\mathcal{TCC}$. Exploiting the overloading of the symbols $-$, $^-$ and $\cap$, we may write:

Definition 2. A *structure* $\mathfrak{A}$ is a pair (A, I), where A is a topological space and $I : \mathbf{V} \to \mathbb{P}(A)$. The *interpretation* $T^{\mathfrak{A}}$ of a term T in a structure $\mathfrak{A} = (A, I)$ is defined inductively with respect to the topological space A as follows:

$$V^{\mathfrak{A}} = I(V) \text{ for all variables } V$$

$$(-T)^{\mathfrak{A}} = -(T^{\mathfrak{A}})$$

$$(T^{-})^{\mathfrak{A}} = (T^{\mathfrak{A}})^{-}$$

$$(T \cap T')^{\mathfrak{A}} = T^{\mathfrak{A}} \cap {T'}^{\mathfrak{A}}.$$

If ϕ is the atomic formula $T = T'$, then $\mathfrak{A}$ *satisfies* ϕ if $T^{\mathfrak{A}} = {T'}^{\mathfrak{A}}$. If ϕ is the atomic formula $c^{\leq k}(T)$, then $\mathfrak{A}$ *satisfies* ϕ if $T^{\mathfrak{A}}$ has at most n components, where n is the (non-negative) integer represented by the binary numeral k. Satisfaction is extended to Boolean combinations of atomic formulas in the obvious way. If $\mathfrak{A}$ satisfies ϕ, we write $\mathfrak{A} \models \phi$. A formula is *satisfiable* if there exists a structure which satisfies it. Two formulas ϕ and ϕ' are *equisatisfiable* if either both are satisfiable or neither is satisfiable.

□

In the sequel, we freely equivocate between binary numerals k and the integers they represent. Under this equivocation, for example, we can say that $\mathfrak{A}$ satisfies $c^{\leq k}(T)$ if and only if $T^{\mathfrak{A}}$ has at most k components. In addition, we allow ourselves to perform arithmetic on binary numerals, for example writing $k+1$ and $k-1$ (if $k > 0$) with the obvious meaning. These expedients avoid cumbersome circumlocutions, and should cause no confusion.

The following abbreviations promote readability in $\mathcal{TCC}$. Let V_0 be some variable. We abbreviate the term $V_0 \cap -V_0$ by $\emptyset$, and any term of the form $-(-T \cap -T')$ by $T \cup T'$. Likewise, we abbreviate any formula of the form $T \cap T' = T'$ by $T' \subseteq T$, any formula of the form $\neg T = T'$ by $T \neq T'$, and any formula of the form $\neg c^{\leq k}(T)$ by $c^{\geq k+1}(T)$. Thus, for any non-zero binary numeral k, $c^{\geq k}(T)$ states that T has at least k components. Finally, we write $c(T)$ in place of the more cumbersome $c^{\leq 1}(T)$ to state that T is connected.

The following simple examples illustrate the language $\mathcal{T}CC$.

Example 1. Let T and T' be any terms. The formula ϕ given by

$$c(T) \wedge c(T') \wedge T \cap T' \neq \emptyset \wedge \neg c(T \cup T')$$

is unsatisfiable.

Proof. A reformulation of Observation 2. □

Example 2. Let T be a term and $W_1, \ldots W_k$ be variables not occurring in T. Then the formula $c^{\geq k}(T)$ is equisatisfiable with the formula ϕ given by:

$$T = \bigcup_{1 \leq i \leq k} W_i \wedge \bigwedge_{1 \leq i \leq k} W_i \neq \emptyset \wedge \bigwedge_{1 \leq i < j \leq n} T \cap W_i^- \cap W_j^- = \emptyset.$$

Proof. If $\mathfrak{A} \models \phi$, then the sets $W_i^{\mathfrak{A}}$ $(1 \leq i \leq k)$ must be unions of disjoint sets of components of $T^{\mathfrak{A}}$. Hence $\mathfrak{A} \models c^{\geq k}(T)$.

Conversely, if $\mathfrak{A} \models c^{\geq k}(T)$, choose any $k-1$ components $B_1, \ldots, B_{k-1}$ of $T^{\mathfrak{A}}$. Define a structure $\mathfrak{A}'$ to be just like A except that $W_i^{\mathfrak{A}'} = B_i$ for all i $(1 \leq i < k)$ and $W_k^{\mathfrak{A}'} = T^{\mathfrak{A}} \setminus (B_1 \cup \ldots \cup B_{k-1})$. Since the variables W_i do not occur in T, we have $T^{\mathfrak{A}'} = T^{\mathfrak{A}}$. It is then easy to check that $\mathfrak{A}' \models \phi$. □

Example 3. Let m and n be integers such that $1 < m < n$, and let $T, T_1, \ldots, T_n$ be any terms. Then the formula ϕ given by

$$c(T) \wedge T = \bigcup_{1 \leq i \leq n} T_i \wedge \bigwedge_{1 \leq i < n} (T_i^- \subseteq T_i \cup T_{i+1}) \wedge$$
$$T_n^- \subseteq T_n \wedge T_1 \neq \emptyset \wedge T_n \neq \emptyset \wedge T_m = \emptyset$$

is unsatisfiable.

Proof. Suppose, for contradiction, $\mathfrak{A} \models \phi$. Let $B = T^{\mathfrak{A}}$, and consider the sets $B_1 = (T_1 \cup \ldots T_{m-1})^{\mathfrak{A}}$ and $B_2 = (T_{m+1} \cup \ldots T_n)^{\mathfrak{A}}$. The conjuncts $T_i^- \subseteq T_i \cup T_{i+1}$ $(1 \leq i < n)$ and $T_n^- \subseteq T_n$ ensure that $B_1^- \cap B_2 = \emptyset$ and $B_2^- \subseteq B_2$, whence $B_1^- \cap B_2^- = \emptyset$. The conjuncts $T_1 \neq \emptyset$ and $T_n \neq \emptyset$ ensure that B_1 and B_2 are nonempty. Finally, the conjuncts $T = \bigcup_{1 \leq i \leq n} T_i$ and $T_m = \emptyset$ ensure that $B = B_1 \cup B_2$. Thus, B is not connected, which is impossible given the conjunct $c(T)$. □

Following standard practice, we write $\models \phi$ if $\mathfrak{A} \models \phi$ for every structure $\mathfrak{A}$—that is to say, if $\neg\phi$ is not satisfiable. Thus we have, for $1 < m < n$ and any terms $T, T_1, \ldots, T_n$:

$$\models \left(c(T) \wedge T = \bigcup_{1 \leq i \leq n} T_i \wedge \right.$$
$$\left. \bigwedge_{1 \leq i < n} (T_i^- \subseteq T_i \cup T_{i+1}) \wedge T_n^- \subseteq T_n \wedge T_1 \neq \emptyset \wedge T_n \neq \emptyset \right) \rightarrow T_m \neq \emptyset$$

The notion of satisfiability in $\mathcal{TCC}$ leads naturally to the following problem.

Definition 3. The problem $\mathcal{TCC}$-SAT is defined as follows :

Instance: A $\mathcal{TCC}$ formula ϕ

Question: Is ϕ satisfiable?

□

We can now state the main result of this paper.

Theorem. *The problem $\mathcal{TCC}$-SAT is NEXPTIME-complete.*

The next two sections are devoted to a proof of this result.

4. Membership in NEXPTIME

4.1. *Conventions and strategy*

We use the term *constraint* to refer to a formula of one of the following types:

Type I: $T = T'$

Type II: $T \neq T'$

Type III: $c^{\leq k}(T)$ $(k \geq 0)$

Type IV: $c^{\geq k}(T)$ $(k \geq 1)$.

If Φ is a set of constraints and $\mathfrak{A}$ a structure, we write $\mathfrak{A} \models \Phi$ to mean that $\mathfrak{A} \models \phi$ for every $\phi \in \Phi$, and we say that Φ is satisfiable if there exists such an $\mathfrak{A}$. If Φ is a finite set of constraints, we take the size of Φ to be the sum of the sizes of its members. The main goal of this section is to prove that the satisfiability of a set of constraints can be decided in non-deterministic exponential time. That $\mathcal{TCC}$-SAT is in NEXPTIME then follows as an easy corollary.

Our strategy is as follows. We prove that any satisfiable set Φ of constraints containing no type-IV constraints is satisfiable in a topological space whose size is bounded by an exponential function of the size of Φ. (We take the size of a topological space to be the cardinality of its carrier set.) We then show that, given any finite set of constraints Φ, we may compute, in polynomial time, an equisatisfiable set of constraints Φ' which is type-IV-free.

4.2. *Small model property*

Let A be a topological space and let $\mathbf{B}$ be a finite set of subsets of A, with the property that $B \in \mathbf{B} \Rightarrow B^- \in \mathbf{B}$. Define a binary relation $\sim$ on A by setting $a \sim b$ if, for all $B \in \mathbf{B}$, $a \in B \Leftrightarrow b \in B$. Let $G = A/\sim$ and define a function $f : A \to G$ by

$$f(a) \quad = \quad \{b \in A \mid b \sim a\}.$$

Observation 5. $|G| \leq 2^{|\mathbf{B}|}$.

Observation 6. *If $a \in A$ and $B \in \mathbf{B}$, then $a \in B$ if and only if $f(a) \in f(B)$.*

Now define a binary relation $\to$ on G by

$$v \to v' \text{ if and only if, for all } B \in \mathbf{B}, v \in f(B) \Rightarrow v' \in f(B^-).$$

Lemma 1. *The directed graph $(G, \to)$ is reflexive and transitive.*

Proof. Since $B \subseteq B^-$, $f(B) \subseteq f(B^-)$, so that $v \to v$ for all $v \in G$. Suppose now that $v \to v' \to v''$. Let $B \in \mathbf{B}$ and $v \in f(B)$. Since $v \to v'$, $v' \in f(B^-)$. Since $B^- \in \mathbf{B}$ and $v' \to v''$, $v'' \in f(B^{--}) = f(B^-)$. Hence $v \to v''$. □

We noted in section 2 that any reflexive, transitive, directed graph $(G, \to)$ can be regarded as a topological space with closure operator $\cdot^{-_G}$ satisfying:

$$H^{-_G} \quad = \quad \{v' \in G \mid v \to v' \text{ for some } v \in H\}.$$

For the sake of readability in the following argument, we shall use $\cdot^-$ to denote the closure operator in the topological space A, and $\cdot^{-_G}$ to denote the closure operator in the topological space G.

We now establish some properties of the function $f : A \to G$.

Lemma 2. *If $v \in G$ and $a \in (f^{-1}(v))^-$, then $v \to f(a)$.*

Proof. Suppose $B \in \mathbf{B}$ and $v \in f(B)$. We must show that $f(a) \in f(B^-)$. By Observation 6 (if-direction), $v \in f(B) \Rightarrow f^{-1}(v) \subseteq B$. Hence $(f^{-1}(v))^- \subseteq B^-$. But then $a \in B^-$ and so $f(a) \in f(B^-)$ as required. □

Lemma 3. *The function $f : A \to G$ is continuous.*

Proof. Let $H \subseteq G$ be closed; we must show that $f^{-1}(H) \subseteq A$ is also closed. Let $H = \{v_0, \ldots v_m\}$, and suppose $a \in (f^{-1}(H))^-$. Since $(f^{-1}(H))^- = (f^{-1}(v_0))^- \cup \ldots \cup (f^{-1}(v_m))^-$, we have $a \in (f^{-1}(v_i))^-$ for some i, so that, by Lemma 2, $v_i \to f(a)$. Since H is closed, $f(a) \in H$, so that $a \in f^{-1}(H)$. Hence $(f^{-1}(H))^- \subseteq f^{-1}(H)$, and so $f^{-1}(H)$ is closed as required. □

Lemma 4. *Let $B \in \mathbf{B}$ such that $-B \in \mathbf{B}$. Then $f(-B) = -f(B)$. Furthermore, Let $B, B' \in \mathbf{B}$ such that $B \cap B' \in \mathbf{B}$. Then $f(B \cap B') = f(B) \cap f(B')$.*

Proof. Instant from Observation 6. □

Lemma 5. $B \in \mathbf{B} \Rightarrow f(B^-) = f(B)^{-_G}$.

Proof. Certainly, $f(B) \subseteq f(B^-)$. Moreover, $B \in \mathbf{B} \Rightarrow f(B^-)$ is closed. For suppose $v \in f(B^-)$ and $v \to v'$: since $B^- \in \mathbf{B}$ and $B^{--} = B^-$, $v' \in f(B^-)$ by the definition of $\to$. It thus suffices to show that $B \in \mathbf{B} \Rightarrow f(B^-) \subseteq f(B)^{-_G}$.

Suppose that $B \in \mathbf{B}$ and $v' \in f(B^-)$. We show that there exists a $v \in f(B)$ such that $v \to v'$. Let $B_1, \ldots, B_m$ be a complete listing of the elements B' of $\mathbf{B}$ for which $v' \notin f(B'^-)$. Let $a' \in A$ such that $f(a') = v'$. Since $B^-, B_1^-, \ldots, B_m^-$ are all elements of $\mathbf{B}$, we have, by Observation 6,

$$a' \in B^- \cap -(B_1^-) \cap \cdots \cap -(B_m^-).$$

The set $-(B_1^-) \cap \cdots \cap -(B_m^-)$ is open in A, so that, by Observation 1, there exists an element a of A such that

$$a \in B \cap -(B_1^-) \cap \cdots \cap -(B_m^-).$$

Let $v = f(a)$. By Observation 6 again,

$$v \in f(B) \cap -f(B_1^-) \cap \cdots \cap -f(B_m^-).$$

Since $v \in f(B)$, we need only show that $v \to v'$. Suppose $B' \in \mathbf{B}$ with $v \in f(B')$. Certainly, then, $v \in f(B'^-)$, whence $B' \neq B_i$ for all i $(1 \leq i \leq m)$. But by the choice of these B_i, $v' \in f(B'^-)$. That is: $B' \in \mathbf{B}$ and $v \in f(B')$ implies $v' \in f(B'^-)$. Hence, $v \to v'$. □

With these preliminaries behind us, let Φ be a type-IV-free set of constraints—i.e. one with no elements of the form $c^{\geq k}(T)$—and let $\mathfrak{A} = \langle A, I \rangle$ be a structure such that $\mathfrak{A} \models \Phi$; we now show how to manufacture a "small" structure satisfying Φ.

Let

$$\begin{aligned} \mathbf{B}' &= \{T^{\mathfrak{A}} \mid T \text{ is a term occurring in } \Phi\} \\ \mathbf{B} &= \mathbf{B}' \cup \{B^- \mid B \in \mathbf{B}'\}. \end{aligned}$$

Thus, $\mathbf{B}$ is a finite set of subsets of A, with the property that $B \in \mathbf{B} \Rightarrow B^- \in \mathbf{B}$. Defining $f : A \to G$ as above, let $f(\mathfrak{A})$ be the structure $\langle f(A), f \circ I \rangle$.

Lemma 6. *For every term T occurring in Φ,*

$$f(T^{\mathfrak{A}}) = T^{f(\mathfrak{A})}.$$

Proof. By induction on the structure of T. If T is a variable, the equation is immediate by the definition of $f(\mathfrak{A})$. The cases $T = -T'$ and $T = T' \cap T''$ are dealt with by Lemma 4. The case $T = T'^{-}$ is dealt with by Lemma 5. □

Lemma 7. *If Φ is type-IV-free, $\mathfrak{A} \models \Phi$, and f is as defined above, then $f(\mathfrak{A}) \models \Phi$.*

Proof. We deal with the three types of constraints in turn. I: $\mathfrak{A} \models T = T' \Rightarrow T^{\mathfrak{A}} = T'^{\mathfrak{A}} \Rightarrow f(T^{\mathfrak{A}}) = f(T'^{\mathfrak{A}}) \Rightarrow T^{f(\mathfrak{A})} = T'^{f(\mathfrak{A})}$ (by Lemma 6) $\Rightarrow f(\mathfrak{A}) \models T = T'$. II: $\mathfrak{A} \models T \neq T' \Rightarrow T^{\mathfrak{A}} \neq T'^{\mathfrak{A}} \Rightarrow f(T^{\mathfrak{A}}) \neq f(T'^{\mathfrak{A}})$ (by Observation 6, since $T^{\mathfrak{A}}$ and $T'^{\mathfrak{A}}$ are both in $\mathbf{B}$) $\Rightarrow T^{f(\mathfrak{A})} \neq T'^{f(\mathfrak{A})}$ (by Lemma 6) $\Rightarrow f(\mathfrak{A}) \models T \neq T'$. III: $\mathfrak{A} \models c^{\leq k}(T) \Rightarrow T^{\mathfrak{A}}$ has at most k components $\Rightarrow f(T^{\mathfrak{A}})$ has at most k components (by Lemma 3) $\Rightarrow T^{f(\mathfrak{A})}$ has at most k components (by Lemma 6) $\Rightarrow f(\mathfrak{A}) \models c^{\leq k}(T)$. □

Corollary 1. *If Φ is a satisfiable set of constraints of types I, II and III, then Φ is satisfied in a structure of size bounded by $2^{2|\Phi|}$.*

Proof. By Lemma 7 and Observation 5, noting that the set $\mathbf{B}$ has at most $2|\Phi|$ elements. □

Of course, Lemma 7 is not correct if type-IV constraints are permitted, since applying a continuous function to a set may decrease the number of its components. It is to constraints of this type that we now turn, therefore.

4.3. *Eliminating type-IV constraints*

The task before us is as follows. Given a set of constraints Φ, compute, in polynomial time, an equisatisfiable set of constraints which is type-IV-free.

Let us first pause to see where the difficulty lies. Recasting Example 2 slightly, we can equisatisfiably replace any constraint $c^{\geq k}(T) \in \Phi$ by the set of constraints

$$\{T = \bigcup_{1 \leq i \leq k} W_i\} \cup \{W_i \neq \emptyset \mid 1 \leq i \leq k\} \cup$$
$$\{T \cap W_i^{-} \cap W_j^{-} = \emptyset \mid 1 \leq i < j \leq n\}.$$

where W_i ($1 \leq i \leq k$) are variables not occurring in Φ. Carrying out this process for all type-IV constraints in Φ (choosing new variables W_i each time) would thus yield an equisatisfiable set of constraints which was type-IV-free.

However, the above replacement process is not a solution to our problem, because it violates the requirement that the resulting set of type-IV-free constraints should be computed in polynomial time. The difficulty is that the number of new variables introduced for a constraint $c^{\geq k}$ is exponential in the size of (i.e. number of digits in) the binary numeral k. Evidently, more work is required. Nevertheless, the idea of Example 2 does give us the following useful result. Call the numeral (or, equivocally, the number) k in a type-IV constraint $c^{\geq k}(T) \in \Phi$, the *exponent* of the constraint.

Lemma 8. *Let Φ be a set of constraints. Then we can compute, in polynomial time, an equisatisfiable set of constraints Φ' in which all type-IV constraints have exponents which are powers of 2.*

Proof. Let $\phi = c^{\geq k}(T) \in \Phi$, and let V be a variable not occurring in Φ. Let k', k'' be binary numerals such that $k = k' + k''$, and let Φ' be the result of replacing ϕ in Φ by the set of constraints

$$\{T \cap V \neq \emptyset, T \cap -V \neq \emptyset, c^{\geq k'}(T \cap V), c^{\geq k''}(T \cap -V), T \cap V^{-} \cap (-V)^{-} = \emptyset\}.$$

By a similar argument to that used in Example 2, Φ and Φ' are equisatisfiable. But of course k can be written as a sum of powers of 2 involving no more terms than there are digits in the numeral k. Hence, by carrying out the above replacement process repeatedly, we obtain, in polynomial time, an equisatisfiable set of constraints satisfying the requirements of the lemma. □

Thus the main hurdle we have to overcome is the following. Given a natural number d, devise a set of constraints of types I, II and III, of size bounded by some fixed polynomial in d, which force some set to have at least 2^d components. We begin with some routine lemmas on structures.

Definition 4. Let $\mathfrak{A} = \langle A, I \rangle$ be a structure and V a variable. The *relativization* of $\mathfrak{A}$ to V is the structure $\mathfrak{A}_V = \langle A_V, I_V \rangle$, where $A_V = I(V)$ (with the subspace topology) and $I_V(V') = I(V') \cap A_V$ for all variables V'.

Let T be a term and V a variable. The *relativization* T_V of T to V is defined inductively as follows:

$$\begin{aligned}
(V_i)_V &= V_i \cap V \\
(T \cap T')_V &= T_V \cap T'_V \\
(-T)_V &= V \cap -T_V \\
(T^{-})_V &= V \cap (T_V)^{-}
\end{aligned}$$

If Φ is a set of constraints and V a variable, the *relativization* Φ_V of Φ to V is the result of replacing every term in Φ by its relativization to V.

□

Thus, the relativization of a structure $\mathfrak{A}$ to V is simply the structure obtained from $\mathfrak{A}$ by chopping away everything outside the subspace $V^{\mathfrak{A}}$. The following lemmas show that the relativization of a set of constraints Φ to V is defined in a "matching" fashion.

Lemma 9. *Let $\mathfrak{A}$ be a structure, T a term, and V a variable. Then $(T_V)^{\mathfrak{A}} = T^{\mathfrak{A}_V}$.*

Proof. Routine induction on the structure of T. The only (slightly) nontrivial case is given by $T = T'^{-}$. We have $(T_V)^{\mathfrak{A}} = ((T'^{-})_V)^{\mathfrak{A}} = (V \cap (T'_V)^{-})^{\mathfrak{A}}$. Setting $B = V^{\mathfrak{A}}$, we obtain $(T_V)^{\mathfrak{A}} = B \cap ((T'_V)^{-})^{\mathfrak{A}} = B \cap (T'_V{}^{\mathfrak{A}})^{-} = B \cap (T'^{\mathfrak{A}_V})^{-}$ (by inductive hypothesis). Now setting $C = T'^{\mathfrak{A}_V}$ so that $C \subseteq B \subseteq A$, Observation 3 gives us $(T_V)^{\mathfrak{A}} = (T'^{\mathfrak{A}_V})^{-_B} = (T'^{-})^{\mathfrak{A}_V}$. □

Lemma 10. *Let $\mathfrak{A}$ be a structure, Φ a set of constraints, and V a variable. Then $\mathfrak{A} \models \Phi_V$ if and only if $\mathfrak{A}_V \models \Phi$.*

Proof. The result for type-I and type-II constraints is immediate from Lemma 9. The result for type-III and type-IV constraints follows from Lemma 9 and Observation 4. □

Lemma 11. *Let Φ be a set of constraints, and V a variable not occurring in Φ. Then Φ and Φ_V are equisatisfiable.*

Proof. If $\mathfrak{A} \models \Phi_V$, then $\mathfrak{A}_V \models \Phi$ by the previous lemma. Conversely, suppose $\mathfrak{A} \models \Phi$. Since V does not occur in Φ, we may suppose that $V^{\mathfrak{A}} = A$. But then it is obvious that $T_V^{\mathfrak{A}} = T^{\mathfrak{A}}$ for every term T, whence $\mathfrak{A} \models \Phi_V$. □

The main idea in the elimination of type-IV constraints involves the simulation of binary arithmetic in structures. In the sequel, if x is a d-digit binary numeral (allowing leading zeros) and $1 \leq i \leq d$, we take $x[i]$ to denote the ith bit of x, counting from right to left (so that $x[1]$ denotes the least significant bit of x). Note that, if x and x' are both d-digit binary numerals, the expressions $x < 2^d - 1$ and (in that case) $x' = x + 1$ are meaningful.

Lemma 12. *Let x and x' be d-digit binary numerals, with $x < 2^d - 1$. Let i be the least value of j such that $x[j] = 0$. Then $x' \notin \{x, x+1\}$ if and only if at least one of the following conditions holds:*

a) for some j ($i < j \leq d$), $x[j] \neq x'[j]$;

b) $x'[i] = 0$, and for some j ($1 \leq j < i$), $x'[j] = 0$;

c) $x'[i] = 1$, and for some j ($1 \leq j < i$), $x'[j] = 1$.

Proof. It is straightforward to check that, if any of a)–c) hold, then $x' \neq x$ and $x' \neq x + 1$. For the converse, suppose that conditions a)–c) all fail; we must show that $x' = x$ or $x' = x + 1$. By the failure of a), x and x' are identical after (i.e. to the left of) the ith bit. If $i = 1$, then $x[1] = 0$, and so we instantly have $x' = x$ or $x' = x + 1$. Thus, we may assume that $i > 1$, so that x has the appearance:

$x[d]$	$\cdots$	$x[i+1]$	0	$1 \cdots 1$

.

Clearly, either $x'[i] = 0$ or $x'[i] = 1$, so that, by the failure of b) and c), and by the fact that $i > 1$, exactly one of the following conditions holds:

d) for all j $(1 \leq j < i)$, $x'[j] = 1$;

e) for all j $(1 \leq j < i)$, $x'[j] = 0$.

If d) holds and e) fails, then by the failure of c), we have $x'[i] = 0$, whence $x' = x$. On the other hand, if e) holds and d) fails, then by the failure of b), we have $x'[i] = 1$, whence $x' = x + 1$. □

In the sequel, we use the (possibly subscripted) letters U, V, W, X and Y to range over variables—i.e. over the elements of $\mathbf{V}$. Let $X_1, \ldots, X_d$ be variables, then. Denote by $\pm X_i$ either of the terms X_i or $-X_i$. Then the terms of the form

$$\pm X_d \cap \cdots \cap \pm X_1$$

correspond in the obvious way to d-digit binary numerals, where X_i represents $x[i] = 1$ and $-X_i$ represents $x[i] = 0$. In addition, we use the letters x and x' equivocally for d-digit binary numerals, integers in the range $[0, 2^d - 1]$ and terms of the above form. Finally, for all i $(1 \leq i \leq d)$, we use X_i^* as an abbreviation for the term

$$-X_i \cap X_{i-1} \cap X_{i-2} \cap \cdots \cap X_2 \cap X_1.$$

(If $i = 1$, then X_i^* is just $-X_1$.)

Recall that we write $\models \phi$ if $\mathfrak{A} \models \phi$ for every structure $\mathfrak{A}$.

Example 4. If x is odd, then $\models x \subseteq X_1$; if x is even, then $\models x \subseteq -X_1$.

Example 5. Taking $d = 5$, the trivial fact

$$\models X_5 \cap X_4 \cap -X_3 \cap X_2 \cap X_1 \subseteq -X_3 \cap X_2 \cap X_1$$

can be written more suggestively as $\models 27 \subseteq X_3^*$, indicating that the first zero bit in the binary representation of 27, counting from the right, is the third bit.

For the next lemma, remember that $2^d - 1$ abbreviates the term $X_d \cap \cdots \cap X_1$.

Lemma 13. *Let $X_1, \ldots, X_d$ be variables ($d \geq 1$), and T' a term. Let Φ consist of the following constraints:*

(1) *For all i, j $(1 \leq i < j \leq d)$,*

$$\begin{aligned} T' \cap (X_i^* \cap X_j)^- \cap -X_j &= \emptyset \\ T' \cap (X_i^* \cap -X_j)^- \cap X_j &= \emptyset \end{aligned}$$

(2) *For all i, j ($1 \le j < i \le d$),*

$$\begin{aligned} T' \cap (X_i^*)^- \cap X_j \cap X_i &= \emptyset \\ T' \cap (X_i^*)^- \cap -X_j \cap -X_i &= \emptyset \end{aligned}$$

(3) $T' \cap (2^d - 1)^- \subseteq (2^d - 1)$.

Suppose $\mathfrak{A} \models \Phi$. Then, for all x, x' ($0 \le x, x' \le 2^d - 1$), $\mathfrak{A} \models T' \cap x^- \cap x' \neq \emptyset$ only if $x' = x$ or $x' = x + 1$.

Proof. Let $\mathfrak{A} \models T' \cap x^- \cap x' \neq \emptyset$ with $x' \neq x$ and $x' \neq x + 1$. By **(3)**, $x \le 2^d - 2$, so let i be the smallest j such that $x[j] = 0$. By Lemma 12, one of the conditions a)–c) listed there holds. Suppose a) holds, with, say, $x[j] = 1$ and $x'[j] = 0$ for some $j > i$. Then we have $\models x \subseteq (X_i^* \cap X_j)$ and $\models x' \subseteq -X_j$. But given that $\mathfrak{A} \models T' \cap x^- \cap x' \neq \emptyset$, we have $\mathfrak{A} \models T' \cap (X_i^* \cap X_j)^- \cap -X_j \neq \emptyset$, contradicting the assumption that $\mathfrak{A}$ satisfies constraints **(1)**. Similarly, supposing that b) or c) hold contradicts the assumption that $\mathfrak{A}$ satisfies constraints **(2)**. □

Lemma 14. *Let $X_1, \ldots, X_d$ be variables and T' a term. Let Φ consist of* **(1)**–**(3)** *of Lemma 13 together with the constraint:*

(4) $c(T')$

stating that T' is connected. Suppose that $\mathfrak{A} \models \Phi$, $\mathfrak{A} \models T' \cap 0 \neq \emptyset$ and $\mathfrak{A} \models T' \cap (2^d - 1) \neq \emptyset$. Then $\mathfrak{A} \models T' \cap x^- \cap (x + 1) \neq \emptyset$ for all x ($0 \le x \le 2^d - 2$).

Proof. By Lemma 13, for all x ($0 \le x \le 2^d - 2$),

$$\mathfrak{A} \models T' \cap x^- \subseteq x \cup (x + 1). \tag{1}$$

Now we apply reasoning similar to that of Example 3. Suppose, for contradiction, that $\mathfrak{A} \models T' \cap {x_0}^- \cap (x_0 + 1) = \emptyset$ for some x_0 ($0 \le x_0 \le 2^d - 2$). Consider the sets $B_1 = ((T' \cap 0) \cup \ldots \cup (T' \cap x_0))^{\mathfrak{A}}$ and $B_2 = ((T' \cap (x_0 + 1)) \cup \ldots \cup (T' \cap (2^d - 1)))^{\mathfrak{A}}$. From (1) and constraint **(3)**, B_1 and B_2 are then closed in the subspace ${T'}^{\mathfrak{A}}$ and are visibly disjoint. Moreover, the fact that $\mathfrak{A} \models T' \cap 0 \neq \emptyset$ and $\mathfrak{A} \models T' \cap (2^d - 1) \neq \emptyset$ implies that B_1 and B_2 are nonempty. But this contradicts constraint **(4)**. □

Lemma 15. *Let $X_1, \ldots, X_d, U, V$ be variables. Setting T' to the term $(U \cup -V)$, let Φ consist of* **(1)**–**(4)** *above together with the constraints*

(5) $V^- \subseteq V$

(6) *for all i (*$1 \leq i \leq d$*),*

$$\begin{aligned}(U \cap X_i)^- \cap (U \cap -X_i) &= \emptyset\\ (U \cap -X_i)^- \cap (U \cap X_i) &= \emptyset\end{aligned}$$

(7) *for all i (*$1 \leq i \leq d$*),*

$$\begin{aligned}(-V \cap X_i)^- \cap (-V \cap -X_i) &= \emptyset\\ (-V \cap -X_i)^- \cap (-V \cap X_i) &= \emptyset\end{aligned}$$

(8) $U \cap 0 \neq \emptyset$ *and* $U \cap (2^d - 1) \neq \emptyset$*.*

Suppose $\mathfrak{A} \models \Phi$*. Then* $\mathfrak{A} \models c^{\geq 2^d}(U)$*.*

Proof. Since $\mathfrak{A}$ satisfies the constraints **(6)**,

$$\mathfrak{A} \models U \cap x^- \cap x' = \emptyset$$

for all distinct x, x' ($0 \leq x, x' \leq 2^d - 1$). It suffices to show that, for all such x, $\mathfrak{A} \models U \cap x \neq \emptyset$, for then each of the sets $(U \cap x)^{\mathfrak{A}}$ must contain points belonging to pairwise distinct components of $U^{\mathfrak{A}}$.

Let $1 \leq x \leq 2^d - 2$. Trivially, $\models x^- = (x \cap V)^- \cup (x \cap -V)^-$, whence

$$\models -V \cap x^- \cap (x+1) \subseteq ((x \cap V)^- \cap -V) \cup ((x \cap -V)^- \cap (x+1) \cap -V).$$

By constraint **(5)**, $\mathfrak{A} \models (x \cap V)^- \cap -V = \emptyset$, and by constraints **(7)**, $\mathfrak{A} \models (x \cap -V)^- \cap (x+1) \cap -V = \emptyset$. Hence

$$\mathfrak{A} \models -V \cap x^- \cap (x+1) = \emptyset. \tag{2}$$

By constraints **(1)**–**(4)** (with T' set to $U \cup -V$) and constraints **(8)**, the conditions of lemma 14, are satisfied, so that if $0 \leq x \leq 2^d - 2$,

$$\mathfrak{A} \models (U \cup -V) \cap x^- \cap (x+1) \neq \emptyset. \tag{3}$$

Putting together (2) and (3) gives us

$$\mathfrak{A} \models U \cap x^- \cap (x+1) \neq \emptyset.$$

Thus, if $0 \leq x \leq 2^d - 2$, then $\mathfrak{A} \models U \cap (x+1) \neq \emptyset$; the fact that $\mathfrak{A} \models U \cap 0 \neq \emptyset$ is guaranteed by constraints **(8)**. $\square$

Lemma 16. *Let T be a term and Φ a set of constraints containing $\phi = c^{\geq 2^d}(T)$ (with $d > 0$). Let $U, V, X_1, \ldots, X_d$ be variables not occurring in Φ. Let Ψ be the set of constraints* **(1)–(8)** *above (with T' set to $U \cup -V$), together with the constraints:*

(9) $U \subseteq T_V$

(10) $U^- \cap (-U)^- \cap T_V = \emptyset$.

Let $\Phi' = (\Phi_V \setminus \{\phi_V\}) \cup \Psi$. Then Φ and Φ' are equisatisfiable.

Proof. Suppose $\mathfrak{A} \models \Phi'$. By Lemma 15, $U^{\mathfrak{A}}$ has at least 2^d components, whence, by constraints **(9)** and **(10)**, $T_V^{\mathfrak{A}}$ also has at least 2^d components. Hence $\mathfrak{A} \models \phi_V$, so that $\mathfrak{A} \models \Phi_V$. By Lemma 10, $\mathfrak{A}_V \models \Phi$.

Conversely, suppose $\mathfrak{A} \models \Phi$. We construct $\mathfrak{B}$ satisfying Φ' as follows. Refer to figure 1. Let $C_0, \ldots, C_{2^d-1}$ be distinct components of $T^{\mathfrak{A}}$, and choose points $q_0, \ldots q_{2^d-1}$ with $q_i \in C_i$ for all i ($0 \leq i \leq 2^d - 1$). Let $P = \{p_0, \ldots, p_{2^d-1}\}$ be a set of 2^d distinct points not in A. Let $B = A \cup P$. For all i ($0 \leq i \leq 2^d - 2$), let $Q_i = \{q_i, q_{i+1}\}$ and let $Q_{2^d-1} = \{q_{2^d-1}\}$. To turn B into a topological space, let the closed sets of B be those of the form

$$A'^- \cup P' \cup \bigcup \{Q_i^- \mid p_i \in P'\}$$

where $A' \subseteq A$ and $P' \subseteq P$. (Note that the closure operators in this expression refer to the topological space A). It is routine to check that B is a topological space; indeed, A has the subspace topology induced by B. To define the structure $\mathfrak{B} = \langle B, J \rangle$, let J be the same as I, with the following exceptions. Set $J(V) = A$ and $J(U) = C_0 \cup \cdots \cup C_{2^d-1}$; furthermore, fix the $J(X_i)$ ($1 \leq i \leq d$) in such a way that, for all x ($0 \leq x \leq 2^d - 1$), $(U \cap x)^{\mathfrak{B}} = C_x$, and $(-V \cap x)^{\mathfrak{B}} = \{p_x\}$, as shown in figure 1. It is obvious that such an assignment is possible.

To see that $\mathfrak{B} \models \Phi'$, define $\mathfrak{A}' = \langle A, I' \rangle$ to be exactly like the structure $\mathfrak{A}$ except that we fix $I'(V) = J(V) = A$, $I'(U) = J(U) = A \cap J(U)$, and, for all i ($1 \leq i \leq d$), $I'(X_i) = A \cap J(X_i)$. Since $U, V, X_1, \ldots, X_d$ do not occur in Φ, $\mathfrak{A}' \models \Phi$. Moreover, by the construction of $\mathfrak{B}$ and $\mathfrak{A}'$, we have $\mathfrak{A}' = \mathfrak{B}_V$. By Lemma 10, $\mathfrak{B} \models \Phi_V$. It is routine to verify, by inspection of figure 1, that $\mathfrak{B} \models \Psi$. □

We have now completed the task of this section.

Theorem 1. *The problem $\mathcal{TCC}$-SAT is in NEXPTIME.*

Proof. Let Φ be a given set of constraints. By Lemmas 8 and 16, we can compute in polynomial time an equisatisfiable set of constraints Φ' such that Φ' is type-IV-free. By corollary 1, if Φ' is satisfiable, then $\mathfrak{A} \models \Phi'$ for some $\mathfrak{A} = \langle A, I \rangle$, with $|A|$ bounded by an exponential function of $|\Phi'|$. In fact, we may assume that A is presented in the form of a reflexive, transitive graph. But we can verify that $\mathfrak{A} \models \Phi'$ in time polynomially bounded by $|A| + |\Phi'|$. □

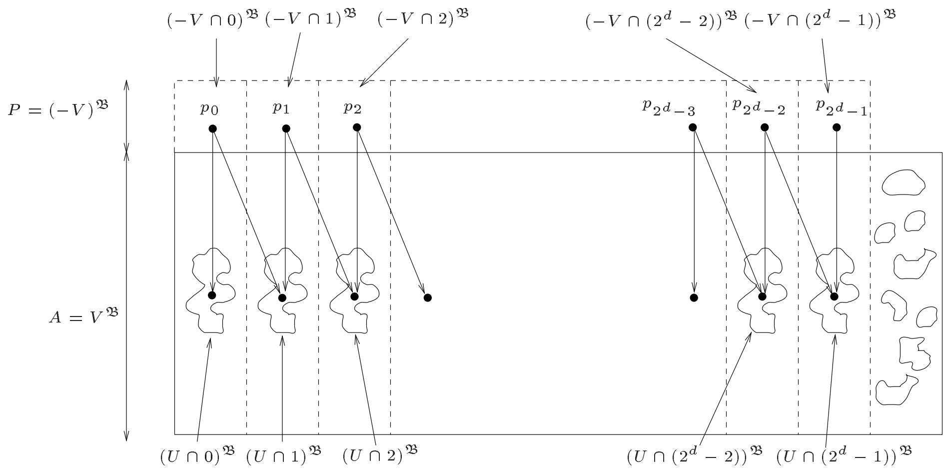

Figure 1. *Construction of the structure $\mathfrak{B}$: the irregularly shaped regions represent the components of $T^{\mathfrak{A}}$; arrows point from each p_i to all points in the corresponding Q_i.*

We end this section with a brief discussion of the techniques employed. Previous investigations of the complexity of topological constraint languages have exploited the relationship between topological spaces and Kripke frames for the modal logic S4. (See section 6 for references.) The filtration technique employed in section 4.2 to prove the small model property for type-IV-free sets of constraints comes as no surprise, therefore: we have avoided the translation of $\mathcal{TCC}$-SAT into a satisfiability problem in modal logic by re-casting the standard filtration argument for S4 in topological terms. The advantage of our approach is of course the ease with which we can deal with connectedness constraints. Lemma 3 assures us that the filtration used to generate small models is a continuous function, and hence preserves type-III constraints. Note that this very simple fact would be completely obscured by translation into propositional modal logics such as S4, in which the property of connectedness is not expressible!

5. NEXPTIME-Hardness

The following definitions are well-known.

Definition 5. A *tiling system* $\mathcal{D}$ is a finite set C, whose elements are called *tiles*, together with a tile $c_1 \in C$ and two relations $H, V \subseteq C \times C$. An *n-tiling* using $\mathcal{D}$ is a function f mapping pairs of integers in the range $[0, n-1]$ to C such that $f(0,0) = c_1$ and for all i, j ($0 \leq i < n$, $0 \leq j \leq n$), $\langle f(i,j), f(i+1,j)\rangle \in H$ and $\langle f(j,i), f(j,i+1)\rangle \in V$.

□

Think of H as expressing *horizontal constraints* (which tiles can go immediately to the right of which other tiles) and V as expressing *vertical constraints* (which tiles can go immediately above which other tiles). Then an n-tiling is an arrangement of tiles on a grid of size n, with c_1 in the bottom left-hand corner, respecting these constraints.

Definition 6. The problem TILING is defined as follows (again, using the usual equivocation between numerals and integers):

Instance: A tiling system $\mathcal{D}$ and a nonzero binary numeral k.

Question: Does $\mathcal{D}$ have a k-tiling?

□

We may take the size of an instance of TILING to be the number of digits in k plus the number of tiles in $\mathcal{D}$. The following result is simple to establish by encoding runs of Turing machines using tilings (see, e.g. Papadimitriou [PAP 94], problem 20.2.10a).

Theorem 2. *The problem TILING is NEXPTIME-hard.*

Indeed, inspection of the encoding shows that Theorem 2 continues to hold in the case where the integer represented by k is restricted to be a power of 2.

To show NEXPTIME hardness of $\mathcal{TCC}$-SAT, it suffices to encode a given tiling problem as a $\mathcal{TCC}$-SAT problem in time bounded by a polynomial function of the original tiling problem. In fact, we show below how a given tiling problem can be polynomially mapped into the problem of determining the satisfiability of a set Φ of constraints of types I, II and III.

5.1. *Establishing a grid*

Let $X_1, \ldots, X_d$, $Y_1, \ldots, Y_d$ be variables. As before, any term

$$\pm X_d \cap \ldots \cap \pm X_1 \cap \pm Y_d \cap \ldots \cap \pm Y_1$$

encodes a pair of integers in the range $[0, 2^d - 1]$ in the obvious way. We use (x, y), (x', y') *etc.* equivocally for such expressions and the corresponding pairs of integers. It helps to think of the (x, y) as squares arranged in a grid-like pattern, with 2^d rows and 2^d columns. Consider the constraints:

(11) for all j, k ($1 \leq j, k \leq d$),

$$\begin{aligned}
(X_j \cap Y_k)^- \cap (-X_j \cap -Y_k) &= \emptyset \\
(-X_j \cap Y_k)^- \cap (X_j \cap -Y_k) &= \emptyset \\
(X_j \cap -Y_k)^- \cap (-X_j \cap Y_k) &= \emptyset \\
(-X_j \cap -Y_k)^- \cap (X_j \cap Y_k) &= \emptyset.
\end{aligned}$$

It is easy to see that, for any structure $\mathfrak{A}$ satisfying (**11**), if $\mathfrak{A} \models (x, y)^- \cap (x', y') \neq \emptyset$, then $x' = x$ or $y' = y$. (Proof: If $x \neq x'$ and $y \neq y'$, let x and x' differ in the jth digit and let y and y' differ in the kth digit. Recalling that $x[j] = 1$ is the same as $\models x \subseteq X_j$, $x[j] = 0$ is the same as $\models x \subseteq -X_j$, and similarly for x', y and y', we cannot have $\mathfrak{A} \models (x, y)^- \cap (x', y') \neq \emptyset$ without violating one of the above constraints.)

For all i ($1 \leq i \leq d$) use the abbreviation X_i^* for the term $-X_i \cap X_{i-1} \cap X_{i-2} \cap \cdots \cap X_2 \cap X_1$ as before, and the abbreviation Y_i^* analogously. By analogy with the constraints (**1**)–(**3**) of Lemma 13, consider the constraints:

(1)$_X$ for all i, j ($1 \leq i < j \leq d$),

$$\begin{aligned}
(X_i^* \cap X_j)^- \cap -X_j &= \emptyset \\
(X_i^* \cap -X_j)^- \cap X_j &= \emptyset
\end{aligned}$$

(2)$_X$ for all i, j ($1 \leq j < i \leq d$),

$$\begin{aligned}
(X_i^*)^- \cap X_j \cap X_i &= \emptyset \\
(X_i^*)^- \cap -X_j \cap -X_i &= \emptyset
\end{aligned}$$

(3)$_X$ $(X_d \cap \cdots \cap X_1)^- \subseteq X_d \cap \cdots \cap X_1$.

(1)$_Y$ for all i, j ($1 \leq i < j \leq d$),

$$\begin{aligned}
(Y_i^* \cap Y_j)^- \cap -Y_j &= \emptyset \\
(Y_i^* \cap -Y_j)^- \cap Y_j &= \emptyset
\end{aligned}$$

(2)$_Y$ for all i, j ($1 \leq j < i \leq d$),

$$\begin{aligned}
(Y_i^*)^- \cap Y_j \cap Y_i &= \emptyset \\
(Y_i^*)^- \cap -Y_j \cap -Y_i &= \emptyset
\end{aligned}$$

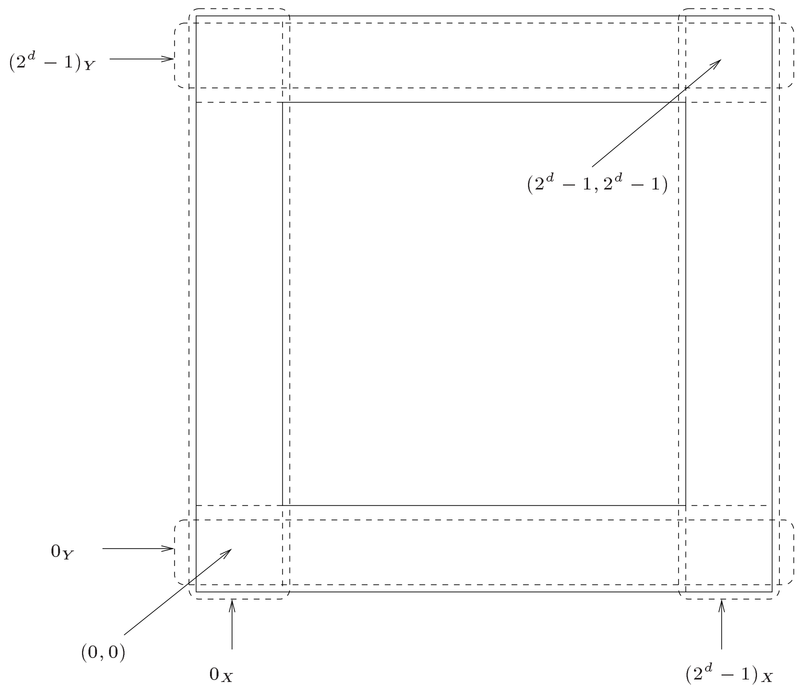

Figure 2. *The grid and its periphery*

(3)$_Y$ $\quad (Y_d \cap \cdots \cap Y_1)^- \subseteq Y_d \cap \cdots \cap Y_1$.

By reasoning identical to that of lemma 13 , we have that, for any structure $\mathfrak{A}$ satisfying **(1)**$_X$–**(3)**$_X$, **(1)**$_Y$–**(3)**$_Y$ and **(11)**, if $\mathfrak{A} \models (x,y)^- \cap (x',y') \neq \emptyset$, then $(x',y') = (x,y)$ or $(x',y') = (x+1,y)$ or $(x',y') = (x,y+1)$.

We now write constraints forcing the (x,y) to connect up in a grid-like way. The idea is similar to that of lemma 14. We need one extra piece of notation. If z is in the range $[0, 2^d-1]$, then write z_X for the term involving the $X_1, \ldots X_d$ representing z, and write z_Y for the term involving the $Y_1, \ldots Y_d$ representing z. Remembering that the (x,y) are to be regarded as squares on a grid, we can think of the z_X as the grid's columns and the z_Y as its rows. Suppose we add the constraints

(12) $\quad (0,0) \neq \emptyset$, $(2^d-1, 2^d-1) \neq \emptyset$, $c(0_X \cup (2^d-1)_Y)$, $c(0_Y \cup (2^d-1)_X)$.

By reasoning identical to that of Lemma 14, we see that, for any structure satisfying the constraints **(1)**$_X$–**(3)**$_X$, **(1)**$_Y$–**(3)**$_Y$, **(11)** and **(12)**, all the squares on the periphery of the grid are forced to be nonempty (figure 2).

Now let us turn our attention to the non-peripheral squares on the grid. Geometrically, the term X_1 picks out the collection of squares in the *odd* columns, and the

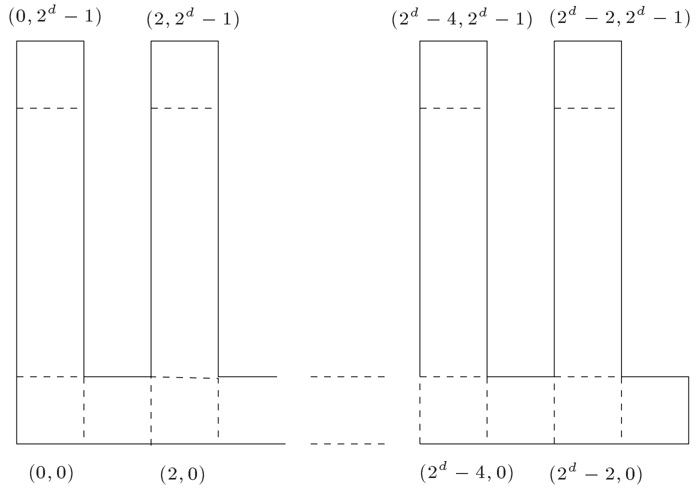

Figure 3. *The comb-like region* $0_Y \cup -X_1$

term $-X_1$, the collection of squares in the *even* columns. Similarly Y_1 picks out the odd rows, and $-Y_1$, the even rows. Thus, the term $0_Y \cup -X_1$ picks out the comb-like region shown in figure 3, consisting of a horizontal bar and 2^{d-1} vertical 'teeth'. Given the above constraints, it is clear that adding the single constraint $c(0_Y \cup -X_1)$ is sufficient to force every square in this comb-like region to be non-empty. For the end-squares of each tooth, being peripheral, are non-empty, and these can only be connected to each other in the relevant region via the other squares in the teeth.

Consider, then the constraints

(13) $c(0_Y \cup -X_1), c(0_Y \cup X_1), c(0_X \cup -Y_1), c(0_X \cup Y_1).$

It is now obvious that if $\mathfrak{A}$ satisfies $(\mathbf{1})_X$–$(\mathbf{3})_X$, $(\mathbf{1})_Y$–$(\mathbf{3})_Y$, (**11**), (**12**) and (**13**), then $\mathfrak{A} \models (x, y) \neq \emptyset$ for all x and y in the range $[0, 2^d - 1]$; in fact, $\mathfrak{A} \models (x, y)^- \cap (x', y') \neq \emptyset$ if and only if (x', y') is one of the squares (x, y), $(x + 1, y)$ or $(x, y + 1)$.

We need one more encoding trick before we are ready to translate a tiling problem into a set of constraints. If T and T' are terms, we take $T \Delta T'$ to abbreviate $(T \cap -T') \cup (-T \cap T')$. Thus, Δ expresses the symmetric difference operator. Then the term $X_1 \Delta Y_1$ picks out the 'black' squares (under a normal chequered pattern), and $-(X_1 \Delta Y_1)$, the 'white' squares. Abbreviate these terms by B and W, respectively. Notice that, if (x, y) and (x', y') are distinct squares falling within the region B, then, for any $\mathfrak{A}$ satisfying the constraints $(\mathbf{1})_X$–$(\mathbf{3})_X$, $(\mathbf{1})_Y$–$(\mathbf{3})_Y$ and (**11**), $\mathfrak{A} \models (x, y)^- \cap (x', y') = \emptyset$. Hence, for any $\mathfrak{A}$ satisfying the constraints $(\mathbf{1})_X$–$(\mathbf{3})_X$, $(\mathbf{1})_Y$–$(\mathbf{3})_Y$ and (**11**)–(**13**), we have $\mathfrak{A} \models c^{\geq 2^{d-1}}(B)$ and $\mathfrak{A} \models c^{\geq 2^{d-1}}(W)$. It follows that if $\mathfrak{A}$ also satisfies the constraints

(14) $c^{\leq 2^{d-1}}(B)$ and $c^{\leq 2^{d-1}}(W)$,

then $\mathfrak{A}$ must interpret each (x, y) as a connected set. Let us denote the set of constraints $(\mathbf{1})_X$–$(\mathbf{3})_X$, $(\mathbf{1})_Y$–$(\mathbf{3})_Y$ and (**11**)–(**14**) by Γ_d (Γ for 'grid'). By inspection, $|\Gamma_d|$ is bounded by some fixed polynomial in d.

5.2. *Encoding a tiling system*

Let $\mathcal{D}$ be a tiling system consisting of the tile set $c_1, \ldots, c_m$ ($m > 1$) and binary relations H and D, and let $k = 2^d$ for some $d \geq 1$. Form the constraints Γ_d, and select distinct variables $C_1, \ldots, C_m$ not occurring in Γ_d. In the sequel, we use the letters C and C' to range over the C_i. Consider the constraints:

(15) for all distinct C, C',

$$C \cap C' = \emptyset$$

(16) $C_m = -(C_1 \cup \cdots \cup C_{m-1})$,

guaranteeing that the sets assigned to these variables are pairwise disjoint and jointly exhaustive, as well as the constraints:

(17) for all distinct C, C',

$$\begin{aligned} (B \cap C)^- \cap (B \cap C') &= \emptyset \\ (W \cap C)^- \cap (W \cap C') &= \emptyset. \end{aligned}$$

Since the constraints Γ_d force every square (x, y) to be connected, any structure $\mathfrak{A}$ satisfying Γ_d and **(15)–(17)** must satisfy the condition that, for any square (x, y), there exists a unique C such that $\mathfrak{A} \models (x, y) \subseteq C$. We may think of C as the tile used to cover the square (x, y). Notice incidentally that constraints **(17)** do not create difficulties for (horizontally or vertically) neighbouring squares covered with different tiles, because no two such neighbouring squares are contained within B, and no two such neighbouring squares are contained within W.

To encode the relations H and V, we add the following following constraints:

(18) for all i, j ($1 \leq i, j \leq m$) such that $\langle c_i, c_j \rangle \notin H$,

$$\begin{aligned} (X_1 \cap C_i)^- \cap -X_1 \cap C_j &= \emptyset \\ (-X_1 \cap C_i)^- \cap X_1 \cap C_j &= \emptyset \end{aligned}$$

(19) for all i, j ($1 \leq i, j \leq m$) such that $\langle c_i, c_j \rangle \notin V$,

$$\begin{aligned} (Y_1 \cap C_i)^- \cap -Y_1 \cap C_j &= \emptyset \\ (-Y_1 \cap C_i)^- \cap Y_1 \cap C_j &= \emptyset. \end{aligned}$$

To see how these constraints have the desired effect, recall that, given Γ_d, if $x \leq 2^d-2$, then $(x,y)^- \cap (x+1,y) \neq \emptyset$, and if $y \leq 2^d-2$, then $(x,y)^- \cap (x,y+1) \neq \emptyset$. Noting that the term X_1 picks out the odd-numbered columns of the grid, and the term $-X_1$, its even-numbered columns, constraints **(18)** impose restrictions on the interpretations of C_i and C_j corresponding exactly to the horizontal tiling restrictions in $\mathcal{D}$. A similar argument shows that **(19)** handles the vertical constraints. Finally, we add a constraint specifying which tile covers the bottom left-hand square:

(20) $(0,0) \subseteq C_1$.

Let $\Delta_{\mathcal{D}}$ be the set of constraints $(\mathbf{15})$–$(\mathbf{20})$.

Summarizing the above argument, we have shown:

Lemma 17. *Let $k = 2^d$ for some $d \geq 1$, let $\mathcal{D}$ be a tiling system with tiles $c_1, \ldots, c_m$, and let Γ_d, $\Delta_{\mathcal{D}}$ be as defined above. Suppose $\mathfrak{A} \models \Gamma_d \cup \Delta_{\mathcal{D}}$. For x, y in the range $[0, k-1]$, set $f(x,y) = c_i$ if $\mathfrak{A} \models (x,y) \subseteq C_i$. Then f is a k-tiling for $\mathcal{D}$.*

Finally, we show that any k-tiling for $\mathcal{D}$ (where $k = 2^d$ for some $d \geq 1$) can be used to manufacture a structure satisfying $\Gamma_d \cup \Delta_{\mathcal{D}}$. Imagine we have such a k-tiling. In each grid square, place two points: an 'input' point and an 'output' point. Now connect these points by arrows as shown in figure 4 (the input points are coloured black, and the output points, white); we also assume each point is connected to itself. By inspection, this graph is reflexive and transitive, and thus defines a topological space, A, with closure operator defined in the familiar way.

To turn A into a structure $\mathfrak{A}$, we interpret the $X_1, \ldots, X_d$ and $Y_1, \ldots, Y_d$ so that the expressions (x,y) pick out the corresponding grid squares in the obvious way. In addition, we let the two points in each grid square (x,y) (under the correspondence just established) be in the extension of the variable C_i just in case (x,y) is tiled with c_i. It is then routine to check that the constraints $\Gamma_d \cup \Delta_{\mathcal{D}}$ are satisfied. Thus, we have,

Lemma 18. *Let $k = 2^d$ for some $d \geq 1$, let $\mathcal{D}$ be a tiling problem, and let Γ_d, $\Delta_{\mathcal{D}}$ be as defined above. Given a tiling of a $k \times k$ grid by $\mathcal{D}$, form the structure $\mathfrak{A}$ as just described. Then $\mathfrak{A} \models \Gamma_d \cup \Delta_{\mathcal{D}}$.*

Theorem 3. *The problem of determining the satisfiability of a set of constraints of types I, II and III is NEXPTIME-hard.*

Proof. Lemmas 17 and 18, and Theorem 2. □

Theorems 1 and 3 complete the proof that the problem $\mathcal{TCC}$-SAT is NEXPTIME-complete.

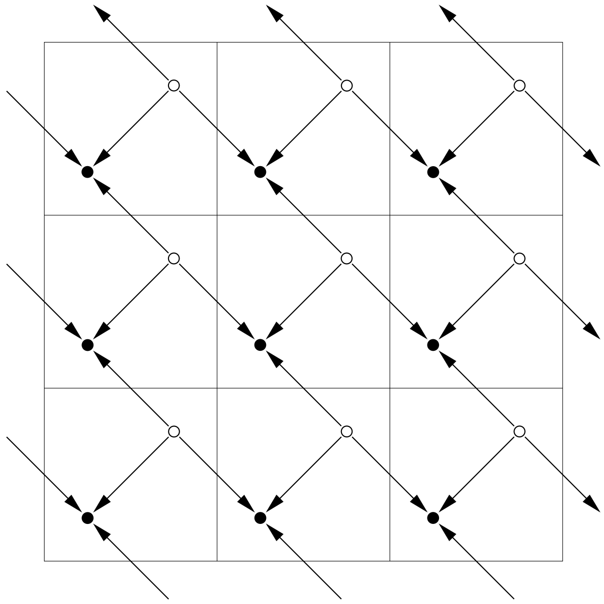

Figure 4. *Manufacturing a topological space from a tiling*

6. Relation to other work

Research into topological constraint languages began with the development of what has now come to be called RCC-theory. (RCC stands for 'region-connection calculus'; for a representative sample of this early work, see Randell, Cui and Cohn [RAN 92].) To relate RCC-theory and its ensuing developments to the present paper requires us occasionally to sacrifice historical accuracy for the sake of logical simplicity. However, none of these exegetical distortions affects any matter of substance.

Let $\mathbf{V}$ be a set of variables, as before, and let $\mathbf{R}$ denote the set of relation-symbols: DC, EC, PO, EQ, TPP and NTPP. (These symbols are mnemonics for their fixed interpretations: DC for "disconnected", EC for "externally connected", PO for "partial overlap", EQ for "equal", TPP for "tangential proper part", and NTPP for "non-tangential proper part".) Then we can define a topological constraint language—let us call it $\mathcal{T}_0$—as follows: atomic formulas of $\mathcal{T}_0$ are simply expressions of the form $R(V, V')$, where $R \in \mathbf{R}$ and $V, V' \in \mathbf{V}$; and formulas of $\mathcal{T}_0$ are Boolean combinations of atomic formulas.

For reasons that need not detain us here, topological constraint languages based on these relations are standardly interpreted only over *regular closed subsets* of topological spaces. (A set B is *regular closed* if it is equal to the closure of its interior—in the usual notation: $B = (B^0)^-$.) Formally, then, we take a $\mathcal{T}_0$-structure to be a pair $\langle A, I\rangle$ where A is a topological space and $I : \mathbf{V} \to \{B \in \mathbb{P}(A) \mid B = (B^0)^-\}$. Satisfaction for atomic formulas is then defined as follows:

$$
\begin{aligned}
\langle A, I\rangle &\models \mathsf{DC}(V_1, V_2) && \text{iff} \quad I(V_1) \cap I(V_2) = \emptyset \\
\langle A, I\rangle &\models \mathsf{EC}(V_1, V_2) && \text{iff} \quad I(V_1) \cap I(V_2) \neq \emptyset \wedge I(V_1)^0 \cap I(V_2)^0 = \emptyset \\
\langle A, I\rangle &\models \mathsf{PO}(V_1, V_2) && \text{iff} \quad I(V_1)^0 \cap I(V_2)^0 \neq \emptyset \wedge I(V_1) \cap -I(V_2) \neq \emptyset \wedge \\
& && \qquad -I(V_1) \cap I(V_2) \neq \emptyset \\
\langle A, I\rangle &\models \mathsf{EQ}(V_1, V_2) && \text{iff} \quad I(V_1) = I(V_2) \\
\langle A, I\rangle &\models \mathsf{TPP}(V_1, V_2) && \text{iff} \quad I(V_1) \cap (-I(V_2))^- \neq \emptyset \wedge I(V_1) \cap -I(V_2) = \emptyset \\
\langle A, I\rangle &\models \mathsf{NTPP}(V_1, V_2) && \text{iff} \quad I(V_1) \cap (-I(V_2))^- = \emptyset.
\end{aligned}
$$

The definition of satisfaction for arbitrary $\mathcal{T}_0$-formulas proceeds as expected. The problem $\mathcal{T}_0$-SAT is that of determining, for a given $\mathcal{T}_0$-formula, whether there exists a $\mathcal{T}_0$-structure satisfying it. (Readers familiar with "RCC-8" who are wondering what has happened to the relations TPPi and NTPPi are reminded that these primitives are obviously definable in $\mathcal{T}_0$ using TPP and NTPP.)

The language $\mathcal{T}_0$ is a modest liberalization of a topological constraint language which has received considerable attention in the literature (Bennett [BEN 96], Renz and Nebel [REN 99]). It follows from the results obtained by Renz and Nebel that $\mathcal{T}_0$-SAT is NP-complete. In fact, Renz [REN 00] provides a comprehensive account of the complexity of satisfiability for a fragment of $\mathcal{T}_0$ and its tractable sublanguages, obtained by restricting the kinds of disjunctions allowed in $\mathcal{T}_0$-formulas. We note in passing that, for the more restricted language studied by Nebel and Renz, the NP-hardness result is non-trivial; by contrast, $\mathcal{T}_0$ as just defined is visibly NP-hard.

One way to increase $\mathcal{T}_0$'s expressiveness is to incorporate function-symbols denoting operations on sets. Thus, Wolter and Zakharyaschev [WOL 00a] extend the usual RCC-language by allowing function-symbols denoting operations within the Boolean algebra of regular closed sets. Liberalizing Wolter and Zakharyaschev's syntax slightly, we may define a topological constraint language $\mathcal{T}_1$ extending $\mathcal{T}_0$ as follows. Terms in $\mathcal{T}_1$ are formed by applying the unary function-symbol * and the binary function-symbol $\cup$ to variables; atomic formulas in $\mathcal{T}_1$ are defined as expressions of the form $R(T, T')$, where $R \in \mathbf{R}$ and T, T' are terms; and formulas in $\mathcal{T}_1$ are Boolean combinations of atomic formulas. Semantically, a $\mathcal{T}_1$-structure is a pair $\langle A, I\rangle$ where A is a topological space and $I : \mathbf{V} \to \{B \in \mathbb{P}(A) \mid B = (B^0)^-\}$. The function I is extended from variables to terms by the rules $I(T^*) = (-I(T))^-$ and $I(T \cup T') = I(T) \cup I(T')$. Satisfaction and the problem $\mathcal{T}_1$-SAT are then defined in the obvious way. It turns out that $\mathcal{T}_1$-SAT is still NP-complete; thus, $\mathcal{T}_1$ represents an increase of expressive power over $\mathcal{T}_0$ without changing the complexity of satisfiability. This result appears to derive primarily from Wolter and Zakharyaschev's careful choice of primitive functions rather than on any special properties of the relations in $\mathbf{R}$. It is likely, therefore, that the result can be extended to richer sets of primitive relations.

What happens if the primitive functions are not restricted to the regular closed algebra? This question is partially answered by Nutt [NUT 99], who investigates the topological constraint language with the three function-symbols $\cap$, $-$, $^{-}$ and the single binary relation-symbol $=$, under the semantics given in section 3. (As with $\mathcal{TCC}$, variables are now interpreted as arbitrary subsets of some topological space—i.e. they need not be regular closed.) Thus, Nutt's language, which we might call $\mathcal{T}_2$, is the fragment of $\mathcal{TCC}$ involving only those atomic propositions expressing equality between two terms. There is no need to include in $\mathcal{T}_2$ the remaining primitives of $\mathbf{R}$, since these are definable using the resources already available. Nutt observes that determining satisfiability in $\mathcal{T}_2$ is essentially the same as determining satisfiability in the modal logic S4, and is thus PSPACE-complete. The translation of topological constraint languages into modal logics in fact originated Bennett's analysis of a language based on the RCC relations; Nutt was the first to present the semantics of topological constraint languages in a rigorous form, allowing a more precise statement of the relevant complexity result.

Though visibly more expressive than $\mathcal{T}_0$ or $\mathcal{T}_1$, $\mathcal{T}_2$ still does not allow us to bound the number of components of a region, or even to state that a region is connected at all. The results of this paper show that adding this capability raises the complexity from PSPACE-complete to NEXPTIME-complete. Thus, in the broader context of topological constraint languages, our result continues a general trend towards considering formalisms of ever greater expressiveness and complexity.

Having seen how $\mathcal{TCC}$ fits into a series of progressively complex topological constraint languages, it is natural to ask what lies beyond. Perhaps the most obvious extension to the expressive power of these languages is the introduction of quantification. The extension of the notion of satisfaction to quantified languages is completely standard, and need not be rehearsed here. Some results have indeed been obtained regarding such languages. However, because this work has been motivated largely by possible applications to spatial reasoning in Artificial Intelligence, most of these results restrict the domain of quantification to certain very well-behaved subsets of the single topological spaces $\mathbb{R}^2$ or $\mathbb{R}^3$. The satisfiability problem for $\mathbb{R}^2$ was shown to be undecidable by Dornheim [DOR 98]; consequently, investigations tend to concentrate on issues such as alternative models (Pratt and Lemon [PRA 97]), expressive power (Papadimitriou, Suciu and Vianu [PAP 96], Pratt and Schoop [PRA 00]) and axiomatization (Pratt and Schoop [PRA 98]). An analysis of these issues for $\mathbb{R}^3$ is given in Pratt and Schoop [PRA ng]. Little has been published on quantified topological languages interpreted over arbitrary topological spaces.

Acknowledgements

The author wishes to thank Paolo Torrini and the anonymous referees for their comments.

7. References

[BEN 96] BENNETT B., "Modal Logics for Qualitative Spatial Reasoning", *Journal of the Interest Group on Pure and Applied Logics*, vol. 4, 1996, p. 23–45.

[CLA 81] CLARKE B. L., "A calculus of individuals based on "connection"", *Notre Dame Journal of Formal Logic*, vol. 22, num. 3, 1981, p. 204–218.

[CLA 85] CLARKE B. L., "Individuals and points", *Notre Dame Journal of Formal Logic*, vol. 26, num. 1, 1985, p. 61–75.

[DOR 98] DORNHEIM C., "Undecidability of Plane Polygonal Mereotopology", COHN A. G., SCHUBERT L. K., SCHUBERT S. C., Eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR '98)*, San Francisco, CA, 1998, Morgan Kaufmann, p. 342-353.

[JON 97] JONSSON P., DRAKENGREN T., "A complete classification of tractability in RCC5", *Journal of Artificial Intelligence Research*, vol. 6, 1997, p. 211–221.

[LAG 22] DE LAGUNA T., "Point, line, and surface as sets of solids", *The Journal of Philosophy*, vol. 19, 1922, p. 449-461.

[NUT 99] NUTT W., "On the Translation of Qualitative Spatial Reasoning Problems into Modal Logics", BURGARD W., CHRISTALLER T., CREMERS A., Eds., *Advances in Artificial Intelligence, Proc. 23rd Annual German Conference on Artificial Intelligence, KI'99*, vol. 1701 of *Lecture Notes in Computer Science*, Berlin, 1999, Springer-Verlag, p. 113–124.

[PAP 94] PAPADIMITRIOU C. H., *Computational Complexity*, Addison-Wesley, Reading, MA, 1994.

[PAP 96] PAPADIMITRIOU C. H., SUCIU D., VIANU V., "Topological Queries in Spatial Databases", *Proceedings of PODS'96*, ACM, 1996, p. 81–92.

[PRA 97] PRATT I., LEMON O., "Ontologies for plane, polygonal mereotopology", *Notre Dame Journal of Formal Logic*, vol. 38, num. 2, 1997, p. 225-245.

[PRA 98] PRATT I., SCHOOP D., "A complete axiom system for polygonal mereotopology of the real plane", *Journal of Philosophical Logic*, vol. 27, num. 6, 1998, p. 621–658.

[PRA 00] PRATT I., SCHOOP D., "Expressivity in polygonal, plane mereotopology", *Journal of Symbolic Logic*, vol. 65, num. 2, 2000, p. 822–838.

[PRA ng] PRATT I., SCHOOP D., "Elementary polyhedral mereotopology", *Journal of Philosophical Logic*, , forthcoming.

[RAN 92] RANDELL D. A., CUI Z., COHN A. G., "A Spatial Logic Based on Regions and Connection", NEBEL B., RICH C., SWARTOUT W., Eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR '92)*, San Mateo, CA, 1992, Morgan Kaufmann, p. 165–176.

[REN 99] RENZ J., NEBEL B., "On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the Region Connection Calculus", *Artificial Intelligence*, vol. 108, 1999, p. 69-123.

[REN 00] RENZ J., "Qualitative Spatial Reasoning with Topological Information", PhD thesis, Albert-Ludwigs-Universität Freiburg im Breisgau, 2000.

[WHI 29] WHITEHEAD A. N., *Process and Reality*, The MacMillan Company, New York, 1929.

[WOL 00a] WOLTER F., ZAKHARYASCHEV M., "Spatial reasoning in RCC-8 with Boolean region terms", HORN W., Ed., *ECAI 2000: Fourteenth European Conference on Artificial Intelligence*, Amsterdam, 2000, IOS Press, p. 244-248.

[WOL 00b] WOLTER F., ZAKHARYASCHEV M., "Spatio-temporal representation and reasoning based on RCC-8", COHN A. G., GIUNCHIGLIA F., SELMAN B., Eds., *Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR-00)*, San Francisco, 2000, Morgan Kaufmann, p. 3–14.