The Complexity of Constraint Satisfaction Problems for Small Relation Algebras

M. Cristani [a]

[a] *Dipartimento di Informatica, Università di Verona, Cà Vignal 2, strada Le Grazie 15, I-37134 Verona*

R. Hirsch [b]

[b] *Department of Computer Science, University College London, Gower Street, London, WC1E 6BT*

The problems that deserve an attack
demonstrate it by a counterattack.
Paul Erdös

Abstract

Andréka and Maddux (1994, Notre Dame Journal of Formal Logic, 35(4)) classified the *small relation algebras* — those with at most 8 elements, or in other terms, at most 3 atomic relations. They showed that there are eighteen isomorphism types of small relation algebras, all representable. For each simple, small relation algebra they computed the *spectrum* of the algebra, namely the set of cardinalities of square representations of that relation algebra.

In this paper we analyze the computational complexity of the problem of deciding the satisfiability of a finite set of constraints built on any small relation algebra. We give a complete classification of the complexities of the general constraint satisfaction problem for small relation algebras. For three of the small relation algebras the constraint satisfaction problem is **NP**-complete, for the other fifteen small relation algebras the constraint satisfaction problem has cubic (or lower) complexity.

We also classify the complexity of the constraint satisfaction problem over *fixed finite representations* of any relation algebra. If the representation has size two or less then the complexity is cubic (or lower), but if the representation is square, finite and bigger than two then the complexity is **NP**-complete.

Key words: Relation algebra, constraint satisfaction problem, computational complexity.

Preprint submitted to Artificial Intelligence

1 Introduction

The study of relation algebra originates in the nineteenth century and constitutes, along with Frege's quantifier logic, the foundation of modern logic [1,2]. From the 1970s onwards, computer scientists working in planning [3,4] and temporal reasoning [5–11] rediscovered relation algebra. Later, scholars working in the field of Knowledge Representation, and specifically Spatial and Temporal Knowledge Representation, also used the formalism of relation algebra [10,12–18]. For them, the principal method of reasoning using a relation algebra was by checking the consistency of a set of constraints over that algebra. So this work became integrated with a wider study of constraint handling in computer science [19–24]. This problem of determining the satisfiability of a set of constraints over a given relation algebra is sometimes called the *network satisfaction problem*.

Examples of relation algebras include the Point Algebra, the Allen Interval Algebra, the Region Connected Calculus [25–27] and a Preference Reasoning Algebra [28]. The point algebra can be used to represent constraints over a linear flow of time, the Allen Interval Algebra expresses constraints between intervals in a linear flow of time, and so on.

A number of results were obtained. For some relation algebras, e.g. the point algebra (see [11, theorem 5.10]), tractable algorithms were obtained and shown to give a sound and complete method of testing consistency. For many algebras though, e.g. the Allen Interval Algebra, this consistency checking problem was shown to be **NP**-complete [21, theorems 2 and 3]. An investigation, very relevant to the work conducted here, is in [29] where some 'small algebras' are studied [2]. Further complexity analysis of various algebras can be found in [31]. Typically, it seems, the complexity of the constraint satisfaction problem for many relation algebras is **NP**-hard. For some pathological, finite relation algebras the problem can even be undecidable [32].

A systematic analysis of the complexity of the constraint handling for finite relation algebras is a challenge for those working in the area between algebraic logic and more practical computer science. So far we have only ad-hoc results.

Email addresses: `cristani@sci@sci.univr.it` (M. Cristani),
`R.Hirsch@cs.ucl.ac.uk` (R. Hirsch).

[1] We would like to thank Enrico Gregorio, Ian Hodkinson and Andrea Lucchini for hints and suggestions. We would also like to thank a lot Ivo Düntsch for having pointed us the relevant literature and Roger Maddux for his invaluable comments.

[2] As a matter of fact we found that the name *small* has a different meaning in [29] with respect to the use we have made here, which originates in [30]. In that case the algebras studied are in fact those generated by partial orders, therefore algebras with four atoms.

The purpose of this paper is to make a start on this complexity analysis by completely classifying the complexity of the constraint satisfaction problem for small relation algebras.

We hope this can be used as a criterion for the usability of such algebras. We handle separately the complexities of two different computational problems: the *general satisfaction problem* — roughly, is a given set of constraints (or equivalently, to use the other terminology, a given network) satisfiable in *some* representation of the relation algebra; and secondly, is a given set of constraints satisfiable in a *specified* representation of the relation algebra.

The paper is organized as follows: in Section 2 we define the terminology employed in the rest of the paper; in Section 3 we provide reference to the relevant literature; Section 4 is the main part, where we provide the results of the paper — see figure 3 for a summary of the complexity results established in this paper and elsewhere; Section 5 takes some conclusions and sketches further developments.

2 Terminology and definitions

Let us shortly recall the three main concepts we are going to use here:

- Relation Algebra;
- Constraint Set;
- Satisfiability Problem.

The very general notion of Relation Algebra we adopt here is based on Tarski [33].

DEFINITION 1 [Tarski] A *Relation Algebra* $\mathcal{A}$ (with domain A) is a nine-tuple

$$\mathcal{A} = (A, +, -, 0, 1, 1', {}^{\smile}, ;)$$

where

$$\begin{aligned} (A, +, -, 0, 1) &\text{ is a boolean algebra} && (1)\\ (x;y);z &= x;(y;z) && (2)\\ (x+y);z &= x;z + y;z && (3)\\ x;1' &= x && (4)\\ (x^{\smile})^{\smile} &= x && (5)\\ (x+y)^{\smile} &= x^{\smile} + y^{\smile} && (6) \end{aligned}$$

$$(x;y)^\smile = y^\smile ; x^\smile \tag{7}$$
$$x^\smile ; -(x;y) + -y = -y \tag{8}$$

Where we consider two different relation algebras ($\mathcal{A}, \mathcal{B}$ say) and we wish to distinguish the constants of the two, we may write $0_\mathcal{A}, 1'_\mathcal{A}, 1_\mathcal{B}$ etc. to indicate which relation algebra we are referring to.

A relation algebra with just one element (#1 below) is called *trivial*.

We write $a \bullet b$ as an abbreviation of $-(-a + -b)$, we write $a \leq b$ as an abbreviation of $a + b = b$ (or, equivalently, $a \bullet b = a$) and we write $a < b$ for $a \leq b \wedge b \not\leq a$. The element $0' = -1'$ is called the *diversity element*.

We name *atoms* (or *basic relations*) the minimal non-zero elements of a relation algebra with respect to $<$. A finite, non-trivial relation algebra always has atoms, and each element of a finite relation algebra is a finite sum of atoms.

Given two relation algebras, $\mathcal{A}, \mathcal{B}$ with domains A, B respectively, their *direct product* $\mathcal{A} \times \mathcal{B}$ is the relation algebra with domain $A \times B$ and operators defined by

$$\begin{aligned}
(a,b) + (a',b') &= (a + a', b + b') \\
(a,b) - (a',b') &= (a - a', b - b') \\
0_{\mathcal{A}\times\mathcal{B}} &= (0_\mathcal{A}, 0_\mathcal{B}) \\
1_{\mathcal{A}\times\mathcal{B}} &= (1_\mathcal{A}, 1_\mathcal{B}) \\
1'_{\mathcal{A}\times\mathcal{B}} &= (1'_\mathcal{A}, 1'_\mathcal{B}) \\
(a,b)^\smile &= (a^\smile, b^\smile) \\
(a,b);(a',b') &= (a;a', b;b')
\end{aligned}$$

for all $a, a' \in \mathcal{A}$ and $b, b' \in \mathcal{B}$.

An algebra is said to be *simple* iff the only congruences over this algebra are the identity and the total binary relation over the algebra or, equivalently, iff $1;x;1 = 1$ for all $x \neq 0$ in the algebra [34, theorem 4.10]. By [34, theorem 4.14] a relation algebra is simple iff it is not the direct product of two non-trivial algebras.

Given an algebra $\mathcal{A}$, and any subset X of the universe of $\mathcal{A}$, we indicate by $\mathfrak{Sg}^{(\mathcal{A})}X$ the subalgebra of $\mathcal{A}$ which is generated by X.

The set of all the relations defined on a set Δ (along with the operations of set union for $+$, set complement for $-$, the empty set for 0, the universal relation Δ^2 for 1, the identity relation $\{(x,x) : x \in \Delta\}$ for $1'$, the converse operator for $^\smile$ and composition of binary relations $|$ for $;$), is henceforth indicated by

$\mathfrak{Re}\Delta$. As a special case, for $n \in \mathbb{N}$, we write $\mathfrak{Re}n$ for the relation algebra of all binary relations over the domain $n = \{0, 1, \ldots, n-1\}$, namely $(\wp(n \times n), \cup, \cap \setminus, \emptyset, n \times n, 1', \smile, |)$. If E is any equivalence relation over Δ we write $\mathfrak{Re}E$ for the algebra of all subrelations of E: $(\wp(E), \cup, \cap, \setminus, \emptyset, E, 1', \smile, |)$. These uses of the $\mathfrak{Re}$ operator must be distinguished by the context. It is easy to check that $\mathfrak{Re}\Delta, \mathfrak{Re}E$ are relation algebras, for any set Δ and for any equivalence relation E.

We say that a relation algebra $\mathcal{A}$ is *proper* (or *concrete*), iff it is a subalgebra of $\mathfrak{Re}E$ for some equivalence relation E. We say that a relation algebra $\mathcal{A}$ is *representable* iff there is an isomorphism $\mathcal{M}$ from $\mathcal{A}$ to a subalgebra of $\mathfrak{Re}E$, for some equivalence relation E over some set M. Such an isomorphism is called a *representation* of $\mathcal{A}$ and M is the *domain* of the representation. If E is an equivalence relation with a single equivalence class (i.e. E is the universal relation over its domain, $E = M \times M$) and $\mathcal{M}$ is an isomorphism from $\mathcal{A}$ to a subalgebra of $\mathfrak{Re}E$ then we say $\mathcal{M}$ is a *square* representation of $\mathcal{A}$. Every representable simple relation algebra has a square representation [34, theorems 4.26 and 4.28].[3]

We follow the model-theoretic convention of identifying a structure with its domain. Thus we may write $x \in \mathcal{M}$ to mean that x belongs to the domain of $\mathcal{M}$. We write $|\mathcal{M}|$ for the cardinality of the domain of $\mathcal{M}$. When there is a possibility of ambiguity we may write $\mathsf{dom}(\mathcal{M})$ for the domain of $\mathcal{M}$. If $a \in \mathcal{A}$ we write $a^{\mathcal{M}}$ for the interpretation of a in $\mathcal{M}$, namely (regarding $\mathcal{M}$ as an isomorphism) $\mathcal{M}(a)$.

Two representations $\mathcal{M}, \mathcal{N}$ of the relation algebra $\mathcal{A}$ are said to be *base isomorphic* if there is a bijection $\iota : \mathsf{dom}(\mathcal{M}) \to \mathsf{dom}(\mathcal{N})$ such that for all $a \in \mathcal{A}$ and all $x, y \in \mathsf{dom}(\mathcal{M})$ we have $(x, y) \in a^{\mathcal{M}}$ iff $(\iota(x), \iota(y)) \in a^{\mathcal{N}}$.

Let us now define the notion of constraint over a Relation Algebra. In order to make it familiar to the wider computer science community working in constraints, we take our definition from Tsang's book [22] and embed it in Tarski's theory exposed above. However, a set of constraints, defined next, is essentially the same as a *network*, and satisfiability of constraints is the same

[3] We restrict to square representations of simple relation algebras for the following reason. Let h, g be representations of the simple relation algebra $\mathcal{A}$, over domains M, N respectively. Let $\Delta \cap \Delta' = \emptyset$. Define another representation f over the domain $M \cup N$ by $f(a) = h(a) \cup g(a)$ $(a \in \mathcal{A})$. Thus if $\mathcal{A}$ has a representation of size n then it also has representations of size kn for $k = 1, 2, 3, \ldots$. In this sense, every representable relation algebra has infinite representations. We do not wish to count these all representations separately as they are not 'essentially different'. The disadvantage of the restriction to square representations is that non-simple relation algebras do not possess square representations, and so, following [30], we do not define the spectrum of a non-simple relation algebra.

as the satisfiability of a network [31].

DEFINITION 2

(1) Given a Relation Algebra $\mathcal{A}$ and a set of variable names $\mathcal{X} = \{x_0, x_1, \ldots, x_n, \ldots\}$, the logical expression ax_ix_j, where a is an element of $\mathcal{A}$, and x_i and x_j are elements of $\mathcal{X}$, is called a *constraint* over $\mathcal{A}$.
(2) Given a representation $\mathcal{M}$ of $\mathcal{A}$, a *variable assignment* v is a partial map from $\mathcal{X}$ to $\mathsf{dom}(\mathcal{M})$. The variable assignment v *satisfies* the constraint ax_ix_j iff $x_i, x_j \in \mathsf{dom}(v)$ and $(v(x_i), v(x_j)) \in a^{\mathcal{M}}$.
(3) A set of constraints Ξ over $\mathcal{A}$ is $\mathcal{M}$-satisfiable iff there is a variable assignment v satisfying all the constraints in Ξ.
(4) The problem of deciding the satisfiability of a finite sets of constraints on a Relation Algebra $\mathcal{A}$, for a given representation $\mathcal{M}$ of $\mathcal{A}$, is henceforth indicated by $\mathcal{M} - SAT(\mathcal{A})$.
(5) The problem, instead, of finding whether a set of constraints over a relation algebra $\mathcal{A}$ can be satisfied in *some* (i.e. at least one) representation of $\mathcal{A}$ is called the Gen-$SAT(\mathcal{A})$ problem.

If Ax_ix_j, Bx_jx_k and Cx_ix_k belong to a set of constraints Ξ then let Ξ' be the same as Ξ but with the constraint Cx_ix_k replaced by $(C \bullet (A; B))x_ix_k$. Clearly Ξ and Ξ' are *equivalent* in that a variable assignment satisfies Ξ if and only if it satisfies Ξ'.

DEFINITION 3 A set of constraints is *closed* if whenever (i) Ax_ix_j, Bx_jx_k are in the set of constraints there is also a constraint Cx_ix_k in the set and $C \leq A; B$, (ii) if Ax_ix_j is in the set of constraints then $A^{\smile}x_jx_i$ is also in the set and (iii) for each variable x_i occuring in a constraint in Ξ, there is a constraint ex_ix_i, for some $e \leq 1'$.

The so-called *propagation algorithm* [6, § 4.1] takes a finite set of constraints, adds in the constraint $1'x_ix_i$ for each variable x_i occuring in a constraint, replaces the constraint Ax_ix_j by $(A \bullet B^{\smile})x_ix_j$ whenever Ax_ix_j and Bx_jx_i are constraints, and then repeatedly replaces a constraint Cx_ix_k by $(C \bullet (A; B))x_ix_k$ whenever $C \bullet (A; B) \neq C$ and Ax_ix_j and Bx_jx_k are in the set, until we obtain an equivalent, closed set of constraints. If this closed set of constraints contains a constraint $0x_ix_j$ (some variables x_i, x_j) then it is clearly unsatisfiable. But if all the constraints are non-zero, we cannot in general be sure whether the constraints are satisfiable or not. Kautz defined a closed set of constraints over the Allen Interval Algebra, none of which were zero, but which nevertheless were unsatisfiable (this set of constraints is in [6, figure 5]). However, as we shall see, for some representations of some algebras the propagation algorithm does solve the satisfiability problem. The propagation algorithm runs in cubic time for finite relation algebras [21, theorem 1], though for infinite relation algebras the propagation algorithm is not guaranteed to terminate.

DEFINITION 4 A set of constraints is called *non-zero* if none of the constraints has the form $0xy$.

A representation $\mathcal{U}$ of a relation algebra $\mathcal{A}$ is called *universal* if every closed, non-zero set of constraints is satisfiable in $\mathcal{U}$.

PROPOSITION 5 *Let $\mathcal{A}$ be a relation algebra*

- *Let $\mathcal{M}$ be a representation of $\mathcal{A}$. The decision problem $\mathcal{M}-SAT(\mathcal{A})$ reduces in cubic time to the problem of telling whether a closed set of constraints is satisfiable in $\mathcal{M}$.*
- *Gen-$SAT(\mathcal{A})$ reduces in cubic time to the problem of telling whether a closed set of constraints is satisfiable in some representation of $\mathcal{A}$.*

PROOF:

In each case the required reduction is the propagation algorithm. □

LEMMA 6 *If a relation algebra $\mathcal{A}$ has a universal representation $\mathcal{U}$ then $\mathcal{U} - SAT(\mathcal{A})$ and Gen-$SAT(\mathcal{A})$ have at worst cubic complexity.*

PROOF:

Take an arbitrary set of constraints Ξ over $\mathcal{A}$. Use the propagation algorithm to find an equivalent, closed set of constraints Ξ'. This takes cubic time. Clearly, if any (hence by closure all) constraint in Ξ' has the form $0xy$ then Ξ' is unsatisfiable in any representation (and by equivalence, Ξ is also unsatisfiable). If Ξ' consists of non-zero constraints then by universality, it is satisfiable in $\mathcal{U}$. Thus Ξ is satisfiable in some representation of $\mathcal{A}$ iff it is satisfiable in $\mathcal{U}$ iff each constraint in Ξ' is non-zero. □

We will focus on the complexities of the simple relation algebras, partly because of the following lemma.

LEMMA 7

(1) *Let $\mathcal{M}, \mathcal{N}$ be representations, with disjoint domains, of relation algebras $\mathcal{A}, \mathcal{B}$ respectively. The* union representation $\mathcal{M} \cup \mathcal{N}$ *of the direct product $\mathcal{A} \times \mathcal{B}$, has as its domain* $\mathsf{dom}(\mathcal{M}) \cup \mathsf{dom}(\mathcal{N})$, *and is defined by*

$$(a, b)^{\mathcal{M} \cup \mathcal{N}} = a^{\mathcal{M}} \cup b^{\mathcal{N}}$$

for $a \in \mathcal{A}$, $b \in \mathcal{B}$.

(2) *Conversely, let $\mathcal{L}$ be a representation of $\mathcal{A} \times \mathcal{B}$. The domain of $\mathcal{L}$ can be partitioned into two parts, M and N, such that the map $a \mapsto (a, 0)^{\mathcal{L}}$*

(for $a \in \mathcal{A}$) is a representation of $\mathcal{A}$ over the domain M and the map $b \mapsto (0, b)^{\mathcal{L}}$ (for $b \in \mathcal{B}$) is a representation of $\mathcal{B}$ over the domain N. $\mathcal{L}$ is base-isomorphic to $\mathcal{M} \cup \mathcal{N}$.

(3) *A representation $\mathcal{M}$ of a simple relation algebra $\mathcal{A}$ is a disjoint union of square representations of $\mathcal{A}$. These square representations are called the* square components *of $\mathcal{M}$.*

(4) *Let $\mathcal{M}, \mathcal{N}$ be representations of $\mathcal{A}, \mathcal{B}$ respectively. The complexity of $(\mathcal{M} \cup \mathcal{N}) - SAT(\mathcal{A} \times \mathcal{B})$ is the maximum of the complexities of $\mathcal{M} - SAT(\mathcal{A})$ and $\mathcal{N} - SAT(\mathcal{B})$.*

(5) *The complexity of Gen-$SAT(\mathcal{A} \times \mathcal{B})$ is the maximum of the complexities of Gen-$SAT(\mathcal{A})$ and Gen-$SAT(\mathcal{B})$.*

(6) *Let $\mathcal{A}$ be a simple relation algebra with finitely many simple, non-base-isomorphic square representations. The complexity of $\mathcal{M} - SAT(\mathcal{A})$ is the maximum of the complexities of $\mathcal{M}_i - SAT(\mathcal{A})$ over the square components $\mathcal{M}_i$ of $\mathcal{M}$.*

PROOF:

The first three parts are easy to verify, and are well-known (see [35, lemma 3.7]).

For the fourth part, let Ξ be any set of constraints over $\mathcal{A} \times \mathcal{B}$. Define an equivalence relation $\sim$ over the variables occuring in Ξ to be smallest equivalence relation containing all pairs (x_i, x_j) where there is a constraint $(a, b)x_i x_j$ in Ξ. Computing $\sim$ takes linear time. For each equivalence class α let Ξ_α be the restriction of Ξ to the constraints using only variables in α. Clearly Ξ is satisfiable in $\mathcal{M} \cup \mathcal{N}$ iff Ξ_α is satisfiable in $\mathcal{M} \cup \mathcal{N}$ for each equivalence class α. Also, Ξ_α is satisfiable in $\mathcal{M} \cup \mathcal{N}$ iff either Ξ_α is satisfiable in $\mathcal{M}$ or Ξ_α is satisfiable in $\mathcal{N}$. The reason for this is that if $m \in \mathcal{M}$ and $n \in \mathcal{N}$ then $(m, n) \notin 1^{\mathcal{M} \cup \mathcal{N}}$, so any variable assignment satisfying Ξ_α must map all the variables occuring in Ξ_α into $\mathcal{M}$ or all these variables into $\mathcal{N}$. This proves the fourth part of the lemma. The proof of the fifth part of the lemma is similar and the last part follows. $\square$

This lemma will allow us to calculate the complexities of algebras #3, #6, #7 and #8, later on.

3 Previous work

In this paper we employ the formalism of relation algebra as used in the computer science literature. For a general reference to relation algebras see [35–37].

There exist eighteen isomorphism types of relation algebra with no more than eight relations. Of these eighteen, one is trivial and four are non-simple. In their paper [30], Andrèka and Maddux have summarized the eighteen types, and proved that the eighteen isomorphism types of relation algebras with no more than eight elements are all representable. They established the spectra of the simple, small algebras, namely the sets of cardinalities of the square representations, see figure 1.

To describe the small relation algebras, [30] provides:

- A *representative* of each algebra, namely a representation of the algebra. This representative will be a representation of the algebra of smallest possible size.
- The atoms of the algebra.
- The composition table of the atoms.

Figure 1 summarizes the representatives and atoms of the eighteen isomorphism types. These results are from [30] where the following notation is adopted (this notation is not essential for reading the current paper): For any natural numbers $\kappa < \alpha < \omega$ and $\lambda < \beta < \omega$

- $\mathsf{P}^{\alpha}_{\kappa} = \{(\lambda, \mu) : \lambda, \mu < \alpha, \mu - \lambda \equiv \kappa (mod \alpha)\}$
- $\mathsf{Q}^{\alpha}_{\kappa} = \mathsf{P}^{\alpha}_{\kappa} \cup (\mathsf{P}^{\alpha}_{\kappa})^{-1}$
- $\mathsf{Q}^{\alpha,\beta}_{\kappa,\lambda} = \{((\mu, \nu), (\zeta, \eta)) : (\mu, \zeta) \in \mathsf{Q}^{\alpha}_{\kappa}, (\nu, \eta) \in \mathsf{Q}^{\lambda}_{\beta}\}$

Figure 2 gives the composition tables of the 18 small relation algebras and is taken from [30].

4 Computational analysis of small relation algebras

We have two complexity issues to consider. In this section we firstly check the complexity of $\mathcal{M} - SAT(\mathcal{A})$ for a square reprsentation $\mathcal{M}$ of a relation algebra $\mathcal{A}$. For finite $\mathcal{M}$ this turns out to be easily characterised, see theorem 9 below. For infinite $\mathcal{M}$ the problem remains open. Secondly, we will check the complexity of Gen-$SAT(\mathcal{A})$ for each of the eighteen small relation algebras.

Lets start with the very small relation algebras.

LEMMA 8 *Each of the following algebras has a square, universal representation (see definition 4). In each case this representation is unique up to base isomorphism, i.e. every square representation of the relation algebra is base isomorphic to the given universal representation: algebras #1, #2, #4 and* $\mathfrak{Re}2$*.*

Hence, by lemma 6, Gen$-SAT()$ *has at worst cubic complexity for these algebras. In fact, Gen-*SAT*(#1) can be solved in constant time and Gen-*SAT*(#2)*

No.	Representative	Atoms	Spectrum
#1	$\mathfrak{Re}0$	No atoms	$\{0\}$
#2	$\mathfrak{Re}1$	$1'$	$\{1\}$
#3	$\mathfrak{Re}1 \times \mathfrak{Re}1$	e_0, e_1	–
#4	$\mathfrak{Gg}^{(\mathfrak{Re}2)}\emptyset$	$1', 0'$	$\{2\}$
#5	$\mathfrak{Gg}^{(\mathfrak{Re}3)}\emptyset$	$1', 0'$	$\{\kappa : \kappa \geq 3\}$
#6	$\mathfrak{Re}1 \times \mathfrak{Re}1 \times \mathfrak{Re}1$	e_0, e_1, e_2	–
#7	$\mathfrak{Re}1 \times \mathfrak{Gg}^{(\mathfrak{Re}2)}\emptyset$	$e_0, e_1, 0'$	–
#8	$\mathfrak{Re}1 \times \mathfrak{Gg}^{(\mathfrak{Re}3)}\emptyset$	$e_0, e_1, 0'$	–
#9	$\mathfrak{Gg}^{(\mathfrak{Re}3)}\{\mathsf{P}^3_1\}$	$1', a, a^\smile$	$\{3\}$
#10	$\mathfrak{Gg}^{(\mathfrak{Re}\mathbb{Q})}\{<\}$	$1', a, a^\smile$	$\{\kappa : \kappa \geq \aleph_0\}$
#11	$\mathfrak{Gg}^{(\mathfrak{Re}7)}\{\mathsf{P}^7_1 \cup \mathsf{P}^7_2 \cup \mathsf{P}^7_4\}$	$1', a, a^\smile$	$\{7\} \cup \{\kappa : \kappa \geq 9\}$
#12	$\mathfrak{Gg}^{(\mathfrak{Re}4)}\{\mathsf{Q}^4_1\}$	$1', a, b$	$\{\kappa : \kappa \geq 6\}$
#13	$\mathfrak{Gg}^{(\mathfrak{Re}6)}\{\mathsf{Q}^6_2\}$	$1', a, b$	$\{\kappa : \kappa \geq 6\}$
#14	$\mathfrak{Gg}^{(\mathfrak{Re}6)}\{\mathsf{Q}^6_3\}$	$1', a, b$	$\{2\kappa : \kappa \geq 3\}$
#15	$\mathfrak{Gg}^{(\mathfrak{Re}9)}\{\mathsf{Q}^9_3\}$	$1', a, b$	$\{\kappa : \kappa \geq 9\}$
#16	$\mathfrak{Gg}^{(\mathfrak{Re}5)}\{\mathsf{Q}^5_1\}$	$1', a, b$	$\{5\}$
#17	$\mathfrak{Gg}^{(\mathfrak{Re}8)}\{\mathsf{Q}^8_1 \cup \mathsf{Q}^8_4\}$	$1', a, b$	$\{\kappa : \kappa \geq 8\}$
#18	$\mathfrak{Gg}^{(\mathfrak{Re}3\times 3)}\{\mathsf{Q}^{3,3}_{1,1}\}$	$1', a, b$	$\{\kappa : \kappa \geq 9\}$

Fig. 1. The eighteen isomorphisms types with one representative, list of atoms and spectrum, from [30].

takes linear time.

PROOF:

The notion of a representation for algebra #1 is pathological — the only representation for this algebra has an empty domain. There aren't any non-zero constraints over this algebra (it has only one element and that element is zero) so the only non-zero closed set of constraints is the empty set of constraints, which is satisfiable. Any non-empty set of constraints is unsatisfiable for the trivial reason that we cannot assign the variables to any points in the domain. Checking if a set of constraints is empty takes just constant time.

If $\mathcal{M}$ is a square representation of algebra #2 then it has just one point in

Algebra#1 has no atoms

#2	$1'$
$1'$	$1'$

#3	e_0	e_1
e_0	e_0	0
e_1	0	e_1

#4	$1'$	$0'$
$1'$	$1'$	$0'$
$0'$	$0'$	$1'$

#5	$1'$	$0'$
$1'$	$1'$	$0'$
$0'$	$0'$	1

#6	e_0	e_1	e_2
e_0	e_0	0	0
e_1	0	e_1	0
e_2	0	0	e_2

#7	e_0	e_1	$0'$
e_0	e_0	0	0
e_1	0	e_1	$0'$
$0'$	0	$0'$	e_1

#8	e_0	e_1	$0'$
e_0	e_0	0	0
e_1	0	e_1	$0'$
$0'$	0	$0'$	$e_1 + 0'$

#9	$1'$	a	$a^\smile$
$1'$	$1'$	a	$a^\smile$
a	a	$a^\smile$	$1'$
$a^\smile$	$a^\smile$	$1'$	a

#10	$1'$	a	$a^\smile$
$1'$	$1'$	a	$a^\smile$
a	a	a	1
$a^\smile$	$a^\smile$	1	$a^\smile$

#11	$1'$	a	$a^\smile$
$1'$	$1'$	a	$a^\smile$
a	a	$0'$	1
$a^\smile$	$a^\smile$	1	$0'$

#12	$1'$	a	b
$1'$	$1'$	a	b
a	a	$1'$	b
b	b	b	$-b$

#13	$1'$	a	b
$1'$	$1'$	a	b
a	a	$-b$	b
b	b	b	$-b$

#14	$1'$	a	b
$1'$	$1'$	a	b
a	a	$1'$	b
b	b	b	1

#15	$1'$	a	b
$1'$	$1'$	a	b
a	a	$-b$	b
b	b	b	1

#16	$1'$	a	b
$1'$	$1'$	a	b
a	a	$-a$	$0'$
b	b	$0'$	$-b$

#17	$1'$	a	b
$1'$	$1'$	a	b
a	a	$-a$	$0'$
b	b	$0'$	1

#18	$1'$	a	b
$1'$	$1'$	a	b
a	a	1	$0'$
b	b	$0'$	1

Fig. 2. The composition tables for the atoms of the 18 small relation algebras, from [30].

its domain [30, theorem 1]. Any non-zero set of constraints is closed and is satisfiable in $\mathcal{M}$ — just map all variables to the single point in the domain of $\mathcal{M}$. Checking to see if all constraints in a set of constraints are non-zero takes just linear time.

Algebra #4 has just one square representation, $\mathcal{M}$ say (up to base isomorphism) [30, theorem 1]. Let the domain of this representation be $\{0, 1\}$.

We show that $\mathcal{M}$ is a universal representation of algebra #4. So, let Ξ be any closed, non-zero set of constraints over algebra #4. Define an assignment h to the variables, satisfying all the constraints, as follows. Let $h(x_0) = 0$. Suppose we have defined h on $x_0, x_1, \ldots, x_{k-1}$ (some $k < n$) such that all constraints involving just these first k variables are satisfied. Next we define h on x_k. If there is a constraints $1'x_k x_j$ or $1'x_j x_k$ (some $j < k$) then let $h(x_k) = h(x_j)$. This is well-defined, because if $1'x_j x_k$ and $1'x_i x_k$ (some $i, j < k$) then since Ξ is closed we must have a constraint $1'x_i x_j$ in Ξ, and so $h(x_i) = h(x_j)$. Similarly, if there is $j < k$ and a constraint $0'x_j x_k$ or $0'x_k x_j$ then define $h(x_k)$ to be the unique domain element not equal to $h(x_j)$. Again, this is well-defined by closure of Ξ. If there are no such constraints involving x_k and x_j (some $j < k$) then, arbitrarily, let $h(x_k) = 0$. Thus we can define h on all the variables and satisfy all the constraints in Ξ. This shows that Ξ is satisfiable in $\mathcal{M}$, so $\mathcal{M}$ is a universal representation of algebra #4.

The algebra $\mathfrak{Re}2$ also has just one square representation (itself) upto base-isomorphism. As in the previous case, we show that this is a universal representation. The domain of our representation is $\{0, 1\}$. Let $\Xi = \{C_{ij} x_i x_j : i, j < n\}$ be a closed, non-zero set of constraints over the variables $x_0, \ldots, x_{n-1}$, where $C_{ij} \in \mathfrak{Re}2$ for $i, j < n$. (If there is no constraint between x_i and x_j for some $i, j < n$ we just let $C_{ij} = 1$.) Closure of the constraints means that $C_{ij}; C_{jk} \geq C_{ik}$, for $i, j, k < n$. We must show that these constraints are satisfiable in $\mathfrak{Re}2$, considered as a representation of itself. Suppose, inductively, that we have defined an assignment h on the variables $x_i : i < k$ (some $k < n$) such that all constraints $C_{ij} x_i x_j : i, j < k$ are satisfied by h. We must now define h on x_k. The two alternatives are 0 and 1. If $h(x_k) = 0$ is consistent with all constraints $C_{jk}(x_j, x_k) \in \Xi' : j \leq k$ then we let $h(x_k) = 0$. Otherwise, there is a constraint $C_{jk}(x_j, x_k)$ such that

$$(h(x_j), 0) \notin C_{jk}^{\mathcal{M}} \tag{9}$$

In this case we let $h(x_k) = 1$. We claim that $(h(x_i), 1) \in C_{ik}^{\mathcal{M}}$ for all $i \leq k$. If this were false, there would be $i \leq k$ and

$$(h(x_i), 1) \notin C_{ik}^{\mathcal{M}} \tag{10}$$

But then $(h(x_i), h(x_j)) \in (C_{ik}; C_{kj})^{\mathcal{M}}$ implies there exists $z \in \{0, 1\}$ with $(h(x_i), z) \in C_{ik}^{\mathcal{M}}$ and $(z, h(x_j)) \in C_{kj}^{\mathcal{M}}$. The possibility $z = 0$ is contradicted by equation 9, while $z = 1$ is contradicted by equation 10. Thus we can extend h, one variable at a time, till it is defined on all n variables, and all constraints in Ξ will be satisfied. Hence $\mathfrak{Re}2$ is a universal representation of itself.

$\square$

We can now deal with $\mathcal{M}-SAT(\mathcal{A})$ for all finite representations of any relation algebra $\mathcal{A}$.

THEOREM 9 *Given a representable relation algebra $\mathcal{A}$ and a finite representation $\mathcal{M}$ of $\mathcal{A}$:*

(1) if $|\mathcal{M}| \leq 2$, then $\mathcal{M} - SAT(\mathcal{A})$ is solvable in cubic time. More specifically, $\mathcal{M} - SAT(\#1)$ can be solved in constant time, $\mathcal{M} - SAT(\#2)$ and $\mathcal{M} - SAT(\#3)$ can be solved in linear time, and $\mathcal{M} - SAT(\#4)$ and $\mathcal{M} - SAT(\mathfrak{Re}2)$ can be solved in cubic time, for any $\mathcal{M}$ with $|\mathcal{M}| \leq 2$.

(2) if $\mathcal{M}$ is square (so $\mathcal{A}$ is simple) and $|\mathcal{M}| > 2$ then $\mathcal{M} - SAT(\mathcal{A})$ is **NP***-complete.*

PROOF:

Suppose that $\mathcal{A}$ is a relation algebra and assume that $\mathcal{M}$ is a finite representation of $\mathcal{A}$. We prove the points of the claim separately.

Point 1

For point 1, observe that the only relation algebras with representations of size two or less are #1, #2, #3, #4 and $\mathfrak{Re}2$. By lemma 8, Gen-$SAT(\#1)$ can be solved in constant time. Since algebra #1 has a unique (empty) representation $\mathcal{M}$, $\mathcal{M} - SAT(\#1)$ also can be solved in constant time. Similarly, lemma 8 proves that Gen-$SAT(\#2)$ and $\mathcal{M} - SAT(\#2)$ can be solved in linear time, for any representation $\mathcal{M}$ of algebra #2. Lemmas 8 and 6 show that Gen-$SAT(\#4)$, $\mathcal{M} - SAT(\#4)$, Gen-$SAT(\mathfrak{Re}2)$ and $\mathcal{M} - SAT(\mathfrak{Re}2)$ can be solved in cubic time, for any square representaiton $\mathcal{M}$ of either algebra. A non-square representation of either of these two algebras must have size at least four.

Algebra #3 is the direct product $\mathfrak{Re}1 \times \mathfrak{Re}1$. Lemma 7 shows that Gen-$SAT(\#3)$ is solvable in linear time.

Point 2

To prove point 2 let $|\mathcal{M}| = k > 2$. We reduce the graph k-colourability problem to $\mathcal{M} - SAT$. Since k-colourability of graphs is known to be **NP**-complete for $k \geq 3$ [38] or [39, page 191], this will prove that $\mathcal{M} - SAT$ is **NP**-hard. It is easy to see that $\mathcal{M}$-SAT is in **NP**, thus it will follow that $\mathcal{M} - SAT$ is **NP**-complete.

An instance of the k-colourability problem is an undirected graph G. G is a yes-instance if it is possible to assign each node i of G one of k colours, $col(i) \in \{0, 1, \ldots, k-1\}$, in such a way that if (i, j) is an edge of G then $col(i) \neq col(j)$. G is a no-instance otherwise. For the reduction, let G be a graph. We reduce G to a set of constraints $\Xi(G)$. To write these constraints we use a distinct variable x_i for each node i of G. Let

$$\Xi(G) = \{(0' x_i x_j) : (i, j) \text{ is an edge of } G\}$$

We can check that this is a correct reduction by identifying the k colours with the points in the domain of $\mathcal{M}$. □

From theorem 9:

COROLLARY 10

(1) *The following are* **NP***-complete:* $\mathcal{M} - SAT(\#9)$*,* $\mathcal{M} - SAT(\#12)$*,* $\mathcal{M} - SAT(\#16)$*, for any representation* $\mathcal{M}$*.*
(2) *The following are* **NP***-complete:* $\mathcal{M} - SAT(\#5)$*,* $\mathcal{M} - SAT(\#11)$*,* $\mathcal{M} - SAT(\#13)$*,* $\mathcal{M} - SAT(\#14)$*,* $\mathcal{M} - SAT(\#15)$*,* $\mathcal{M} - SAT(\#17)$*,* $\mathcal{M} - SAT(\#18)$*, for any finite representation of the algebra.*
(3) *The following are* **NP***-complete: Gen-*$SAT(\#9)$*, Gen-*$SAT(\#12)$*, Gen-*$SAT(\#16)$*.*

PROOF:

The first two parts are direct from theorem 9.

For the last part, observe that each of the following algebras has only one square representation, up to base isomorphism: #9 [30, theorem 1], #12 [30, theorem 4(i)], #16 [30, theorem 5]. It follows, for each of these algebras, that $\mathcal{M} - SAT(\#\mathrm{n})$ is identical to Gen-$SAT(\#\mathrm{n})$.

□

LEMMA 11 *Any infinite representation of algebra #5 is universal.*

PROOF:

Let Ξ be a closed, non-zero set of constraints and let $\mathcal{M}$ be any infinite representation. Suppose inductively that h is an assignment of the variables $x_0, \ldots, x_{k-1}$ (some k) into $\mathcal{M}$ such that all the constraints in Ξ involving only these k variables are satisfied. We extend h to x_k. If Ξ contains a constraint $1'x_i x_k$ (some $i < k$) let $h(x_k) = h(x_i)$. This is well-defined, by closure of Ξ. Otherwise, let $h(x_k)$ be any point in the domain of $\mathcal{M}$, with $h(x_k) \notin \{h(x_i) : i < k\}$. Since $\mathcal{M}$ is infinite, this can be done. Clearly, constraints from Ξ involving $x_0, \ldots, x_k$ are still satisfied by this. So we can extend h until it is defined on all the variables and satisfies all of Ξ. □

Hence by lemma 6, if $\mathcal{M}$ is an infinite representation of algebra #5 then $\mathcal{M} - SAT(\#5)$ and Gen-$SAT(\#5)$ have cubic complexities.

The complexity of algebra #10 has been analyzed in various papers in particular in the Temporal Reasoning community (see [19,21,40]).

THEOREM 12 ([11, theorem 4.1]) *Every representation of algebra #10 is elementarily equivalent to the representative based on the rational numbers, given in figure 1. Gen-*$SAT(\#10)$ *and* $\mathcal{M} - SAT(\#10)$ *(any representation* $\mathcal{M}$*) are solvable on deterministic machines in cubic time.*

We will construct universal representations of algebras #11, #17 and #18. In each case the universal representation will be based on a *random graph*.

PROPOSITION 13 *Countable graphs* $\mathbf{R}, \mathbf{T}$ *and* $\mathbf{N}$ *exist with the following properties.*

$\mathbf{R}$ *is an undirected graph. Every finite undirected graph embeds into* $\mathbf{R}$. $\mathbf{R}$ *is* ultrahomogeneous *— every finite partial isomorphism*[4] *from* $\mathbf{R}$ *to itself extends to an automorphism of* $\mathbf{R}$.

$\mathbf{T}$ *is a* tournament *(i.e. a directed graph with no reflexive edges such that for all pairs of distinct nodes* $x, y \in \mathbf{T}$ *exactly one of* (x, y) *and* (y, x) *is an edge of* $\mathbf{T}$*). Every finite tournament embeds in* $\mathbf{T}$. $\mathbf{T}$ *is ultrahomogenous.*

$\mathbf{N}$ *is an undirected, triangle-free graph. Every finite triangle-free graph embeds into* $\mathbf{N}$. $\mathbf{N}$ *is ultrahomogeneous.*

PROOF:

These graphs can be constructed in two ways — as *random constructions* or as *Fraïssé limits.* See [41, §7.1 and §7.4] or [42, §2.10] for a discussion about these constructions. In the former approach, we can construct the graph $\mathbf{R}$ as follows. Start with countably many nodes. For each pair of nodes x and y, include (x, y) as an edge with probability $\frac{1}{2}$. The resulting graph will be isomorphic to $\mathbf{R}$ with probability 1. The tournament $\mathbf{T}$ can be constructed as a random tournament in this way too. The triangle-free graph $\mathbf{N}$ can also be constructed randomly, but you have to be careful with the construction and fix it so that no triangles are included.

Here we outline in slightly more detail the second approach. To construct $\mathbf{R}$ as a Fraïssé limit, you have to check that the class $\mathcal{F}$ of all finite, undirected graphs has the following three properties.

Hereditary Property If $A \in \mathcal{F}$ and B is an induced subgraph of A (more generally A is a finite substructure of B) then $B \in \mathcal{F}$

Joint Embedding Property If $A, B \in \mathcal{F}$ then there is $C \in \mathcal{F}$ such that A and B embed in C

Amalgamation Property If $A, B, C \in \mathcal{F}$ and there are embeddings $e : A \to B$ and $f : A \to C$ then there is $D \in \mathcal{F}$ and embeddings $g : B \to D$ and $h : C \to D$ such that $ge = hg$.

Fraïssé's theorem [41, theorem 7.1.2] states that if K is a class of finitely generated (for graphs this is just finite) structures with the Hereditrary Property, the Joint Embedding Property and the Amalgamation Property, then there is an ultrahomogeneous, countable structure X into which every member of K embeds, furthermore the only finitely generated structures which embed into X are the members of K. So Fraïssé's theorem proves

[4] A finite partial isomorphism of a graph is a partial 1-1 map, whose domain and range are finite sets of nodes of the graph, taking edges and non-edges to edges and non-edges respectively.

that a countable, undirected, ultrahomogenous graph exists and every finite graph embeds into it, as required by this proposition.

Similarly, it is easy to check that the class of all finite tournaments and the class of all finite triangle-free graphs have the hereditary property, the joint embedding property and the amalgamation property. Fraïssé's theorem yields the required graphs $\mathbf{T}$ and $\mathbf{N}$. □

We can use these graphs to provide the universal representations we need.

LEMMA 14 *Gen-SAT(#11), Gen-SAT(#17) and Gen-SAT(#18) have universal representations.*

PROOF:

We define a universal representation $\mathcal{T}$ of algebra #11, based on the tournament $\mathbf{T}$ of proposition 13. The domain of $\mathcal{T}$ is the (countable) set of nodes of $\mathbf{T}$. We interpret the atoms of algebra #11 by letting

$$\begin{aligned}(1')^{\mathcal{T}} &= \{(x,x) : x \text{ is a vertex of } \mathbf{T}\}\\ a^{\mathcal{T}} &= \{(x,y) : x \neq y \text{ and } (x,y) \text{ is an edge of } \mathbf{T}\}\\ (a^{\smile})^{\mathcal{T}} &= \{(x,y) : x \neq y \text{ and } (y,x) \text{ is an edge of } \mathbf{T}\}\end{aligned}$$

Having defined the interpretation of the three atoms, it is easy to extend this to all eight elements of algebra #11.

$$\alpha^{\mathcal{T}} = \bigcup\{\beta^{\mathcal{T}} : \beta \leq \alpha,\ \beta \text{ is an atom}\}$$

for any element $\alpha \in$#11. The fact that none of $1'^{\mathcal{T}}, a^{\mathcal{T}}, (a^{\smile})^{\mathcal{T}}$ are empty is enough to show that this defines a 1-1 map from algebra #11 into $\mathfrak{Re}(\mathsf{nodes}(\mathbf{T}))$, and the fact that $\mathbf{T}$ is ultrahomogenous. Indeed the much weaker fact that for every partial isomorphism p of $\mathbf{T}$ of size two and every node $t \in \mathbf{T}$ there is a partial isomorphism p^+ extending p with $t \in \mathsf{dom}(p^+)$) is enough to show that $(x;y)^{\mathcal{T}} = x^{\mathcal{T}}|y^{\mathcal{T}}$ for any x, y belonging to algebra #11. To illustrate how this works, we show that $(a;a)^{\mathcal{T}} = a^{\mathcal{T}}|a^{\mathcal{T}}$. Well

$$\begin{aligned}(x,y) \in a^{\mathcal{T}}|a^{\mathcal{T}} &\Leftrightarrow \exists z\ ((x,z) \in a^{\mathcal{T}} \wedge (z,y) \in a^{\mathcal{T}})\\ &\Leftrightarrow \exists z\ ((x,z),(z,y) \text{ are both edges of } \mathbf{T})\\ &\Rightarrow x \neq y\\ &\Leftrightarrow (x,y) \in (0')^{\mathcal{T}} = (a;a)^{\mathcal{T}}\end{aligned}$$

The third (one-way) implication holds because $\mathbf{T}$ is a tournament, so you cannot have both of (x,z) and (z,x) being edges of $\mathbf{T}$. This implication may be reversed. Suppose $x \neq y \in \mathbf{T}$. Since $\mathbf{T}$ is a tournament, either (x,y) or (y,x) is an edge. Without loss, assume the former. Since all finite tournaments embed in $\mathbf{T}$, there are three nodes $x', y', z' \in \mathbf{T}$ and $(x',y'), (x',z'), (z',y')$ are edges of $\mathbf{T}$. By ultrahomogeneity (or the weaker property described above) the partial isomorphism $\{(x', x), (y', y)\}$ extends to a partial isomorphism $\{(x', x), (y', y), (z', z)\}$ for some node $z \in \mathbf{T}$. So (x, z) and (z, y) are edges of $\mathbf{T}$. This reverses the third implication. Since this implication can be reversed, we have $(x, y) \in a^{\mathcal{T}}|a^{\mathcal{T}} \Leftrightarrow (x, y) \in (a; a)^{\mathcal{T}}$, as required. Similarly, other products of elements are correctly represented. This is the critical point in showing that $\mathcal{T}$ is indeed a representation of the algebra.

For universality, let $\{C_{ij}x_ix_j : i, j < n\}$ be some closed, non-zero set of constraints over algebra #11. Suppose for contradiction that these constraints are not satisfiable in $\mathcal{T}$. Further, suppose that the number, n, of distinct variables occuring in the set of constraints is least possible for such a situation.

If there are $k \neq l < n$ and $C_{kl} = 1'$, then $\{C_{ij}x_ix_j : i, j < n,\ k \neq i,\ k \neq j\}$ is a non-zero, closed set of constraints not satisfiable in $\mathcal{T}$ with a smaller set of variables, contrary to assumption.

Hence we can assume, for all $i \neq j < n$, that either $a \leq C_{ij}$ or $a^{\smile} \leq C_{ij}$. Closure of the constraints implies that $C_{ji} = C_{ij}^{\smile}$, so $a \leq C_{ij} \Leftrightarrow a^{\smile} \leq C_{ji}$, for $i \neq j < n$. Define a tournament $\mathbf{C}$ with nodes $\{0, 1, \ldots, n-1\}$ by letting (i, j) be an edge of $\mathbf{C}$ (for each $i < j < n$) iff $a \leq C_{ij}$, and (j, i) is an edge of $\mathbf{C}$ iff (i, j) is not an edge of $\mathbf{C}$.

Every finite tournament embeds into $\mathbf{T}$, so let ι be an embedding of $\mathbf{C}$ into $\mathbf{T}$. This embedding determines a variable assignment v, given by $v(x_i) = \iota(i)$, for $i < n$. We have to check that v satisfies all the constraints. Let $C_{ij}x_ix_j$ be one of the constraints and suppose $i < j$. If $a \leq C_{ij}$ then (i, j) is an edge of $\mathbf{C}$ and hence $(\iota(i), \iota(j))$ is an edge of $\mathbf{T}$. Therefore, $(v(x_i), v(x_j)) = (\iota(i), \iota(j)) \in a^{\mathcal{T}} \subseteq C_{ij}^{\mathcal{T}}$. If $a \not\leq C_{ij}$ then $a^{\smile} \leq C_{ij}$, (j, i) is an edge of $\mathbf{T}$ and $(\iota(j), \iota(i))$ is an edge of $\mathbf{T}$. So $(\iota(i), \iota(j))$ is not an edge of $\mathbf{T}$, hence $(v(x_i), v(x_j)) = (\iota(i), \iota(j)) \in (a^{\smile})^{\mathcal{T}} \subseteq C_{ij}^{\mathcal{T}}$. If $j < i < n$ then $(v(x_j), v(x_i)) \in C_{ji}^{\mathcal{T}} \Rightarrow (v(x_i), v(x_j)) \in C_{ij}^{\mathcal{T}}$. Thus all constraints are satisfied by v.

This proves that $\mathcal{T}$ is universal.

In a very similar way we can construct universal representations $\mathcal{N}, \mathcal{R}$ of algebras #17 and #18 respectively. For the former we let

$$\begin{aligned}(1')^{\mathcal{N}} &= \{(x, x) : x \text{ is a vertex of } \mathbf{N}\}\\ a^{\mathcal{N}} &= \{(x, y) : x \neq y \text{ and } (x, y) \text{ is an edge of } \mathbf{N}\}\\ b^{\mathcal{N}} &= \{(x, y) : x \neq y \text{ and } (x, y) \text{ is not an edge of } \mathbf{N}\}\end{aligned}$$

and for the latter we let

$$\begin{aligned}(1')^{\mathcal{R}} &= \{(x, x) : x \text{ is a vertex of } \mathbf{R}\}\\ a^{\mathcal{R}} &= \{(x, y) : x \neq y \text{ and } (x, y) \text{ is an edge of } \mathbf{R}\}\\ b^{\mathcal{R}} &= \{(x, y) : x \neq y \text{ and } (x, y) \text{ is not an edge of } \mathbf{R}\}\end{aligned}$$

As with the first case, the fact that all atoms have non-empty interpretation plus the ultrahomogeneity of the graph is enough to prove that $\mathcal{N}$ and $\mathcal{R}$ are representations of algebras #17 and #18 respectively. And the fact that every finite triangle-free graph embeds in $\mathbf{N}$ (respectively, every finite graph embeds in $\mathbf{R}$) suffices to show that these are universal representations. $\Box$

LEMMA 15 *Algebra #13 has a universal representation.*

PROOF:

A universal representation $\mathcal{M}$ of algebra #13 can be constructed by taking as a domain the disjoint union of two countably infinite sets X and Y. To interpret the atoms,

$$\begin{aligned} 1'^{\mathcal{M}} &= \{(z,z) : z \in X \cup Y\} \\ a^{\mathcal{M}} &= \{(x,x') : x \neq x' \in X\} \cup \{(y,y') : y \neq y' \in Y\} \\ b^{\mathcal{M}} &= \{(x,y),(y,x) : x \in X, y \in Y\} \end{aligned}$$

so $b^{\mathcal{M}}$ is a bipartite graph over X, Y. To see that this is indeed a representation, just check that composition of the binary relations just defined corresponds exactly to the composition table given for algebra #13. For universality, let $\{C_{ij}(x_i,x_j) : i,j < n\}$ be a non-zero closed set of constraints. We claim that these constraints are satisfiable in $\mathcal{M}$.

For the claim, define constraints B_{ij} over algebra #4 by letting $B_{ij} \geq 1'_{\text{alg.}\#4}$ iff $C_{ij} \bullet (-b) \neq 0$, and $B_{ij} \geq 0'_{\text{alg.}\#4}$ iff $C_{ij} \geq b$, for all $i,j < n$. Since $(-b);(-b) = b;b = -b$ and $(-b);b = b;(-b) = b$, it follows that $\{B_{ij}x_ix_j : i,j < n\}$ is a closed, non-zero set of constraints over algebra #4. By lemma 8 these constraints are satisfiable in the universal, two element representation $\mathcal{M}$ of algebra #4. Let the two elements of $\mathcal{M}$ be $\{0,1\}$, say, and let h be a variable assignment into $\mathcal{M}$ satisfying the constraints $\{B_{ij}x_ix_j : i,j < n\}$. Let $I_0 = \{i < n : h(i) = 0\}$ and $I_1 = \{i < n : h(i) = 1\}$. Define new constraints $\{D_{ij}(x_i,x_j) : i,j \in I_0\}$ over algebra #5 by $D_{ij} = C_{ij} \bullet (-b)$. This is a closed set of non-zero constraints and so, by lemma 11, there is a variable assignment f_0 from the variables $\{x_i : i \in I_0\}$ into a model, whose domain we may as well take to be X, satisfying all these constraints. Similarly, we can define constraints $\{E_{ij}(x_i,x_j) : i,j \in I_1\}$ over algebra #5 in exactly the same way, and thus find a variable assignment f_1 from $\{x_i : i \in I_1\}$ into a representation whose domain is Y. The variable assignment $f = f_0 \cup f_1$ is defined on all variables $\{x_i : i < n\}$ and satisfies all the constraints $\{C_{ij} : i,j < n\}$.

$\Box$

LEMMA 16 *Algebra #14 has a universal representation.*

PROOF:

Let the domain $M = \{n, n' : n \in \mathbb{N}\}$, i.e. take two disjoint copies of the natural numbers. To interpret the atoms of algebra #14 let

$$1'^{\mathcal{M}} = \{(n,n),(n',n') : n \in \mathbb{N}\}$$
$$a^{\mathcal{M}} = \{(n,n'),(n',n) : n \in \mathbb{N}\}$$
$$b^{\mathcal{M}} = 1^{\mathcal{M}} \setminus (1'^{\mathcal{M}} \cup a^{\mathcal{M}})$$

As with the previous lemma, you can check that this defines a representation by comparing the composition of the binary relations just defined with those in the composition table for algebra #14. We must check that it is universal. So let $\Xi = \{C_{ij} : i,j < n\}$ be any non-zero, closed set of constraints. The element $-b = 1 + a$ is an *equivalence element*, i.e. $-b \geq 1', (-b)^{\smile} = (-b);(-b) = -b$. Let $\sim$ be the binary relation over the set of variables occuring in Ξ defined by $x_i \sim x_j$ iff $C_{ij} \leq -b$, for $i,j < n$. By closure of the constraints, $\sim$ is an equivalence relation over the set of variables. The restriction of algebra #14 to the elements below $-b$ is a relation algebra isomorphic to algebra #4. So, for each $\sim$-equivalence class α, the restriction Ξ_α of Ξ to the constraints using only variables in α is a non-zero, closed set of constraints over algebra #4. By lemma 8 there is an assignment h_α to the variables in Ξ_α into a representation of algebra #4. Since the only square representation of this algebra has size two, we can take the domain of the representation to be $\{n, n'\}$, for some $n = n(\alpha) \in \mathbb{N}$, and we can assume that $n(\alpha) = n(\beta) \Rightarrow \alpha = \beta$, for any equivalence classes α and β. Now for any variables x_i, x_j where $i \not\sim j$ we have $b \leq C_{ij}$ (else $i \sim j$). So $h =_{def} \bigcup_{\text{equiv. classes } \alpha} h_\alpha$ is a variable assignment satisfying all of Ξ.

$\square$

LEMMA 17 *Algebra #15 has a universal representation.*

PROOF:

For the domain take $\mathbb{N} \times \mathbb{N}$. Interpret that atoms by

$$1'^{\mathcal{M}} = \{(p,p) : p \in \mathbb{N} \times \mathbb{N}\}$$
$$a^{\mathcal{M}} = \{((m,n),(m',n)) : m,m',n \in \mathbb{N},\ m \neq m'\}$$
$$b^{\mathcal{M}} = \{((m,n),(m',n')) : m,m',n,n' \in \mathbb{N},\ n \neq n'\}$$

Let $\Xi = \{C_{ij}x_ix_j : i,j < n\}$ be a non-zero closed set of constraints over algebra #15. The element $1' + a = -b$ is an equivalence element. Define an equivalence relation $\sim$ over the variables in Ξ by $x_i \sim x_j$ iff $C_{ij} \leq -b$. This is indeed an equivalence relation, by the closure of the constraints. As in the previous lemma, if we restrict algebra #15 to the elements below $-b$ we get a relation algebra isomorphic, this time, to algebra #5. So, if we restrict Ξ to those constraints using only variables occuring in a given $\sim$-equivalence class α, we get a non-zero, closed set of constraints Ξ_α over algebra #5. By lemma 11 there is a variable assignment h_α satisfying Ξ_α. We can take the domain of the representation of algebra #5 to be $\{(m,n) : m \in \mathbb{N}\}$, for some integer $n = n(\alpha)$, since this is an infinite set, and we can assume that n is unique to α (i.e. $n(\alpha) = n(\beta) \Rightarrow \alpha = \beta$ for any equivalence classes α, β). Then $h =_{def} \bigcup_\alpha h_\alpha$ is a variable assignment satisfying all of Ξ.

□

No.	Gen-SAT	$\mathcal{M} - SAT$
#1	$O(1)$	$O(1)$
#2	$O(n)$	$O(n)$
#3	$O(n)$	$O(n)$
#4	$O(n^3)$	$O(n^3)$
#5	$O(n^3)$	$\begin{cases} O(n^3) & \mathcal{M} \text{ infinite, square} \\ \textbf{NPC} & \mathcal{M} \text{ finite} \end{cases}$
#6	$O(n)$	$O(n)$
#7	$O(n^3)$	$O(n^3)$
#8	$O(n^3)$	$\begin{cases} O(n^3) & \mathcal{M} \text{ infinite, square} \\ \textbf{NPC} & \mathcal{M} \text{ finite} \end{cases}$
#9	**NPC**	**NPC**
#10	$O(n^3)$	$O(n^3)$
#11	$O(n^3)$	**NPC** if $\mathcal{M}$ is finite
#12	**NPC**	**NPC**
#13	$O(n^3)$	**NPC** if $\mathcal{M}$ is finite
#14	$O(n^3)$	**NPC** if $\mathcal{M}$ is finite
#15	$O(n^3)$	**NPC** if $\mathcal{M}$ is finite
#16	**NPC**	**NPC**
#17	$O(n^3)$	**NPC** if $\mathcal{M}$ is finite
#18	$O(n^3)$	**NPC** if $\mathcal{M}$ is finite

Fig. 3. Summary of results. The eighteen isomorphisms types with complexity results. Only representations obtained by taking the disjoint union of one square representation for each simple component are considered here.

Pulling all this together:

THEOREM 18 *The complexities of Gen-SAT(#1)–Gen-SAT(#18) are as* *given in figure 3.*

PROOF:

The first four complexities are given in lemma 8. The complexity of Gen-$SAT(\#5)$ is given in lemma 11. Algebra #6 is the direct product $\#2 \times \#2 \times \#2$, so Gen-$SAT(\#6)$ has the same complexity as Gen-$SAT(\#2)$, by lemma 7, namely $O(n)$. Similarly, algebra #7 is isomorphic to $\#2 \times \#4$ so by lemma 7 its complexity is $O(n^3)$. Algebra #8 is isomorphic to $\#2 \times \#5$ so its complexity is $O(n^3)$. Gen-$SAT(\#9)$, Gen-$SAT(\#12)$ and Gen-$SAT(\#16)$ are **NP**-complete, by corollary 10. Gen-$SAT(\#10)$ has cubic complexity, by theorem 12. Gen-$SAT(\#11)$, Gen-$SAT(\#17)$ and Gen-$SAT(\#18)$ have cubic complexity, by lemma 14 and lemma 6. The complexities of Gen-$SAT(\#13)$, Gen-$SAT(\#14)$ and Gen-$SAT(\#15)$ are cubic, by lemma 6 and lemmas 15, 16 and 17, respectively. □

5 Conclusions and further work

We have analyzed the computational complexity of the Gen-SAT problem on the eighteen small relation algebras classified by Maddux and Andrèka in [30]. This analysis provides a complete computational account for the small relation algebras.

Some problems about computational complexity of the constraint satisfaction problem for small relation algebras over specified representations remain open. We still need to establish whether the $\mathcal{M} - SAT$ problems are tractable for infinite models (other than those used for the above analysis).

Another important problem arising from the applications is to restrict Gen-$SAT(\mathcal{A})$ to a specified subset S of the relation algebra $\mathcal{A}$. That is, we want to know if a set of constraints $\{\sigma_{ij}x_ix_j : i, j < n\}$, where $\sigma_{ij} \in S$ for $i, j < n$, is satisfiable in a representation of $\mathcal{A}$. It can happen that the complexity of this restricted problem is lower than that of Gen-$SAT(\mathcal{A})$. The analysis of subsets has been studied for many algebras used, in particular, for knowledge representation as in the case of Allen's algebra [43–46], for the Region Connection Calculus [47–49], for the congruence algebra [50], but a general solution is still a long way off.

Finally, an observation about our results, leading to two further problems. For any small relation algebra $\mathcal{A}$ we have seen that the complexity of Gen-$SAT(\mathcal{A})$ is either cubic (because $\mathcal{A}$ has a universal representation) or **NP**-complete. Two problems arise.

PROBLEM 1 *Find a relation algebra $\mathcal{A}$ with no universal representation but where the complexity of Gen-$SAT(\mathcal{A})$ is polynomial.*

PROBLEM 2 *Find a relation algebra $\mathcal{A}$ such that the complexity of Gen-$SAT(\mathcal{A})$ is polynomial, but worse than cubic.*

References

[1] R. Maddux, The origin of relation algebras in the development and axiomatization of the calculus of relations, Studia Logica 3/4 (1991) 421–455.

[2] I. Anellis, N. Houser, Nineteenth century roots of algebraic logic and universal algebra, in: Andréka et al. [51], pp. 1–36.

[3] J. Allen, J. Koomen, Planning using a temporal world model, in: Proc. 8th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1983, pp. 711–714.

[4] P. Ladkin, A. Reinefeld, A symbolic approach to interval constraint problems, Artificial Intelligence and Symbolic Mathematical Computing 737 (1993) 65–84.

[5] J. Allen, An interval-based representation of temporal knowledge, in: P. J. Hayes (Ed.), Proc. 7th International Joint Conference on Artificial Intelligence, William Kaufmann, 1981, pp. 221–226.

[6] J. Allen, Maintaining knowledge about temporal intervals, Communications of the ACM 26 (11) (1983) 832–843.

[7] J. Allen, Towards a general theory of action and time, Artificial Intelligence 23 (2) (1984) 123–154.

[8] J. Allen, P. Hayes, A commonsense theory of time, in: A. Joshi (Ed.), Proc. 9th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1985, pp. 528–531.

[9] J. Allen, H. Kautz, A model of naïve temporal reasoning, in: J. Hobbs, R. Moore (Eds.), Contributions to Artificial Intelligence, Volume 1, Ablex, Norwood, N.J., 1983.

[10] R. Dechter, I. Meiri, J. Pearl, Temporal constraint networks, Artificial Intelligence 49 (1991) 61–95.

[11] P. Ladkin, R. Maddux, On binary constraint problems, Journal of the Association of Computing Machinery 41 (1994) 435–469.

[12] T. Dean, D. McDermott, Temporal database management, Artificial Intelligence 32 (1987) 1–55.

[13] F. Anger, D. Mitra, R. Rodriguez, Temporal Constraint Networks in Nonlinear Time, in: Workshop Notes of the Spatial and Temporal Reasoning at ECAI98, Brighton, UK, 1998.

[14] F. Anger, D. Mitra, R. Rodriguez, Satisfiability in Nonlinear Time: Algorithms and Complexity, in: Proceedings of the Florida Artificial Intelligence Research Society conference, Orlando (USA), 1999.

[15] B. Bennett, Determining consistency of topological relations, Constraints 3 (2&3) (1998) 213–225.

[16] M. Egenhofer, Reasoning about binary topological relations, in: O. Gunther, H. J. Schek (Eds.), Advances in Spatial Databases-Second Symposium SSD '91, no. 525 in Lecture Notes in Computer Science, Springer-Verlag, New York, NY, USA, 1991, pp. 143–160.

[17] A. Frank, Qualitative Spatial Reasoning about Distances and Directions, Journal of Visual Languages and Computing 3 (4) (1992) 343–371.

[18] A. Mackworth, C. Freuder, The complexity of some polynomial network consistency algorithms for constraint satisfation problems, Artificial Intelligence 25 (1) (1985) 65–74.

[19] M. Vilain, H. Kautz, Constraint propagation algorithms for temporal reasoning, in: Proceedings of the fifth AAAI, 1986, pp. 377–382.

[20] R. Valdéz-Peréz, The satisfiability of temporal constraint networks, in: Proc. AAAI, 1987.

[21] M. Vilain, H. Kautz, P. van Beek, Constraint propagation algorithms for temporal reasoning — a revised report, in: D. S. Weld, J. de Kleer (Eds.), Readings in Qualitative Reasoning about Physical Systems, Morgan Kaufmann, 1989, pp. 373–381.

[22] E. Tsang, Foundations of Constraint Satisfaction, Academic Press, 1993.

[23] J. Lever, B. Richards, parcplan: a planning architecture with parallel actions, resources and constraints, in: Proceedings of ISMIS, 1994.

[24] R. Hirsch, Tractable approximations for temporal constraint handling, Artificial Intelligence 116 (2000) 287–295.

[25] I. Düntsch, H. Wang, S. McCloskey, A relation algebraic approach to the region connection calculus, Theoretical Computer Science 255 (2001) 63–83.

[26] I. Düntsch, H. Wang, E. Orlowska, Algebras of approximating regions, Fundamenta Informaticae 46 (2001) 71–82.

[27] I. Düntsch, G. Schmidt, M. Winter, A necessary relation algebra for mereotopology, Studia Logica 69 (2001) 381–409.

[28] M. Cristani, Many-sorted preference relations, in: F. Giunchiglia, D. Fensel, D. McGuiness, M. Williams (Eds.), KR'02: Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, 2002.

[29] I. Düntsch, Small integral relation algebras generated by a partial order, Periodica Mathematica Hingarica 23 (1991) 129–138.

[30] H. Andréka, R. Maddux, Representations for small relation algebras, Notre Dame J. Formal Logic 35 (4) (1994) 550–562.

[31] R. Hirsch, Expressive power and complexity in algebraic logic, Journal of Logic and Computation 7 (3) (1997) 309–351.

[32] R. Hirsch, A relation algebra with undecidable network satisfaction problem, Bulletin of the Interest Group in Pure and Applied Logics 7 (4) (1999) 547–554.

[33] A. Tarski, On the calculus of relations, J. Symbolic Logic 6 (1941) 73–89.

[34] B. Jónsson, A. Tarski, Boolean algebras with operators II, American Journal of Mathematics 74 (1952) 127–162.

[35] R. Hirsch, I. Hodkinson, Relation algebras by games, North-Holland, Elsevier Science, Amsterdam, 2002.

[36] R. Maddux, Introductory course on relation algebras, finite-dimensional cylindric algebras, and their interconnections, in: Andréka et al. [51], pp. 361–392.

[37] R. Maddux, Relation algebras, in preparation.

[38] R. Karp, Reducibility among combinatorial problems, in: R. Miller, J. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.

[39] M. Garey, D. Johnson, Computers and Intractability — a guide to the theory of **NP**-completeness, Freeman and Co., 1979.

[40] P. Ladkin, R. Maddux, On Binary Constraint Problems, Journal of the ACM 41 (3) (1994) 435–469.

[41] W. Hodges, Model theory, Encylopedia of mathematics and its applications, Cambridge University Press, 1995, second reprint.

[42] P. Cameron, Oligomorphic permutation groups, Vol. 152 of London Math. Soc. Lecture Note Series, Cambridge University Press, 1990.

[43] B. Nebel, H. Bürckert, Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra, Journal of the ACM 42 (1) (1995) 43–66.

[44] T. Drakengren, P. Jonsson, Maximal tractable subclasses of Allen's interval algebra: Preliminary report, in: Proceedings of the 13th (US) National Conference on Artificial Intelligence (AAAI-96), Portland, OR, USA, 1996, pp. 389–394.

[45] T. Drakengren, P. Jonsson, Twenty-one large tractable subclasses of Allen's algebra, Artificial Intelligence 93 (1-2) (1997) 297–319.

[46] P. Jonsson, T. Drakengren, C. Bäckström, Computational complexity of relating time points with intervals, Artificial Intelligence 109 (1-2) (1999) 273–295.

[47] P. Jonsson, T. Drakengren, A Complete Classification of Tractability in the Spatial Theory RCC-5, Journal of Artificial Intelligence Research 6 (1997) 211–221, research Note.

[48] J. Renz, B. Nebel, On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the Region Connection Calculus, Artificial Intelligence 108 (1-2) (1999) 69–123.

[49] J. Renz, Maximal Tractable Fragments of the Region Connection Calculus: A Complete Analysis, in: Proccedings of the 17th International Conference on Artificial Intelligence (IJCAI 99), 1999.

[50] M. Cristani, The Complexity of Reasoning about Spatial Congruence, Journal of Artificial Intelligence Research 11 (1999) 361–390.

[51] H. Andréka, J. Monk, I. Németi (Eds.), Algebraic logic, Vol. 54 of Colloq. Math. Soc. J. Bolyai, North-Holland, Amsterdam, 1991.