On Binary Constraint Problems

Peter B. Ladkin*
Institut für Informatik
Universität Bern
Längassstrasse 51
CH-3012 Bern
Switzerland

Roger D. Maddux†
Department of Mathematics
400 Carver Hall
Iowa State University
Ames, IA 50011
USA

March 1994

Appeared in: Journal of the ACM, May 1994.

Abstract

The concepts of binary constraint satisfaction problems can be naturally generalized to the relation algebras of Tarski. The concept of path-consistency plays a central role. Algorithms for path-consistency can be implemented on matrices of relations and on matrices of elements from a relation algebra. We give an example of a 4-by-4 matrix of infinite relations on which no iterative local path-consistency algorithm terminates. We give a class of examples over a fixed finite algebra on which all iterative local algorithms, whether parallel or sequential, must take quadratic time. Specific relation algebras arising from interval constraint problems are also studied: the Interval Algebra, the Point Algebra, and the Containment Algebra.

1 Introduction

The logical study of binary relations is classical [8], [9], [51], [52], [56], [53], [54]. Following this tradition, Tarski formulated the theory of binary relations as an algebraic theory called *relation algebra* [59][1].

Constraint satisfaction problems (CSP's) are a class of logical problems concerning relations. Traditional constraint problems have been concerned with finite relations [7], [12], [14], [35], [36], [37], [49], [67].

*Current address: Dept. of Computing Science, University of Stirling, Stirling FK9 4LA, Scotland.

†We wish to thank Giuseppe Scollo of the University of Twente for his extensive commentary on the precursor of this paper, and two anonymous referees for their extensive and helpful critique. The first author wishes to thank Kestrel Institute for initial support of work on binary constraint problems.

[1]This should not be confused with the algebraic formulation of operations on relational data bases, usually called *relational algebra*, which bears a closer relationship to cylindric algebra (also formulated by Tarski), as shown in [16].

There has been considerable interest in solving constraint problems involving intervals, stemming from the work of Allen [1] in temporal reasoning. In these problems, the underlying domain of values is infinite, and each constraint is an infinite relation. Allen adapted a constraint propagation algorithm to an infinite domain to help in the analysis of such problems. See [2], [6], [13], [20], [22], [24], [25], [26], [27], [28], [29], [30], [31], [32], [45], [46], [50], [55], [62], [63], [64], and [66].

We formulate CSP concepts and methods using relation algebras. We believe this clarifies the mathematics of binary constraint satisfaction methods, and allows problems with finite or potentially infinite domains to be handled in a uniform way. For example, many results may be proved for finite relation algebras (*i.e.*, those having only finitely many relations) with no cardinality restriction on the size of the domains of the constraint relations. It also allows the study of problems in which the smallest class of relations closed under the usual constraint-manipulation operations is infinite, *e.g.*, the class of union-of-convex interval relations in [21], which cannot easily be handled with traditional methods. One important class of algebras is the class of atomic relation algebras. All finite relation algebras are also atomic, but some atomic algebras are infinite, *i.e.*, they contain infinitely many relations. The assumption of finiteness or atomicity frequently appears in results below. Some of our results have been used in code for binary constraint problem manipulation [55].

Synopsis In Section 2, we introduce binary constraint problems, operations on binary relations, constraint matrices, and path-consistency. Section 3 introduces relation algebras, and formulates the necessary relation algebraic definitions. Sections 2 and 3 contain most of the background material.

Section 4 presents four particular relation algebras. Two of these, the Interval Algebra ***IA*** (defined implicitly in [1]) and the Point Algebra ***PA***, are well-known in the interval constraint literature. We prove, in a precise sense, that both the ***PA*** and ***IA*** have canonical representations on a countable set, to which all other representations are isomorphic. The Containment Algebra ***CA***, a subalgebra of the ***IA***, is introduced, as is the Hard Algebra ***HA***. Both these algebras are used later in examples.

In Section 5, we formulate constraint matrices over relation algebras (RA matrices) and investigate their properties. RA matrices generalize constraint problems to relation algebras, and are equivalent to them when the relation algebras are algebras of binary relations (proper relation algebras). We define a closed matrix to be one that is symmetric and satisfies the path-consistency condition. We show that closure computations must terminate for RA matrices of order 3. The notion of proto-solvability is formulated, and its relation to actual solutions is clarified. This concept, under various names, has played a significant role in the literature on interval constraint problems, as has the concept of minimal labeling. We define the minimal labeling of a matrix M, and characterize it as the smallest closed matrix equivalent to M. We give an example of a ***CA***-matrix which has a solution in one representation but no solution in another. We show that closed ***PA***-matrices with no zero labels have solutions in the canonical representation, and use this to make some asymptotic complexity observations.

In Section 6, we formulate a scheme defining iterative local algorithms for closure. We show that every iterative local algorithm computes the most general closed reduction of an RA matrix, and that all iterative local algorithms terminate on any RA matrix over a finite relation algebra, and also on all RA matrices of order 3. We give an example of a constraint matrix of order 4 on which no iterative local algorithm terminates. We also comment on a specific iterative local algorithm `CMat` for constraint matrices, and its use in code for solving constraint problems.

Finally, Section 7 presents a class of examples over the Hard Algebra, which is representable on an 84-element set, on which any serial or parallel iterative local algorithm must take quadratic time. This result is close to the crude parallel upper bound estimate for `CMat` of $O(n^2 \log n)$, and so the parallel asymptotic complexity of iterative local algorithms is approximately characterized.

2 Binary Constraint Problems

A *constraint problem* of order n consists of a finite set of n variables $x_0, \ldots, x_{n-1}$, a set U (called the *universe* of the problem), together with a collection of constraints on the values from U which may be assigned to the variables [49]. The problem is *solvable* if the constraints can be satisfied by some assignment of values $a_0, \ldots, a_{n-1} \in U$ to the variables $x_0, \ldots, x_{n-1}$, in which case the sequence $\langle a_0, \ldots, a_{n-1} \rangle$ is called a *solution*. Two problems are *equivalent* if they have the same set of solutions. Constraint problems may also be formulated using the vocabulary of first-order logic [35], [36].

If every one of the constraints in a problem involves two variables (possibly the same) and asserts that the pair of values assigned to those variables must lie in a certain binary relation, then the constraint problem is called *binary*. We will be concerned exclusively with binary constraint problems.

Unary constraints as binary constraints Unary constraints can easily be expressed as binary constraints. For example, the assertion that $a_0 \in X$ is equivalent to the assertion that $\langle a_0, a_0 \rangle \in X \times X$.

2.1 Constraint networks

Every binary constraint problem can be represented as a labeled digraph, in which each node is labeled by a variable, each ordered pair of nodes is connected by only one directed edge, and each directed edge is labeled with the relation which constrains the two variables at the tail and head. Such a labeled digraph is called a *constraint network*. If all the constraint relations are symmetric, then we can represent the associated network as an undirected graph, as in Theorem 5.9.

2.2 Operations on relations

As indicated above, a binary relation is just a set of ordered pairs, denoted $\langle a, b\rangle$. For any two binary relations R and S, $R \cap S$ is the *intersection* of R and S, $R \cup S$ is the *union* of R and S, $R;S$ is the *relative product* (or *compositum*) of R and S, and $\breve{R}$ is the *converse* of R. These operations are defined as usual:

$$
\begin{aligned}
R \cap S &= \{\langle a, b\rangle : \langle a, b\rangle \in R \text{ and } \langle a, b\rangle \in S\},\\
R \cup S &= \{\langle a, b\rangle : \langle a, b\rangle \in R \text{ or } \langle a, b\rangle \in S\},\\
R;S &= \{\langle a, b\rangle : \text{for some } c,\ \langle a, c\rangle \in R \text{ and } \langle c, b\rangle \in S\},\\
\breve{R} &= \{\langle a, b\rangle : \langle b, a\rangle \in R\}.
\end{aligned}
$$

Composition (;) and *conversion* ($\breve{\ }$) of binary relations were introduced by De Morgan [8], [9, p55f, p208ff]. Composition was called *relative multiplication* by Peirce [52] and Schröder [56].

In expressions written without full parentheses, unary operators should be computed first, followed by relative multiplication, intersection, and union, in that order. Since we shall use at least one non-associative operation (matrix relative multiplication, defined in 2.4 below, and not to be confused with relative multiplication of binary relations, which is associative), repeated binary operations at the same computation priority level, written without parentheses, should be computed from left to right. For example, $R \cup S \cap T;\breve{V} = R \cup (S \cap (T;(\breve{V})))$ and $R;S;T = (R;S);T$.

Three special binary relations are the *empty relation* $\emptyset$ which contains no pairs at all, the *identity relation* $Id = \{\langle a, b\rangle : a = b\}$ and the *diversity relation* $Di = \{\langle a, b\rangle : a \neq b\}$.

The *domain* of a relation R is $\{a : \text{for some } b,\ \langle a, b\rangle \in R\}$, and the *range* of R is $\{b : \text{for some } a,\ \langle a, b\rangle \in R\}$. The *field* of a relation is the union of its domain and range.

2.3 Constraint matrices

Let P be a binary constraint problem with variables $x_0, \ldots, x_{n-1}$ and universe U. Choose an ordered pair of variables (which may coincide), say $\langle x_i, x_j\rangle$. Suppose that P has several constraints on $\langle x_i, x_j\rangle$, that is, the pair $\langle a_i, a_j\rangle$ is required to belong to several binary relations, say $R_1, \ldots, R_m$. Then these constraints can be condensed to a single constraint asserting that $\langle a_i, a_j\rangle$ must belong to the relation $R_1 \cap \cdots \cap R_m$. The resulting problem is clearly equivalent to P. Another possibility is that there is no constraint on $\langle x_i, x_j\rangle$. It is easy to add a superfluous constraint on $\langle x_i, x_j\rangle$ which results in an equivalent problem: just insist that $\langle a_i, a_j\rangle$ must belong to $U \times U$, and in case $x_i = x_j$, use $(U \times U) \cap Id$ instead of $U \times U$ to reflect the fact that only one value can be assigned to a single variable. If there is a constraint on $\langle x_i, x_i\rangle$, we may intersect that constraint with Id and still have an equivalent problem. With slight terminological abuse, we say that a relation is an *identity relation* if it is included in Id. We therefore restrict our attention to those binary constraint problems which have *exactly one* constraint on each ordered pair of variables, and if the two variables are the same, then the constraint on them is an identity relation. The main reason for doing so is that such a binary constraint problem with n variables and universe U can be simply viewed as an n-by-n matrix M of binary relations over U: the relation M_{ij} (in row i, column j) is the constraint on $\langle x_i, x_j \rangle$. Note that if M is an n-by-n matrix of relations which arises in this way from a binary constraint problem, then the relations on the main diagonal of M are all identity relations, *i.e.*, $M_{ii} \subseteq Id$ for all $i < n$. Conversely, any such matrix determines some binary constraint problem with universe $U(M) = \bigcup_{i,j<n}(\text{field of } M_{ij})$. Note that $U(M)$ may be a proper subset of the universe of the problem from which the matrix M arose. We refer to matrices with identity relations on the main diagonal as *constraint matrices*.

The solutions of a constraint matrix M are those n-tuples $\langle a_0, \ldots, a_{n-1} \rangle \in U(M)^n$ such that $\langle a_i, a_j \rangle \in M_{ij}$ whenever $i, j < n$. Note that if there are $i, j < n$ such that $M_{ij} = \emptyset$, then M has no solutions. This condition is sufficient but not necessary for M to have no solutions.

The constraint network corresponding to a constraint matrix M is a "labeled complete digraph", a digraph with exactly one directed edge from every node to each (possibly the same) node, whose labels are the entries of the matrix. We will therefore sometimes refer to M_{ij} as the *label* on the *edge* $\langle i, j \rangle$, to the triple $\langle i, j, k \rangle$ as a *triangle* in the matrix, and to the matrix itself as a network.

2.4 Operations on matrices of relations

Let M and N be n-by-n matrices of binary relations. Then $\breve{M}$ is the *converse* of M, $M \cdot N$ is the *intersection* of M and N, and $M;N$ is the *relative product* of M and N. These matrices are defined by the following equations, which hold for all $i, j < n$:

$$
\begin{aligned}
(\breve{M})_{ij} &= (M_{ji})\breve{}, \\
(M \cdot N)_{ij} &= M_{ij} \cap N_{ij}, \\
(M;N)_{ij} &= M_{i0};N_{0j} \cap M_{i1};N_{1j} \cap \cdots \cap M_{i,n-1};N_{n-1,j} = \bigcap_{k<n} M_{ik};M_{kj}.
\end{aligned}
$$

It is interesting to note, and easy to show, that relative multiplication is associative when applied to binary relations or to 1-by-1 matrices of relations, but it is not associative when applied to n-by-n matrices of relations for $n > 1$, even if they have identity relations on the diagonal and are therefore constraint matrices. For example, if

$$
M = \begin{pmatrix} \{\langle 0,0\rangle, \langle 1,1\rangle, \langle 2,2\rangle\} & \{\langle 0,1\rangle\} \\ \emptyset & \{\langle 0,0\rangle, \langle 1,1\rangle, \langle 2,2\rangle\} \end{pmatrix}
$$

and

$$
N = \begin{pmatrix} \{\langle 0,0\rangle, \langle 1,1\rangle, \langle 2,2\rangle\} & \{\langle 0,2\rangle\} \\ \{\langle 1,0\rangle, \langle 2,0\rangle\} & \{\langle 0,0\rangle, \langle 1,1\rangle, \langle 2,2\rangle\} \end{pmatrix},
$$

then

$$
\begin{pmatrix} \{\langle 0,0\rangle\} & \emptyset \\ \emptyset & \emptyset \end{pmatrix} = (M;N);N \neq M;(N;N) = \begin{pmatrix} \emptyset & \emptyset \\ \emptyset & \emptyset \end{pmatrix}.
$$

(Incidentally, M is a constraint matrix with no solutions because one of its labels is $\emptyset$, but the constraint matrix N has exactly one solution, namely $\langle 0,2\rangle$.)

If $N \cdot M = N$, *i.e.*, $N_{ij} \subseteq M_{ij}$ for all $i,j < n$, then we say that N is a *reduction* of M, and symbolize this by writing $N \leq M$ or $M \geq N$. We write $N < M$ or $M > N$ in case $N \leq M$ and $N \neq M$. It is easy to see that this defines a partial ordering on matrices of relations. The greatest lower bound of the two matrices M and N in this ordering is $M \cdot N$.

Let $M^2 = M;M$. We shall use iterated squaring, but since relative multiplication of constraint matrices is not associative, we do not define general powers M^k.

2.5 Path-consistency and symmetry

An n-by-n constraint matrix M is *path-consistent* if $M \leq M^2$. By the definitions above, M is path-consistent just in case $M_{ij} \subseteq M_{ik};M_{kj}$ for all $i,j,k < n$. An easy proof by induction [49] shows that M is path-consistent if and only if for every finite set of indices $i_1, i_2, \ldots, i_m < n$, and all $j,k < n$, we have

$$M_{jk} \subseteq M_{ji_1};M_{i_1i_2};M_{i_2i_3};\cdots;M_{i_{m-1}i_m};M_{i_mk}.$$

This indicates the origin of the term "path-consistent".

For any solution $\langle a_0,\ldots,a_{n-1}\rangle$ we must have $\langle a_i,a_j\rangle \in M_{ik};M_{kj}$ for any $i,j,k < n$. This necessary condition may be used as a pruning technique to narrow down the potential choices of a_i and a_j in a solution search: we calculate the relation $M_{ik};M_{kj}$ and intersect it with M_{ij} to form $M_{ij} \cap M_{ik};M_{kj}$, the new, potentially smaller, relation (and thus tighter constraint). We use this as the new constraint on $\langle x_i,x_j\rangle$, and repeat these steps for all edges in the triangle $\langle i,j,k\rangle$. We call this a *triangle operation.* We say that a triangle is *stable* if none of the labels between i,j,k changes during a triangle operation. Clearly a network is path-consistent if every triangle is stable.

A constraint matrix M is *symmetric* if $M = \breve{M}$. A constraint matrix can be path-consistent without being symmetric. For example, if

$$M = \begin{pmatrix} \{\langle 0,0\rangle, \langle 2,2\rangle\} & \{\langle 0,1\rangle, \langle 2,3\rangle, \langle 0,3\rangle\} \\ \{\langle 1,0\rangle, \langle 3,2\rangle, \langle 1,2\rangle\} & \{\langle 1,1\rangle, \langle 3,3\rangle\} \end{pmatrix}$$

then $M = M^2$ but $M \neq \breve{M}$. (The constraint matrix M has exactly two solutions, namely $\langle 0,1\rangle$ and $\langle 2,3\rangle$.)

A constraint matrix is *closed* if it is both path-consistent and symmetric.

3 Relation algebras in general

We state here concepts and results concerning relation algebras which are needed later in the paper. For proofs and more information on relation algebras see [5], [17], [18], [19], [38], [41], [42], [44], and [61].

3.1 Definition of Boolean algebras

Before defining relation algebras, we recall that a *Boolean algebra* is an algebra of the form

$$\langle B, +, \cdot, ^{-}, 0, 1\rangle$$

which satisfies the identities

$$\begin{array}{ll} x + (y + z) = (x + y) + z, & x \cdot (y \cdot z) = (x \cdot y) \cdot z, \\ x + y = y + x, & x \cdot y = y \cdot x, \\ x \cdot y + x = x, & (x + y) \cdot x = x, \\ x \cdot y + z = (x + z) \cdot (y + z), & (x + y) \cdot z = x \cdot z + y \cdot z, \\ x + \overline{x} = 1, & x \cdot \overline{x} = 0. \end{array}$$

Every Boolean algebra also satisfies the identities

$$\begin{array}{l} x + x = x, \\ x \cdot x = x, \\ \overline{\overline{x}} = x, \\ \overline{x + y} = \overline{x} \cdot \overline{y}, \\ \overline{x \cdot y} = \overline{x} + \overline{y}. \end{array}$$

The partial ordering of a Boolean algebra is defined thus: $x \leq y$ if $x \cdot y = x$.

3.2 Definition of relation algebras

$\mathfrak{A}$ is a *relation algebra* (with *universe* A) if

$$\mathfrak{A} = \langle A, +, \cdot, ^{-}, 0, 1, ;, ^{\smile}, 1'\rangle,$$

where $\langle A, +, \cdot, ^{-}, 0, 1\rangle$ is a Boolean algebra (called the *Boolean part* (or *reduct*) of $\mathfrak{A}$), ; is a binary operation, $^{\smile}$ is a unary operation, and $1' \in A$, and the following identities hold:

$$\begin{array}{ll} (1) & (x;y);z = x;(y;z), \\ (2) & (x + y);z = x;z + y;z, \\ (3) & x;1' = x = 1';x, \\ (4) & \breve{\breve{x}} = x, \\ (5) & (x + y)^{\smile} = \breve{x} + \breve{y}, \\ (6) & (x;y)^{\smile} = \breve{y};\breve{x}, \\ (7) & \breve{x};\overline{x;y} \cdot y = 0. \end{array}$$

(The second equality could be deleted, since it is derivable from the others.) The first six of these identities can be expressed by saying that $\langle A, ;, ^{\smile}, 1'\rangle$ is an involuted monoid whose operations distribute over the Boolean + (from both sides, by (2), (4), and (6)), and the final identity ties this monoid to the Boolean part of $\mathfrak{A}$. We refer to 1 as the *Boolean unit* of $\mathfrak{A}$, and to 1' as the *identity element* of $\mathfrak{A}$.

3.3 Useful facts and more definitions

Arithmetic In every relation algebra, the following formulas are equivalent: $x;y \leq z$, $\breve{x};\overline{z} \leq \overline{y}$, and $\overline{z};\breve{y} \leq \overline{x}$. This was first proved for binary relations by De Morgan [8], [9, p224], who called it Theorem K. Every relation algebra satisfies the following two variations on De Morgan's Theorem K:

(1) the formulas $x \cdot y;z = 0$, $y \cdot x;\breve{z} = 0$, and $z \cdot \breve{y};x = 0$ are equivalent,
(2) $x \cdot y;z = x \cdot (y \cdot x;\breve{z});(z \cdot \breve{y};x)$ and $0 = 0;x = x;0$.

Some additional theorems which hold in all relation algebras are

(3) $\breve{\overline{x}} = \overline{\breve{x}}$,
(4) $\breve{1'} = 1'$, $\breve{0} = 0$, $\breve{1} = 1$,
(5) $(x \cdot y)^{\smile} = \breve{x} \cdot \breve{y}$,
(6) if $x \leq 1'$ then $x;y \cdot z = y \cdot x;z$ and $y;x \cdot z = y \cdot z;x$,
(7) if $x, y \leq 1'$ then $x;y = x \cdot y$,
(8) if $x \leq y$ then $\breve{x} \leq \breve{y}$, $x;z \leq y;z$, and $z;x \leq z;y$.

Because of (3) we will always use "$\breve{\overline{x}}$" in preference to "$\overline{\breve{x}}$".

Proper relation algebras and representability A relation algebra $\mathfrak{A}$ is *proper* if its universe is a set of binary relations, and its operations coincide with the usual set-theoretic operations on these relations. In more detail, this means that 0 is the empty relation, 1 is a relation which contains all others in A, 1' is the identity relation over the field of 1, and, for all relations $x, y \in A$, $x + y$ is the union of x and y, $x \cdot y$ is the intersection of x and y, $\overline{x}$ is the complement of x with respect to 1, $x;y$ is the relative product of x and y, and $\breve{x}$ is the converse of x. It is easy to show that the Boolean unit 1 in a proper relation algebra $\mathfrak{A}$ is an equivalence relation. The field of the equivalence relation 1 is called the *base* of $\mathfrak{A}$. If 1 has exactly one equivalence class X, then the $\mathfrak{A}$ is called *an algebra of relations on* X or a *proper relation algebra on* X.

Every equivalence relation E is the Boolean unit of a proper relation algebra. Indeed, the set of all binary relations included in E forms a proper relation algebra, denoted $\mathfrak{Sb}(E)$. If E has exactly one equivalence class X, then the resulting algebra is denoted by $\mathfrak{Re}(X)$. Thus $\mathfrak{Re}(X)$ has all possible binary relations on X in its universe, and $\mathfrak{Re}(X) = \mathfrak{Sb}(X \times X)$. For example, $\mathfrak{Re}(\emptyset)$, $\mathfrak{Re}(\{a\})$, and $\mathfrak{Re}(\{a, b\})$ are relation algebras with exactly 1, 2, and 16 elements, respectively.

A *representation* of a relation algebra $\mathfrak{A}$ is an isomorphism, in the usual model-theoretic or algebraic sense, from $\mathfrak{A}$ onto a subalgebra of a proper relation algebra $\mathfrak{B}$. We refer to the base of $\mathfrak{B}$ as the *base* of the representation. We also require a notion of isomorphism for representations, which is slightly different from the model-theoretic notion, since representations are themselves isomorphisms. Two representations R and R' of a relation algebra $\mathfrak{A}$ (with bases B and B', respectively) are *isomorphic* if there is a one-to-one correspondence $h : B \to B'$ such that $R(x) = h;R'(x);\breve{h}$, equivalently

$R'(x) = \breve{h};R(x);h$, for every $x \in A$. Note that if R and R' are isomorphic representations with bases B and B', respectively, then the cardinalities of B and B' must be the same.

A relation algebra $\mathfrak{A}$ is *representable* if it has a representation, and *nonrepresentable* if it does not. In contrast to the fact that every Boolean algebra is isomorphic to a Boolean algebra of sets (by Stone's Theorem), there are nonrepresentable relation algebras, first found by Lyndon [33].

Subalgebras of proper relation algebras are proper, and it is easy to show that direct products of proper relation algebras are isomorphic to proper relation algebras. It follows that the class of representable relation algebras is closed under the formation of subalgebras and direct products. Tarski [60] showed that it is also closed under the formation of homomorphic images, and by Birkhoff's Theorem [3, 11.9] is therefore an equational class, that is, it has an axiomatization consisting entirely of identities. Monk [48] proved that every set of identities which axiomatizes the class of representable relation algebras must be infinite, and Lyndon [34] produced an explicit recursive set of identities which does so.

Simplicity By definition, a relation algebra $\mathfrak{A}$ is *simple* if it has at least two elements and exactly two congruence relations. Congruence relations arise from homomorphisms. If h is a homomorphism from an algebra $\mathfrak{A}$ into an algebra $\mathfrak{B}$, then $\{\langle x, y\rangle : h(x) = h(y)\}$ is a congruence relation on $\mathfrak{A}$. Therefore, a simple relation algebra can have no nontrivial homomorphisms. Indeed, every homomorphism h from a simple relation algebra $\mathfrak{A}$ to a relation algebra $\mathfrak{B}$ is either an isomorphism of $\mathfrak{A}$ onto a subalgebra of $\mathfrak{B}$ (and the congruence relation of h is the identity relation on the universe A of $\mathfrak{A}$), or else $\mathfrak{B}$ has exactly one element (and the congruence relation of h is the relation $A \times A$). It turns out that simplicity for relation algebras has a nice characterization. A relation algebra $\mathfrak{A}$ is simple just in case $0 \neq 1$ and $1;x;1 = 1$ for every nonzero element x of $\mathfrak{A}$ [19, 4.10]. $\mathfrak{Re}(\emptyset)$ has exactly one element, so it is not simple, but if X is not empty, then it is easy to check, using the characterization of simplicity, that $\mathfrak{Re}(X)$ is simple. On the other hand, if X and Y are disjoint nonempty sets, then the proper relation algebra $\mathfrak{Sb}((X \times X) \cup (Y \times Y))$ is not simple, since there is a nontrivial homomorphism h which maps $\mathfrak{Sb}((X \times X) \cup (Y \times Y))$ onto $\mathfrak{Sb}(X \times X)$, defined by $h(x) = x \cap (X \times X)$ for every $x \subseteq (X \times X) \cup (Y \times Y)$. (In fact, $\mathfrak{Sb}((X \times X) \cup (Y \times Y))$ is isomorphic to the direct product of $\mathfrak{Sb}(X \times X)$ with $\mathfrak{Sb}(Y \times Y)$, and h is just the projection function onto the first component of the product.) All the relation algebras in Section 4 are simple.

It turns out that a simple relation algebra $\mathfrak{A}$ is representable if and only if there is a representation R which embeds $\mathfrak{A}$ into $\mathfrak{Re}(X)$ for some set X [19, 4.26]. If R is such a representation, we say that R is a *representation of* $\mathfrak{A}$ *over the set* X.

If a simple relation algebra $\mathfrak{A}$ has a representation R over a set X, then $\mathfrak{A}$ automatically has representations which are not over *any* set. This is so because, for example, $\mathfrak{Re}(X)$ can be embedded into $\mathfrak{Sb}((X' \times X') \cup (X'' \times X''))$, where X' and X'' are two disjoint copies of X (with $x \in X$ corresponding to $x' \in X'$ and $x'' \in X''$). We obtain this embedding by sending $\{\langle x_0, x_1\rangle\}$ to $\{\langle x'_0, x'_1\rangle\} \cup \{\langle x''_0, x''_1\rangle\}$, and extending this map in the natural way to all binary relations on X. By composing R with this embedding, we get a representation which embeds $\mathfrak{A}$ into $\mathfrak{Sb}((X' \times X') \cup (X'' \times X''))$. This representation is not over any set, and is not isomorphic to R (excluding trivial exceptions). This observation has a crucial effect on the wording of Theorem 4.1, so we illustrate it with an example. There is a relation algebra $\mathfrak{A}$ with four elements, 0, 1', 1, and 0', where $0' = \overline{1'}$, and a representation R over the two-element set $\{a, b\}$, defined by $R(0) = \emptyset$, $R(1') = \{\langle a, a\rangle, \langle b, b\rangle\}$, $R(0') = \{\langle a, b\rangle, \langle b, a\rangle\}$, and $R(1) = \{\langle a, a\rangle, \langle b, b\rangle, \langle a, b\rangle, \langle b, a\rangle\}$. Notice that $R(0');R(0') = R(1')$, so $0';0' = 1'$ in $\mathfrak{A}$. From this equation we can easily show that, for any set X, if $\mathfrak{A}$ has a representation over X, then X also has exactly two elements. We cannot, however, conclude that the base of every representation of $\mathfrak{A}$ has exactly two elements, in view of the following counterexample. Choose two distinct elements a' and b' not in $\{a, b\}$. Define a representation S by setting $S(0) = \emptyset$, $S(1') = \{\langle a, a\rangle, \langle b, b\rangle, \langle a', a'\rangle, \langle b', b'\rangle\}$, $S(0') = \{\langle a, b\rangle, \langle b, a\rangle, \langle a', b'\rangle, \langle b', a'\rangle\}$, and $S(1) = \{\langle a, a\rangle, \langle b, b\rangle, \langle a, b\rangle, \langle b, a\rangle, \langle a', a'\rangle, \langle b', b'\rangle, \langle a', b'\rangle, \langle b', a'\rangle\}$. One can now check that S is an embedding of $\mathfrak{A}$ into $\mathfrak{Sb}((\{a, b\} \times \{a, b\}) \cup (\{a', b'\} \times \{a', b'\}))$. The base of R is $\{a, b\}$, but the base of S is $\{a, b, a', b'\}$, so R and S are not isomorphic, because the bases of isomorphic representations must have the same cardinality. Furthermore, S cannot be isomorphic to any representation of $\mathfrak{A}$ over any set, because $\mathfrak{A}$ can be represented only over sets of cardinality 2. By generalizing the construction of S, we can produce representations of $\mathfrak{A}$ with bases of arbitrarily large cardinality, including all finite even cardinals and all infinite cardinals. Thus $\mathfrak{A}$ has, up to isomorphism, only one representation over a set, but many nonisomorphic representations.

Atoms An *atom* of a relation algebra $\mathfrak{A}$ is a minimal nonzero element, *i.e.*, x is an atom if $x \neq 0$ and for every $y \in A$, either $x \cdot y = 0$ or $x \cdot \overline{y} = 0$. In every relation algebra, if x is an atom, then so is $\breve{x}$. A relation algebra is *atomic* if every nonzero element has an atom below it. In every relation algebra, relative multiplication and conversion distribute over arbitrary sums. It follows that in every atomic relation algebra, the relative product of any two elements and the converse of any element can be determined from the relative products and converses of the atoms below those elements. This fact is very useful in the presentation of finite relation algebras. Every finite relation algebra is atomic, and its Boolean part is completely determined by the number of atoms. All that remains is to specify the identity element 1' and the relative products and converses of atoms, and this is all a presentation needs to contain. We present four finite relation algebras in this manner in Section 4.

Equivalence elements and relativization These notions are used in the proof of Theorem 4.6.

An element z of a relation algebra $\mathfrak{A}$ is an *equivalence element* if $z = \breve{z} = z;z$. If $\mathfrak{A}$ is a proper relation algebra, then z is an equivalence element if and only if z is an equivalence relation.

The *relativization of* $\mathfrak{A}$ *to* z is the algebra with universe $\{x : z \geq x \in A\}$ and operations $+^z, \cdot^z, {}^{-z}, ;^z, {}^{\smile z}, 1\text{'}^z$ defined by

$$\begin{aligned} x +^z y &= x + y, \\ x \cdot^z y &= x \cdot y, \\ x^{-z} &= \overline{x} \cdot z, \\ x ;^z y &= x ; y \cdot z, \\ x^{\smile z} &= \breve{x} \cdot z, \\ 1\text{'}^z &= 1\text{'} \cdot z. \end{aligned}$$

The relativization of $\mathfrak{A}$ to z is defined for any element z, but may not even be a relation algebra [38]. However, if z is an equivalence element of $\mathfrak{A}$, then the relativization of $\mathfrak{A}$ to z is also a relation algebra [19, p132].

4 Examples of relation algebras

4.1 The Point Algebra

The *Point Algebra* ***PA*** is a finite relation algebra with exactly 3 atoms (hence 8 elements). Its atoms are 1', b, and $\breve{b}$ ("b" for "before"). Since the only atoms of ***PA*** are 1', b, and $\breve{b}$, we have $1 = 1\text{'} + b + \breve{b}$. Notice, by our choice of notation, that we have already specified the identity element (it's one of the atoms) and the converses of all the atoms (since $\breve{1\text{'}} = 1\text{'}$ and $\breve{\breve{x}} = x$ in every relation algebra). The relative products of atoms are given by the table

;	1'	b	$\breve{b}$
1'	1'	b	$\breve{b}$
b	b	b	1
$\breve{b}$	$\breve{b}$	1	$\breve{b}$

to be read thus:

;	$\cdots$	y
$\vdots$		$\vdots$
x	$\cdots$	$x;y$

The Point Algebra is representable. Let $\mathbb{Q}$ be the set of rational numbers with the usual ordering. We can get a representation $R^{\boldsymbol{PA}}$ of the ***PA*** just by setting

$$\begin{aligned} R^{\boldsymbol{PA}}(1\text{'}) &= \{\langle q, r\rangle : q = r,\ q, r \in \mathbb{Q}\}, \\ R^{\boldsymbol{PA}}(b) &= \{\langle q, r\rangle : q < r,\ q, r \in \mathbb{Q}\}, \end{aligned}$$

since the other values of $R^{\boldsymbol{PA}}$ are determined by these two in the natural way, namely,

$$\begin{aligned} R^{\boldsymbol{PA}}(0) &= \emptyset, \\ R^{\boldsymbol{PA}}(\breve{b}) &= R^{\boldsymbol{PA}}(b)^{\smile}, \\ R^{\boldsymbol{PA}}(1\text{'} + b) &= R^{\boldsymbol{PA}}(1\text{'}) \cup R^{\boldsymbol{PA}}(b), \\ R^{\boldsymbol{PA}}(b + \breve{b}) &= R^{\boldsymbol{PA}}(b) \cup R^{\boldsymbol{PA}}(b)^{\smile}, \\ R^{\boldsymbol{PA}}(1\text{'} + \breve{b}) &= R^{\boldsymbol{PA}}(1\text{'}) \cup R^{\boldsymbol{PA}}(b)^{\smile}, \\ R^{\boldsymbol{PA}}(1\text{'} + b + \breve{b}) &= R^{\boldsymbol{PA}}(1\text{'}) \cup R^{\boldsymbol{PA}}(b) \cup R^{\boldsymbol{PA}}(b)^{\smile}. \end{aligned}$$

¿From these equations and the definitions of $R^{\boldsymbol{PA}}(1')$ and $R^{\boldsymbol{PA}}(b)$ we get

$$\begin{aligned}
R^{\boldsymbol{PA}}(\breve{b}) &= \{\langle q,r\rangle : q > r,\ q,r \in \mathbb{Q}\},\\
R^{\boldsymbol{PA}}(1' + b) &= \{\langle q,r\rangle : q \leq r,\ q,r \in \mathbb{Q}\},\\
R^{\boldsymbol{PA}}(b + \breve{b}) &= \{\langle q,r\rangle : q \neq r,\ q,r \in \mathbb{Q}\},\\
R^{\boldsymbol{PA}}(1' + \breve{b}) &= \{\langle q,r\rangle : q \geq r,\ q,r \in \mathbb{Q}\},\\
R^{\boldsymbol{PA}}(1' + b + \breve{b}) &= \mathbb{Q} \times \mathbb{Q}.
\end{aligned}$$

Part of the proof that $R^{\boldsymbol{PA}}$ is indeed a representation is the verification that the relations in $\{R^{\boldsymbol{PA}}(x) : x \in \boldsymbol{PA}\}$ form a proper relation algebra. This algebra is finite even though the base of the representation is infinite. (On the other hand, there can be no infinite algebra on a finite base, since there are only finitely many relations on a finite set.)

Theorem 4.1 *If R is a representation of the **PA** over a set V, then $\langle V, R(b)\rangle$ is a dense linear ordering without endpoints. Further, all representations of the **PA** over a countable set are isomorphic to the representation $R^{\boldsymbol{PA}}$.*

Proof: Assume R is a representation of the $\boldsymbol{PA}$ over a set V, *i.e.*, R embeds $\boldsymbol{PA}$ into $\mathfrak{Re}(V)$. Then $R(b)$ is transitive because $b;b \leq b$, $R(b)$ is dense because $b \leq b;b$, $R(b)$ is strict because $b \cdot 1' = 0$, $R(b)$ is total because $b + \breve{b} + 1' = 1$, and $R(b)$ has no maximum or minimum because $1 = b;1$ and $1 = 1;b$. Therefore $\langle V, R(b)\rangle$ is a dense linear ordering without endpoints. In case V is countable, there is a model-theoretic isomorphism h from $\langle V, R(b)\rangle$ onto $\langle \mathbb{Q}, <\rangle$. We use h to show that the representations R and $R^{\boldsymbol{PA}}$ are isomorphic. That h is a one-to-one correspondence between V and $\mathbb{Q}$ can be expressed by either of the equivalent equations $R^{\boldsymbol{PA}}(1') = \breve{h};R(1');h$ or $R(1') = h;R^{\boldsymbol{PA}}(1');\breve{h}$. That h maps the relation $R(b)$ onto $<$ can be expressed, according to the definition of $R^{\boldsymbol{PA}}(b)$, by either of the equivalent equations $R^{\boldsymbol{PA}}(b) = \breve{h};R(b);h$ or $R(b) = h;R^{\boldsymbol{PA}}(b);\breve{h}$. So far we know, in case x is either 1' or b, that $R^{\boldsymbol{PA}}(x) = \breve{h};R(x);h$ and $R(x) = h;R^{\boldsymbol{PA}}(x);\breve{h}$, and from these two special cases we can show that the last two equations hold for every $x \in \boldsymbol{PA}$, thanks to the atom structure of the $\boldsymbol{PA}$. ∎

Corollary 4.2 ***PA*** *has no representation over a finite set.*

4.2 The Interval Algebra

The *Interval Algebra* $\boldsymbol{IA}$ has 13 atoms: 1', p, $\breve{p}$, d, $\breve{d}$, o, $\breve{o}$, m, $\breve{m}$, s, $\breve{s}$, f, and $\breve{f}$. (The letters used to denote the atoms come from the words "precedes", "during", "overlaps", "meets", "starts", and "finishes".) The relative products of atoms in the $\boldsymbol{IA}$ are given in the following table. Some definitions of elements used in the table are given after the table.

$;$	1’	p	$\breve{p}$	d	$\breve{d}$	o	$\breve{o}$	m	$\breve{m}$	s	$\breve{s}$	f	$\breve{f}$
1’	1’	p	$\breve{p}$	d	$\breve{d}$	o	$\breve{o}$	m	$\breve{m}$	s	$\breve{s}$	f	$\breve{f}$
p	p	p	1	ρ	p	p	ρ	p	ρ	p	p	ρ	p
$\breve{p}$	$\breve{p}$	1	$\breve{p}$	$\breve{\lambda}$	$\breve{p}$	$\breve{\lambda}$	$\breve{p}$	$\breve{\lambda}$	$\breve{p}$	$\breve{\lambda}$	$\breve{p}$	$\breve{p}$	$\breve{p}$
d	d	p	$\breve{p}$	d	1	ρ	$\breve{\lambda}$	p	$\breve{p}$	d	$\breve{\lambda}$	d	ρ
$\breve{d}$	$\breve{d}$	λ	$\breve{\rho}$	ν	$\breve{d}$	γ	$\breve{\beta}$	γ	$\breve{\beta}$	γ	$\breve{d}$	$\breve{\beta}$	$\breve{d}$
o	o	p	$\breve{\rho}$	β	λ	α	ν	p	$\breve{\beta}$	o	γ	β	α
$\breve{o}$	$\breve{o}$	λ	$\breve{p}$	$\breve{\gamma}$	$\breve{\rho}$	ν	$\breve{\alpha}$	γ	$\breve{p}$	$\breve{\gamma}$	$\breve{\alpha}$	$\breve{o}$	$\breve{\beta}$
m	m	p	$\breve{\rho}$	β	p	p	β	p	$f^\sharp$	m	m	β	p
$\breve{m}$	$\breve{m}$	λ	$\breve{p}$	$\breve{\gamma}$	$\breve{p}$	$\breve{\gamma}$	$\breve{p}$	$s^\sharp$	$\breve{p}$	$\breve{\gamma}$	$\breve{p}$	$\breve{m}$	$\breve{m}$
s	s	p	$\breve{p}$	d	λ	α	$\breve{\gamma}$	p	$\breve{m}$	s	$s^\sharp$	d	α
$\breve{s}$	$\breve{s}$	λ	$\breve{p}$	$\breve{\gamma}$	$\breve{d}$	γ	$\breve{o}$	γ	$\breve{m}$	$s^\sharp$	$\breve{s}$	$\breve{o}$	$\breve{d}$
f	f	p	$\breve{p}$	d	$\breve{\rho}$	β	$\breve{\alpha}$	m	$\breve{p}$	d	$\breve{\alpha}$	f	$f^\sharp$
$\breve{f}$	$\breve{f}$	p	$\breve{\rho}$	β	$\breve{d}$	o	$\breve{\beta}$	m	$\breve{\beta}$	o	$\breve{d}$	$f^\sharp$	$\breve{f}$

$$
\begin{aligned}
\alpha &= p + o + m \\
\beta &= d + o + s \\
\gamma &= \breve{d} + o + \breve{f} \\
f^\sharp &= 1\text{'} + f + \breve{f} \\
s^\sharp &= 1\text{'} + s + \breve{s} \\
\rho &= \alpha + \beta = p + d + o + m + s \\
\lambda &= \alpha + \gamma = p + \breve{d} + o + m + \breve{f} \\
\nu &= \gamma + \breve{\gamma} + s^\sharp = \beta + \breve{\beta} + f^\sharp = 1\text{'} + d + \breve{d} + o + \breve{o} + s + \breve{s} + f + \breve{f}
\end{aligned}
$$

Let $\mathbb{L} = R^{\boldsymbol{PA}}(b) = \{\langle q, r\rangle : q, r \in \mathbb{Q},\ q < r\}$. A representation $R^{\boldsymbol{IA}}$ of the ***IA*** over $\mathbb{L}$ is given by the following table.

equals:	$R^{\boldsymbol{IA}}(1\text{'}) = \{\langle\langle q,r\rangle, \langle q',r'\rangle\rangle : q = q' < r = r',\ q,r,q',r' \in \mathbb{Q}\}$
precedes:	$R^{\boldsymbol{IA}}(p) = \{\langle\langle q,r\rangle, \langle q',r'\rangle\rangle : q < r < q' < r',\ q,r,q',r' \in \mathbb{Q}\}$
during:	$R^{\boldsymbol{IA}}(d) = \{\langle\langle q,r\rangle, \langle q',r'\rangle\rangle : q' < q < r < r',\ q,r,q',r' \in \mathbb{Q}\}$
overlaps:	$R^{\boldsymbol{IA}}(o) = \{\langle\langle q,r\rangle, \langle q',r'\rangle\rangle : q < q' < r < r',\ q,r,q',r' \in \mathbb{Q}\}$
meets:	$R^{\boldsymbol{IA}}(m) = \{\langle\langle q,r\rangle, \langle q',r'\rangle\rangle : q < r = q' < r',\ q,r,q',r' \in \mathbb{Q}\}$
starts:	$R^{\boldsymbol{IA}}(s) = \{\langle\langle q,r\rangle, \langle q',r'\rangle\rangle : q = q' < r < r',\ q,r,q',r' \in \mathbb{Q}\}$
finishes:	$R^{\boldsymbol{IA}}(f) = \{\langle\langle q,r\rangle, \langle q',r'\rangle\rangle : q' < q < r = r',\ q,r,q',r' \in \mathbb{Q}\}$

Next we have a few results on subalgebras and representations of the ***IA***.

Theorem 4.3 *The **PA** can be embedded in the **IA**. The **IA** has no representation over a finite set.*

Proof: Set

$$
\begin{aligned}
I(1') &= 1', \\
I(b) &= \lambda + s = p + \breve{d} + o + m + s + \breve{f}, \\
I(\breve{b}) &= \breve{\lambda} + \breve{s} = \breve{p} + d + \breve{o} + \breve{m} + \breve{s} + f,
\end{aligned}
$$

and extend I in the natural way to all of the ***PA***. Since the ***IA*** has a subalgebra isomorphic to the ***PA***, and the ***PA*** has no representation over a finite set, the same is true for the ***IA***. ∎

Theorem 4.4 *If $x \in \{p, \breve{p}, m, \breve{m}, o, \breve{o}\}$, then the **IA** is generated by x.*

Proof: Let $x \in \{p, \breve{p}, m, \breve{m}, o, \breve{o}\}$. Note that the ***IA*** is generated by x just in case it is generated by $\breve{x}$, so we need only show p, m, and o generate the ***IA***. That m generates the ***IA*** follows immediately from the following equations, which may be checked against the table of relative products.

$$
\begin{aligned}
1' &= m;\breve{m} \cdot \breve{m};m, \\
p &= m;m, \\
s &= \breve{m};m \cdot m;m;\breve{m}, \\
f &= m;\breve{m} \cdot \breve{m};\breve{m};m, \\
d &= s;f, \\
o &= \breve{f};s.
\end{aligned}
$$

Next we show that p generates the ***IA***. It will follow that o also generates the ***IA***, since $p = o;(o;o \cdot \overline{o})$. First, from the table of relative products we have

$$
\begin{aligned}
\lambda &= \breve{\overline{p}};p, \\
\rho &= p;\breve{\overline{p}},
\end{aligned}
$$

so we use ρ and λ together with p to generate the remaining atoms:

$$
\begin{aligned}
d &= \breve{p};\overline{p} \cdot p;\breve{\overline{p}}, \\
s &= \rho \cdot \overline{\lambda} \cdot \breve{\overline{\lambda}}, \\
f &= \breve{\rho} \cdot \overline{\rho} \cdot \breve{\overline{\rho}}, \\
o &= \breve{f};s, \\
m &= s;\breve{f} \cdot \overline{p} \cdot \overline{o}.
\end{aligned}
$$

∎

Results on the subalgebras generated by the other atoms of the ***IA***, and other elements occurring in the composition table, are summarized in the next theorem. Note that 1' is a symmetric atom in each subalgebra of the ***IA***. The algebra ***CA*** which occurs in the statement of (f) is introduced in the next section.

Theorem 4.5 *In the **IA**:* (a) *d generates a subalgebra with* 8 *atoms,* 4 *of which are symmetric;* (b) *f and s generate isomorphic subalgebras with* 4 *atoms,* 2 *symmetric;* (c) *$f^\sharp$ and $s^\sharp$ generate isomorphic symmetric subalgebras with* 3 *atoms;* (d) *α generates a subalgebra with* 6 *atoms,* 2 *symmetric;* (e) *β and γ generate the whole **IA**;* (f) *ν generates a subalgebra isomorphic to the **CA**.*

Proof: The results may be obtained by straightforward relation-algebraic computations similar to those in the previous theorem. ∎

Theorem 4.6 *If R is a representation of the* ***IA*** *over a countable set V, then R is isomorphic to the representation $R^{\boldsymbol{IA}}$.*

Proof: Referring to the composition table for the ***IA***, it may be seen that $s^\sharp$ and $f^\sharp$ both contain 1', and are equivalence elements. Hence $R(f^\sharp)$ and $R(s^\sharp)$ are equivalence relations on V. The relativizations of the ***IA*** to $s^\sharp$ and $f^\sharp$ are therefore relation algebras. Both of them are isomorphic to ***PA***, as can be seen from inspection of the composition table. Choose some element $x \in V$. Let

$$X = \{y : \langle y, x\rangle \in R(f^\sharp)\},$$
$$Y = \{y : \langle x, y\rangle \in R(m)\}.$$

By definition, X is the $R(f^\sharp)$-equivalence class of x. We claim that Y is an $R(s^\sharp)$-equivalence class. We have to show that if $y, y' \in Y$, then $\langle y, y'\rangle \in R(s^\sharp)$, and also that if $y \in Y$ and $\langle y, y'\rangle \in R(s^\sharp)$ then $y' \in Y$. First assume $y, y' \in Y$. Then $\langle x, y\rangle \in R(m)$ and $\langle x, y'\rangle \in R(m)$, so $\langle y, y'\rangle \in R(m)^{\smile};R(m) = R(\breve{m};m) = R(s^\sharp)$. Second, assume $y \in Y$ and $\langle y, y'\rangle \in R(s^\sharp)$. Then $\langle x, y\rangle \in R(m)$, so $\langle x, y'\rangle \in R(m);R(s^\sharp) = R(m;s^\sharp) = R(m)$. Hence $y' \in Y$ as required, and we have shown that Y is an $R(s^\sharp)$-equivalence class.

Since the relativizations of ***IA*** to both $s^\sharp$ and to $f^\sharp$ are isomorphic to ***PA*** and V is countable, by Theorem 4.1 we have

$\langle X, R(f^\sharp)\rangle$ and $\langle Y, R(s^\sharp)\rangle$ are both countable dense linear orderings without endpoints.

Now suppose that $x' \in X$ and $y' \in Y$. Then $\langle x', x\rangle \in R(f^\sharp)$ and $\langle x, y'\rangle \in R(m)$, so $\langle x', y'\rangle \in R(f^\sharp);R(m) = R(f^\sharp;m) = R(m)$. It follows that $X \times Y \subseteq R(m)$, so X and Y are disjoint, since $m \cdot 1' = 0$. Combining these observations, we conclude

$\langle X \cup Y, R(f^\sharp + s^\sharp + m)\rangle$ is a dense linear ordering without endpoints.

Let $\langle Q', L'\rangle$ be the structure of the nonzero rationals with the usual ordering, *i.e.*, $Q' = \{q : 0 \neq q \in \mathbb{Q}\}$, and $qL'r$ if and only if $q, r \in Q'$ and $q < r$. Since $\langle Q', L'\rangle$ is a dense linear ordering without endpoints, all countable dense linear orderings without endpoints are isomorphic to the rationals, and the rationals are homogeneous [4], we may let g be an isomorphism from $\langle X \cup Y, R(f^\sharp + s^\sharp + m)\rangle$ to $\langle Q', L'\rangle$ which maps X to the negative rationals and Y to the positive rationals. We use g to define a one-to-one correspondence h from V to the base $\mathbb{L}$ of $R^{\boldsymbol{IA}}$. For each $z \in V$ we shall let $h(z) = \langle h_1(z), h_2(z)\rangle$, so now we define h_1 and h_2.

We begin with h_1. Let z be any element of V. We split into three cases. First note that $R(m)$, $R(p)$, and $R(\overline{p} \cdot \overline{m})$ form a partition of $V \times V$, *i.e.*, every pair is in exactly one of these three relations. So we can define h_1 for each case separately. If $z \in Y$, *i.e.*, $\langle x, z\rangle \in R(m)$, then we let $h_1(z) = 0$. Next assume $\langle x, z\rangle \in R(\overline{p} \cdot \overline{m})$. From the table we get $\overline{p} \cdot \overline{m} \leq f^\sharp ; s^\sharp$, so there is some w such that $\langle x, w\rangle \in R(f^\sharp)$ and $\langle w, z\rangle \in R(s^\sharp)$. There can be only one such w. To see this, suppose $\langle x, w'\rangle \in R(f^\sharp)$ and $\langle w', z\rangle \in R(s^\sharp)$. Then $\langle w, w'\rangle \in R(f^\sharp)^\smile ; R(f^\sharp) \cap R(s^\sharp) ; R(s^\sharp)^\smile = R(f^{\sharp\smile} ; f^\sharp \cdot s^\sharp ; \breve{s}) = R(1')$, so $w = w'$. $R(f^\sharp)$ is symmetric, so $w \in X$ and $g(w)$ is defined. Let $h_1(z) = g(w)$. Finally assume $\langle x, z\rangle \in R(p)$. Note that $p = m ; m$, so there is some w such that $\langle x, w\rangle \in R(m)$ and $\langle w, z\rangle \in R(m)$. This w is unique since $\breve{m} ; m \cdot m ; \breve{m} = 1'$, and $w \in Y$, so $g(w)$ is defined. We let $h_1(z) = g(w)$. This completes the definition of h_1.

We define h_2 similarly, using the partition of $V \times V$ into $R(f^\sharp)$, $R(\rho)$, and $R(\breve{\rho})$. If $\langle x, z\rangle \in R(f^\sharp)$, then $z \in X$ and we set $h_2(z) = 0$. Suppose $\langle x, z\rangle \in R(\rho)$. Since $\rho = m ; f^\sharp$, there is some w such that $\langle x, w\rangle \in R(m)$ and $\langle w, z\rangle \in R(f^\sharp)$. This w is in Y and w is unique since $\breve{m} ; m \cdot f^\sharp ; f^{\sharp\smile} = 1'$. In this case we let $h_2(z) = g(w)$. Finally, assume $\langle x, z\rangle \in R(\breve{\rho})$. We have $\breve{\rho} = f^\sharp ; \breve{m}$ so there is a w such that $\langle x, w\rangle \in R(f^\sharp)$ and $\langle w, z\rangle \in R(\breve{m})$. Then $w \in X$ and is unique since $f^{\sharp\smile} ; f^\sharp \cdot \breve{m} ; m = 1'$. Set $h_2(z) = g(w)$. This completes the construction of h_2, and hence also h.

The remainder of the proof is a verification that h works properly. For example, we show that $h_2(z) = h_1(z')$ if and only if $\langle z, z'\rangle \in R(m)$. We consider the three cases for the definition of $h_1(z')$. We show the left-to-right directions only, since the necessary reasoning is reversible.

a: $\langle x, z'\rangle \in R(m)$: Thus $h_1(z') = 0 = h_2(z)$ so $z \in X$ and thus $\langle z, x\rangle \in R(f^\sharp)$ and $\langle z, z'\rangle \in R(f^\sharp) ; R(m) = R(f^\sharp ; m) = R(m)$.

b: $\langle x, z'\rangle \in R(p)$: Then $h_1(z') = g(w) = h_2(z) \neq 0$ where $\langle x, w\rangle \in R(m)$ and $\langle w, z'\rangle \in R(m)$. $\langle x, w\rangle \in R(m)$ implies we're in case $\langle x, z\rangle \in R(\rho)$ of the definition of $h_2(z)$, hence $\langle w, z\rangle \in R(f^\sharp)$, so $\langle z, w\rangle \in R(f^\sharp)$ which with $\langle w, z'\rangle \in R(m)$ yields $\langle z, z'\rangle \in R(f^\sharp ; m) = R(m)$.

c: $\langle x, z'\rangle \in R(\overline{p} \cdot \overline{m})$: So $\langle x, w\rangle \in R(f^\sharp), \langle w, z'\rangle \in R(s^\sharp)$, where $h_1(z') = g(w) = h_2(z) \neq 0$. $\langle x, w\rangle \in R(f^\sharp)$ puts us in the case $\langle x, z\rangle \in R(\breve{\rho})$ of the definition of $h_2(z)$, so $\langle w, z\rangle \in R(\breve{m})$ and $\langle z, z'\rangle \in R(m ; s^\sharp) = R(m)$.

The reasoning for the other twelve relations is similar. ∎

4.3 The Containment Algebra

The *Containment Algebra* ***CA*** has five atoms: 1', c, $\breve{c}$, n, and a (from "contained in", "nonempty intersection", and "apart"), where $\breve{n} = n$ and $\breve{a} = a$. The relative products of atoms are

;	1'	c	$\breve{c}$	n	a
1'	1'	c	$\breve{c}$	n	a
c	c	c	1	$c + n + a$	a
$\breve{c}$	$\breve{c}$	$1' + c + \breve{c} + n$	$\breve{c}$	$\breve{c} + n$	$\breve{c} + n + a$
n	n	$c + n$	$\breve{c} + n + a$	1	$\breve{c} + n + a$
a	a	$c + n + a$	a	$c + n + a$	1

The $\boldsymbol{CA}$ is isomorphic to a subalgebra of the $\boldsymbol{IA}$. To get an isomorphism I which embeds the $\boldsymbol{CA}$ into the $\boldsymbol{IA}$, just set

$$
\begin{aligned}
I(1') &= 1', \\
I(c) &= s + f + d, \\
I(\breve{c}) &= \breve{s} + \breve{f} + \breve{d}, \\
I(n) &= o + \breve{o}, \\
I(a) &= p + \breve{p} + m + \breve{m},
\end{aligned}
$$

and extend I to all of the $\boldsymbol{CA}$ in the usual way. It follows from the representability of the $\boldsymbol{IA}$ that the $\boldsymbol{CA}$ is also representable. Composing the embedding I with the representation $R^{\boldsymbol{IA}}$ of the $\boldsymbol{IA}$ over $\mathbb{L}$ yields a representation $R^{\boldsymbol{CA}} = I; R^{\boldsymbol{IA}}$ of the $\boldsymbol{CA}$ over $\mathbb{L}$. To get an essentially different representation of $\boldsymbol{CA}$, let Ω be the set of all open subsets of the reals $\mathbb{R}$. Then set

$$
\begin{aligned}
R_o^{\boldsymbol{CA}}(1') &= \{\langle X, Y\rangle : X, Y \in \Omega,\ X = Y\}, \\
R_o^{\boldsymbol{CA}}(c) &= \{\langle X, Y\rangle : X, Y \in \Omega,\ X \subset Y\}, \\
R_o^{\boldsymbol{CA}}(\breve{c}) &= \{\langle X, Y\rangle : X, Y \in \Omega,\ X \supset Y\}, \\
R_o^{\boldsymbol{CA}}(a) &= \{\langle X, Y\rangle : X, Y \in \Omega,\ X \cap Y = \emptyset\}, \\
R_o^{\boldsymbol{CA}}(n) &= \{\langle X, Y\rangle : X, Y \in \Omega,\ X \cap Y \neq \emptyset,\ X \not\subseteq Y,\ Y \not\subseteq X\},
\end{aligned}
$$

and extend $R_o^{\boldsymbol{CA}}$ to all of $\boldsymbol{CA}$. The $\boldsymbol{CA}$ has no representation over a finite set, for if R is any representation of the $\boldsymbol{CA}$, then $R(c)$ is a transitive dense relation disjoint from Id. There is no such relation on a finite set.

4.4 The Hard Algebra

The *Hard Algebra* $\boldsymbol{HA}$ has 4 atoms, namely 1', a, b, and c, with converses $\breve{a} = a$, $\breve{b} = b$, and $\breve{c} = c$. The relative products of atoms are given by

;	1'	a	b	c
1'	1'	a	b	c
a	a	$1' + a + b$	$a + b + c$	$b + c$
b	b	$a + b + c$	1	$a + b + c$
c	c	$b + c$	$a + b + c$	1

It follows from [38, Theorem 5.19] that $\boldsymbol{HA}$ is representable over an infinite set, but $\boldsymbol{HA}$ is also representable on a set with 84 elements. To see this, let S be a set containing exactly 9 elements, and let T be the set of subsets of S which contain exactly 3 elements. Thus $|T|$, the cardinality of T, is 84 (9 taken 3 at a time). We get a representation $R^{\boldsymbol{HA}}$ of $\boldsymbol{HA}$ over T by setting

$$
\begin{aligned}
R^{\boldsymbol{HA}}(1') &= \{\langle X, Y\rangle : X, Y \in T,\ |X \cap Y| = 3\}, \\
R^{\boldsymbol{HA}}(a) &= \{\langle X, Y\rangle : X, Y \in T,\ |X \cap Y| = 2\}, \\
R^{\boldsymbol{HA}}(b) &= \{\langle X, Y\rangle : X, Y \in T,\ |X \cap Y| = 1\}, \\
R^{\boldsymbol{HA}}(c) &= \{\langle X, Y\rangle : X, Y \in T,\ |X \cap Y| = 0\}.
\end{aligned}
$$

A key fact in the verification that R^{HA} is a representation is that there cannot be two disjoint 3-element sets each of which shares two elements with a third 3-element set. This is why, in the ***HA***, we have $0 = c \cdot a;a = a \cdot c;a = a \cdot a;c$. Incidentally, representations over larger sets can be obtained by letting S contain more than 9 elements, but fewer than 9 will not work.

The ***HA*** was invented as part of the second author's construction of the hard examples in Section 7; its finite representability was discovered only later, by accident, during the investigation of a certain class of proper relation algebras.

5 Matrices over relation algebras

We may formulate concepts of binary constraint problems in a more general manner using relation algebras. An n-by-n *matrix over* $\mathfrak{A}$, where $\mathfrak{A}$ is a relation algebra, is an n-by-n matrix M of elements of $\mathfrak{A}$. The *order* of M is $\mathrm{ord}(M) = n$. If M satisfies the *diagonal condition*, namely $M_{ii} \leq 1'$ for all $i < n$, then M is called a *constraint matrix over* $\mathfrak{A}$ (or an *RA matrix over* $\mathfrak{A}$ or, briefly, an $\mathfrak{A}$-*matrix*). Except for some minor notational modifications, the definitions of intersection ($M \cdot N$), relative product ($M;N$), and converse ($\breve{M}$) for matrices of relations also apply to matrices over arbitrary relation algebras, as do the concepts of reduction ($N \leq M$), symmetry ($M = \breve{M}$), path-consistency ($M \leq M^2$), and closure ($M = \breve{M} \leq M^2$). Next we present some useful elementary results concerning constraint matrices over a fixed relation algebra $\mathfrak{A}$.

Theorem 5.1 *Let M be an $\mathfrak{A}$-matrix. Then $M^2 \leq M$, and M is path-consistent if and only if $M = M^2$.*

Proof: For all $i, j < \mathrm{ord}(M)$, $(M^2)_{ij} \leq M_{ii};M_{ij} \leq 1';M_{ij} = M_{ij}$. Thus $M^2 \leq M$. By definition, M is path-consistent if and only if $M \leq M^2$, so M is path-consistent if and only if $M = M^2$. ∎

Theorem 5.2 *If M is an $\mathfrak{A}$-matrix, then $\breve{\breve{M}} = M$.*

Proof: For all $i, j < \mathrm{ord}(M)$, $(\breve{\breve{M}})_{ij} = ((\breve{M})_{ji})^{\smile} = ((M_{ij})^{\smile})^{\smile} = M_{ij}$. ∎

Theorem 5.3 *If M and N are $\mathfrak{A}$-matrices, then $(M \cdot N)^{\smile} = \breve{M} \cdot \breve{N}$.*

Proof: For all $i, j < \mathrm{ord}(M)$, $((M \cdot N)^{\smile})_{ij} = ((M \cdot N)_{ji})^{\smile} = (M_{ji} \cdot N_{ji})^{\smile} = (M_{ji})^{\smile} \cdot (N_{ji})^{\smile} = \breve{M}_{ij} \cdot \breve{N}_{ij} = (\breve{M} \cdot \breve{N})_{ij}$. ∎

Theorem 5.4 *If M and N are $\mathfrak{A}$-matrices, then $(M;N)^{\smile} = \breve{N};\breve{M}$.*

Proof: For all $i,j < \mathrm{ord}(M)$ we have $((M\,;N)^{\smile})_{ij} = ((M\,;N)_{ji})^{\smile} = (\prod_{k<n} M_{jk}\,;N_{ki})^{\smile} = \prod_{k<n}(M_{jk}\,;N_{ki})^{\smile} = \prod_{k<n}(N_{ki})^{\smile};(M_{jk})^{\smile} = \prod_{k<n} \breve{N}_{ik}\,;\breve{M}_{kj} = (\breve{N}\,;\breve{M})_{ij}$. ∎

Theorem 5.5 *If M is an $\mathfrak{A}$-matrix, then $M \cdot \breve{M}$ is symmetric.*

Proof: For all $i,j < \mathrm{ord}(M)$, $(M \cdot \breve{M})_{ij} = M_{ij} \cdot \breve{M}_{ij} = ((M_{ij})^{\smile})^{\smile} \cdot \breve{M}_{ij} = (\breve{M}_{ji})^{\smile} \cdot (M_{ji})^{\smile} = (\breve{M}_{ji} \cdot M_{ji})^{\smile} = ((\breve{M} \cdot M)_{ji})^{\smile} = ((\breve{M} \cdot M)^{\smile})_{ij} = ((M \cdot \breve{M})^{\smile})_{ij}$. ∎

Theorem 5.6 *If M is a symmetric $\mathfrak{A}$-matrix of order 3, then $(M^2)^2$ is closed.*

Proof: Let $N = M \cdot M'$, where M' is the matrix

$$\begin{pmatrix} M_{01}\,;M_{11}\,;M_{12}\,;M_{22}\,;M_{20} & M_{00}\,;M_{02}\,;M_{22}\,;M_{21}\,;M_{11} & M_{00}\,;M_{01}\,;M_{11}\,;M_{12}\,;M_{22} \\ M_{11}\,;M_{12}\,;M_{22}\,;M_{20}\,;M_{00} & M_{10}\,;M_{00}\,;M_{02}\,;M_{22}\,;M_{21} & M_{11}\,;M_{10}\,;M_{00}\,;M_{02}\,;M_{22} \\ M_{22}\,;M_{21}\,;M_{11}\,;M_{10}\,;M_{00} & M_{22}\,;M_{20}\,;M_{00}\,;M_{01}\,;M_{11} & M_{20}\,;M_{00}\,;M_{01}\,;M_{11}\,;M_{12} \end{pmatrix}.$$

Obviously, $N \le M$. Since M is symmetric, we have $M = \breve{M}$, so for all $i,j < n$, $(M_{ij})^{\smile} = \breve{M}_{ji} = M_{ji}$. From this we can easily see that N is symmetric as well. For example,

$$\begin{aligned} \breve{N}_{12} &= (N_{21})^{\smile} \\ &= (M_{21} \cdot M_{22}\,;M_{20}\,;M_{00}\,;M_{01}\,;M_{11})^{\smile} \\ &= (M_{21})^{\smile} \cdot (M_{22}\,;M_{20}\,;M_{00}\,;M_{01}\,;M_{11})^{\smile} \\ &= (M_{21})^{\smile} \cdot (M_{11})^{\smile};(M_{01})^{\smile};(M_{00})^{\smile};(M_{20})^{\smile};(M_{22})^{\smile} \\ &= \breve{M}_{12} \cdot \breve{M}_{11}\,;\breve{M}_{10}\,;\breve{M}_{00}\,;\breve{M}_{02}\,;\breve{M}_{22} \\ &= M_{12} \cdot M_{11}\,;M_{10}\,;M_{00}\,;M_{02}\,;M_{22} \\ &= N_{12}. \end{aligned}$$

Next we show N is path-consistent, *i.e.*, $N \le N^2$. The computations make use of axioms (1) and (6) from Section 3.2, and identities (2), (5), (6), (7), and (8) from Section 3.3. The computations are all similar, but fall into different types, depending on whether subscripts are distinct or repeated. Out of 27 computations we present four which illustrate the various types.

$$\begin{aligned} N_{01} &= M_{01} \cdot M_{00}\,;M_{02}\,;M_{22}\,;M_{21}\,;M_{11} \\ &= M_{01} \cdot (M_{00}\,;M_{02} \cdot M_{01}\,;(M_{22}\,;M_{21}\,;M_{11})^{\smile})\,;(M_{22}\,;M_{21}\,;M_{11} \cdot (M_{00}\,;M_{02})^{\smile};M_{01}) \\ &\le (M_{00}\,;M_{02} \cdot M_{01}\,;(M_{22}\,;M_{21}\,;M_{11})^{\smile})\,;(M_{22}\,;M_{21}\,;M_{11} \cdot (M_{00}\,;M_{02})^{\smile};M_{01}) \\ &= (M_{00}\,;M_{02} \cdot M_{01}\,;M_{11}\,;M_{12}\,;M_{22})\,;(M_{22}\,;M_{21}\,;M_{11} \cdot M_{20}\,;M_{00}\,;M_{01}) \\ &= (M_{02} \cdot M_{00}\,;M_{01}\,;M_{11}\,;M_{12}\,;M_{22})\,;(M_{22}\,;M_{21} \cdot M_{20}\,;M_{00}\,;M_{01}\,;M_{11}) \\ &= (M_{02} \cdot M_{00}\,;M_{01}\,;M_{11}\,;M_{12}\,;M_{22})\,;(M_{21} \cdot M_{22}\,;M_{20}\,;M_{00}\,;M_{01}\,;M_{11}) \\ &= N_{02}\,;N_{21}. \end{aligned}$$

This first computation can be used to shorten the next one.

$$\begin{aligned}
N_{10} &= \breve{N}_{10} = (N_{01})\breve{} \leq (N_{02}\,;N_{21})\breve{} = (N_{21})\breve{}\,;(N_{02})\breve{} = \breve{N}_{12}\,;\breve{N}_{20} = N_{12}\,;N_{20}.\\
N_{00} &= M_{00}\cdot M_{01}\,;M_{11}\,;M_{12}\,;M_{22}\,;M_{20}\\
&\leq (M_{01}\cdot M_{00}\,;(M_{11}\,;M_{12}\,;M_{22}\,;M_{20})\breve{})\,;(M_{11}\,;M_{12}\,;M_{22}\,;M_{20}\cdot (M_{01})\breve{}\,;M_{00})\\
&= (M_{01}\cdot M_{00}\,;M_{02}\,;M_{22}\,;M_{21}\,;M_{11})\,;(M_{11}\,;M_{12}\,;M_{22}\,;M_{20}\cdot M_{10}\,;M_{00})\\
&= (M_{01}\cdot M_{00}\,;M_{02}\,;M_{22}\,;M_{21}\,;M_{11})\,;(M_{11}\,;M_{12}\,;M_{22}\,;M_{20}\,;M_{00}\cdot M_{10})\\
&= N_{01}\,;N_{10}.
\end{aligned}$$

Finally, $N_{00} \leq M_{00} \leq 1'$ by the diagonal condition on M, so $N_{00} = N_{00}\cdot N_{00} = N_{00}\,;N_{00}$.

We may conclude that N satisfies the diagonal condition and is closed, since N is both symmetric and path-consistent. From $N \leq M$ it follows that $N^2 \leq M^2$. But N is path-consistent, so $N = N^2 \leq M^2$. This in turn implies $N^2 \leq (M^2)^2$, so, using the path-consistency of N again, we get $N \leq (M^2)^2$. Next we show $(M^2)^2 \leq N$. Again, we only give a sample computation. The others are similar. We have $((M^2)^2)_{01} \leq M_{01}$ by Theorem 5.1, and

$$\begin{aligned}
((M^2)^2)_{01} &\leq (M^2)_{02}\,;(M^2)_{21}\\
&= (M_{00}\,;M_{02}\cdot M_{01}\,;M_{12}\cdot M_{02}\,;M_{22})\,;(M_{20}\,;M_{01}\cdot M_{21}\,;M_{11}\cdot M_{22}\,;M_{21})\\
&\leq (M_{00}\,;M_{02}\cdot M_{02}\,;M_{22})\,;(M_{21}\,;M_{11})\\
&= (M_{02}\cdot M_{00}\,;M_{02}\,;M_{22})\,;(M_{21}\,;M_{11})\\
&\leq M_{00}\,;M_{02}\,;M_{22}\,;M_{21}\,;M_{11},
\end{aligned}$$

so $((M^2)^2)_{01} \leq M_{01}\cdot M_{00}\,;M_{02}\,;M_{22}\,;M_{21}\,;M_{11} = N_{01}$. Thus we have $N = (M^2)^2$. Since N is closed, so is $(M^2)^2$, as desired. ∎

Atomic RA matrices and proto-solvability An $\mathfrak{A}$-matrix M is *atomic* if every M_{ij} is an atom of $\mathfrak{A}$. N is an *atomic reduction* of M if $N \leq M$ and N is atomic. M is *proto-solvable (over $\mathfrak{A}$)* if there is a closed atomic reduction N of M. Such an N is called a *proto-solution of M over $\mathfrak{A}$*. We denote the class of all proto-solutions of M over $\mathfrak{A}$ by $\mathrm{PS}(\mathfrak{A}, M)$.

If $\mathfrak{A}$ is atomic and M is the $\mathfrak{A}$-matrix of order n defined for all $i, j < n$ by

$$M_{ij} = \begin{cases} 1' & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases},$$

then $\mathrm{PS}(\mathfrak{A}, M)$ is the set of n-by-n *basic matrices* of $\mathfrak{A}$, which is denoted by $B_n\mathfrak{A}$ [39, §2(a)], [40, Def. 2(i)], [41, Def. 34(i)], [42, Def. 46]. Thus a matrix of elements of a relation algebra $\mathfrak{A}$ is a *basic* matrix if and only if it is closed, its entries are atoms, and it satisfies the diagonal condition. Basic matrices play an important role in algebraic logic [40], [43], and in algebraic topology [10, §2], where they form the cells of simplicial complexes over an arbitrary group. (The subsets of a group form a relation algebra in a natural way [5, §1].)

Solutions Whenever R is a homomorphism which maps the relation algebra $\mathfrak{A}$ into the relation algebra $\mathfrak{B}$, and M is a matrix of elements of $\mathfrak{A}$, then we define $R(M)$ to be the matrix of elements of $\mathfrak{B}$ with labels $(R(M))_{ij} = R(M_{ij})$ for $i, j < \mathrm{ord}(M)$. If M is an $\mathfrak{A}$-matrix (that is, it satisfies the diagonal condition), then $R(M)$ is a $\mathfrak{B}$-matrix, because the homomorphism R must map the identity element of $\mathfrak{A}$ to that of $\mathfrak{B}$, hence $(R(M))_{ii} = R(M_{ii}) \leq R(1') = 1'$ for all $i < \mathrm{ord}(M)$. If $\mathfrak{B}$ is a proper relation algebra and R is one-to-one, then R is a representation. In this case, we say the $\mathfrak{A}$-matrix M is *R-solvable* if $R(M)$ has a solution, and $\langle a_0, \ldots, a_{\mathrm{ord}(M)-1}\rangle$ is an *R-solution* of M if it is a solution of $R(M)$. If R' is some other representation of $\mathfrak{A}$, then M may be R-solvable but not R'-solvable, or vice versa, as shown by Theorem 5.9. If $\mathfrak{A}$ is already a proper relation algebra, then a solution of M is just an R-solution, where R is the identity isomorphism mapping $\mathfrak{A}$ onto itself.

The relation between proto-solvability and solutions $\mathfrak{A}$-matrices may have proto-solutions, but yet no R-solutions for certain representations R. However, there is a particular case in which proto-solutions do correspond to solutions, and vice versa. A constraint matrix arising from a constraint problem with universe X is an $\mathfrak{Re}(X)$-matrix. In this case, solutions and proto-solutions of the constraint matrix correspond in a one-to-one manner. For example, if M is an $\mathfrak{Re}(X)$-matrix of order 3, then every solution $\langle a_0, a_1, a_2\rangle$ of M has its corresponding proto-solution

$$P(a_0, a_1, a_2) = \begin{pmatrix} \{\langle a_0, a_0\rangle\} & \{\langle a_0, a_1\rangle\} & \{\langle a_0, a_2\rangle\} \\ \{\langle a_1, a_0\rangle\} & \{\langle a_1, a_1\rangle\} & \{\langle a_1, a_2\rangle\} \\ \{\langle a_2, a_0\rangle\} & \{\langle a_2, a_1\rangle\} & \{\langle a_2, a_2\rangle\} \end{pmatrix}.$$

Conversely, if P' is a proto-solution of M, then there is a solution $\langle a_0, a_1, a_2\rangle$ such that $P' = P(a_0, a_1, a_2)$. To see this, first note that the atoms of $\mathfrak{Re}(X)$ are singleton relations of the form $\{\langle a, b\rangle\}$ with $a, b \in X$. Thus the entries of P' are singleton relations, and the ones on the main diagonal of P' have the form $\{\langle a, a\rangle\}$. Suppose $P'_{00} = \{\langle a, a\rangle\}$, $P'_{11} = \{\langle b, b\rangle\}$, and $P'_{22} = \{\langle c, c\rangle\}$. Then, for example, $P'_{01} \subseteq P'_{00} ; P'_{01} ; P'_{11}$ since P' is closed, so $P'_{01} = \{\langle a, b\rangle\}$. Continuing in this way, we conclude that $P' = P(a, b, c)$. This correspondence between solutions and proto-solutions (basic matrices) has been noted elsewhere [39, p79], [40, p965], [41, p111f].

Theorem 5.7 shows that a solution of a constraint matrix over a complete atomic proper relation algebra generates a proto-solution. Theorem 5.9 shows the converse fails in general. However, proto-solvability does entail $R^{\boldsymbol{PA}}$-solvability for the ***PA*** (Theorem 5.10) and $R^{\boldsymbol{IA}}$-solvability for the ***IA*** (which follows from Theorem 5.10 using the correspondence of basic ***IA***-matrices to basic ***PA***-matrices given in [27] or [63]).

Theorem 5.7 *Let $\mathfrak{A}$ be a complete atomic proper relation algebra, and let M be an $\mathfrak{A}$-matrix. Then every solution of M is a solution of a proto-solution of M.*

Proof: Suppose $\langle a_0, \ldots, a_{n-1}\rangle$ is a solution of M. For all $i, j < \mathrm{ord}(M)$, we have $\langle a_i, a_j\rangle \in M_{ij}$, but M_{ij} is the union of the set of atoms below it since $\mathfrak{A}$ is complete, atomic, and proper, so it follows directly that there is an atom $N_{ij} \leq M_{ij}$ such that $\langle a_i, a_j \rangle \in N_{ij}$. We claim N is a proto-solution. We have to show N is path-consistent and symmetric, which we do by contradiction. Path-consistency fails only if for some $i, j, k < \mathrm{ord}(M)$, $N_{ik} \not\subseteq N_{ij} ; N_{jk}$. Since N_{ik} is an atom, it follows that $N_{ik} \cap N_{ij} ; N_{jk} = \emptyset$. But, since $\langle a_0, \ldots, a_{n-1} \rangle$ is a solution of N, we have $\langle a_i, a_k \rangle \in N_{ik}$, $\langle a_i, a_j \rangle \in N_{ij}$, and $\langle a_j, a_k \rangle \in N_{jk}$. These conditions imply $\langle a_i, a_k \rangle \in N_{ik} \cap N_{ij} ; N_{jk} \neq \emptyset$, a contradiction. Similarly, symmetry fails only if there are some $i, j < \mathrm{ord}(M)$ such that $N_{ij} \neq \breve{N}_{ij} = (N_{ji})\breve{}$. Converses of atoms are atoms, so this yields $N_{ij} \cap (N_{ji})\breve{} = \emptyset$. Since $\langle a_0, \ldots, a_{n-1} \rangle$ is a solution of N, we have $\langle a_i, a_j \rangle \in N_{ij}$ and $\langle a_j, a_i \rangle \in N_{ji}$, hence $\langle a_i, a_j \rangle \in N_{ij} \cap (N_{ji})\breve{} \neq \emptyset$, a contradiction. ∎

Minimal labelings. We define the *minimal labeling* of an $\mathfrak{A}$-matrix M to be the $\mathfrak{A}$-matrix L such that for all $i, j < \mathrm{ord}(M)$, $L_{ij} = \sum\{P_{ij} : P \in \mathrm{PS}(\mathfrak{A}, M)\}$. L_{ij} is called the *minimal label* for edge $\langle i, j \rangle$. This definition coincides with the usual definitions, *e.g.*, [49] (in which it is called a *minimal network*), [6], [62], whenever $\mathfrak{A}$ is an atomic proper relation algebra in which every basic matrix is solvable.

Theorem 5.8 *If $\mathfrak{A}$ is a complete atomic proper relation algebra in which every basic matrix is solvable, and M is an $\mathfrak{A}$-matrix, then the minimal labeling L of M is closed, and is also the smallest $\mathfrak{A}$-matrix equivalent to M.*

Proof: Let R be an atom of $\mathfrak{A}$ below L_{ij}. By the definition of minimal labeling, there is some $P \in \mathrm{PS}(\mathfrak{A}, M)$ such that $P_{ij} = R$. Since every proto-solution has a solution, there is a solution $\langle a_0, \ldots, a_{n-1} \rangle$ of P (hence also of M) such that $\langle a_i, a_j \rangle \in R$. For any $\mathfrak{A}$-matrix M' equivalent to M, $\langle a_0, \ldots, a_{n-1} \rangle$ is also a solution of M', by definition. Thus $R \cap M'_{ij} \neq \emptyset$, but R is an atom, so $R \subseteq M'_{ij}$. Therefore any atom below L_{ij} is also below $\bigcap\{M'_{ij} : M' \text{ is equivalent to } M\}$, and hence $L_{ij} \subseteq \bigcap\{M'_{ij} : M' \text{ is equivalent to } M\}$, *i.e.*, L is a reduction of every $\mathfrak{A}$-matrix equivalent to M.

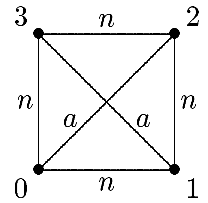
To show that L is equivalent to M, let $\langle a_0, \ldots, a_{n-1} \rangle$ be a solution of M. By Theorem 5.7, $\langle a_0, \ldots, a_{n-1} \rangle$ is a solution of some $P \in \mathrm{PS}(\mathfrak{A}, M)$. By the definition of L, $P \leq L$, so $\langle a_0, \ldots, a_{n-1} \rangle$ is also a solution of L. The converse is immediate.

To show L is path-consistent, assume it is not. Thus, for some $i, j, k < \mathrm{ord}(M)$, $L_{ik} \not\subseteq L_{ij} ; L_{jk}$. Since $\mathfrak{A}$ is atomic, it follows that there is some atom R such that $R \subseteq L_{ik}$ and $R \cap L_{ij} ; L_{jk} = \emptyset$. Let $P \in \mathrm{PS}(\mathfrak{A}, M)$ such that $R = P_{ik}$. Then $P_{ij} \subseteq L_{ij}$ and $P_{jk} \subseteq L_{jk}$ so $P_{ij} ; P_{jk} \leq L_{ij} ; L_{jk}$ and thus $P_{ik} \cap P_{ij} ; P_{jk} = \emptyset$, contradicting the path-consistency of P. A similar argument shows that L is also symmetric. Consequently L is closed.

Thus L is a reduction of all RA matrices equivalent to M, is itself equivalent to M, and is closed. ∎

Theorem 5.9 *Let*

$$M = \begin{pmatrix} 1\text{'} & n & a & n \\ n & 1\text{'} & n & a \\ a & n & 1\text{'} & n \\ n & a & n & 1\text{'} \end{pmatrix} =$$



Then M is a basic $\boldsymbol{CA}$-matrix (a proto-solution) which is $R_o^{\boldsymbol{CA}}$-solvable but not $R^{\boldsymbol{CA}}$-solvable.

Proof: The entries of M are atoms of the $\boldsymbol{CA}$, and M has the identity element of the $\boldsymbol{CA}$ along its main diagonal. It is easy to check that M is closed. Thus M is a basic $\boldsymbol{CA}$-matrix. It follows that $R^{\boldsymbol{CA}}(M)$ and $R_o^{\boldsymbol{CA}}(M)$ are closed constraint matrices. It is easy to find a solution to $R_o^{\boldsymbol{CA}}(M)$, so we only show that $R^{\boldsymbol{CA}}(M)$ has no solution. To do so, we argue by contradiction. Suppose $\langle\langle q_0, r_0\rangle, \langle q_1, r_1\rangle, \langle q_2, r_2\rangle, \langle q_3, r_3\rangle\rangle$ is a solution of $R^{\boldsymbol{CA}}(M)$, where $q_0 < r_0$, $q_1 < r_1$, $q_2 < r_2$, and $q_3 < r_3$. Then $\langle\langle q_0, r_0\rangle, \langle q_2, r_2\rangle\rangle \in R^{\boldsymbol{CA}}(a)$, so without loss of generality assume $q_0 < r_0 < q_2 < r_2$. We have $\langle\langle q_0, r_0\rangle, \langle q_1, r_1\rangle\rangle \in R^{\boldsymbol{CA}}(n)$, hence either $q_0 < q_1 < r_0 < r_1$ or $q_1 < q_0 < r_1 < r_0$, and $\langle\langle q_1, r_1\rangle, \langle q_2, r_2\rangle\rangle \in R^{\boldsymbol{CA}}(n)$, hence either $q_1 < q_2 < r_1 < r_2$ or $q_2 < q_1 < r_2 < r_1$. Of these four apparent possibilities, only one is consistent with $r_0 < q_2$, namely, $q_0 < q_1 < r_0 < q_2 < r_1 < r_2$. Similarly, from $\langle\langle q_0, r_0\rangle, \langle q_3, r_3\rangle\rangle \in R^{\boldsymbol{CA}}(n)$ and $\langle\langle q_3, r_3\rangle, \langle q_2, r_2\rangle\rangle \in R^{\boldsymbol{CA}}(n)$ we conclude that $q_0 < q_3 < r_0 < q_2 < r_3 < r_2$. It follows that $\langle\langle q_1, r_1\rangle, \langle q_3, r_3\rangle\rangle \in R^{\boldsymbol{CA}}(1\text{'} + c + \breve{c} + n)$, contradicting $\langle\langle q_1, r_1\rangle, \langle q_3, r_3\rangle\rangle \in R^{\boldsymbol{CA}}(a)$. ∎

Theorem 5.10 *If M is a closed $\boldsymbol{PA}$-matrix with no 0-labels, then M is $R^{\boldsymbol{PA}}$-solvable.*

Proof: Let M be a $\boldsymbol{PA}$-matrix of order n with $M_{ij} \neq 0$ for all $i, j < n$. We will use standard notation for convex sets of rational numbers, *e.g.*, $(0,1) = \{r : r \in \mathbb{Q},\ 0 < r < 1\}$, except that for each $r \in (0,1)$ we let $\overline{\{r\}} = (0, r) \cup (r, 1)$. We will construct a solution $\langle a_0, \ldots, a_{n-1}\rangle$ inductively. Let a_0 be any element of $(0,1)$. Suppose $0 < i < n$ and $a_0, \ldots, a_{i-1}$ have been chosen so that $\langle a_j, a_k\rangle \in R^{\boldsymbol{PA}}(M_{jk})$ for all $j, k < i$ (in words, $a_0, \ldots, a_{i-1}$ satisfy the constraints placed on them by M) and $M_{jk} = 1\text{'}$ whenever $a_j = a_k$. (The latter proviso is crucial, and will be used near the end of the proof.) We will show how to choose a_i.

For every $j < i$ let

$$I_j = \begin{cases} \{a_j\} & \text{if } M_{ij} = 1\text{'} \\ (0, a_j] & \text{if } M_{ij} = b + 1\text{'} \\ [a_j, 1) & \text{if } M_{ij} = \breve{b} + 1\text{'} \\ (0,1) & \text{if } M_{ij} = b + \breve{b} + 1\text{'} \\ (0, a_j) & \text{if } M_{ij} = b \\ (a_j, 1) & \text{if } M_{ij} = \breve{b} \\ \overline{\{a_j\}} & \text{if } M_{ij} = b + \breve{b} \end{cases}$$

The cases in the definition of I_j are exhaustive because no label in M is 0. The definition of I_j is arranged so that, for every $r \in (0,1)$, $r \in I_j$ if and only if $\langle r, a_j\rangle \in R^{PA}(M_{ij})$ (which is equivalent to $\langle a_j, r\rangle \in R^{PA}(M_{ji})$, by the symmetry of M).

First, we claim that $I_j \cap I_k \neq \emptyset$ whenever $j, k < i$. The are 49 possibilities altogether, given that I_j and I_k can each have seven different values. The cases are indicated in the table below. The possible values for I_k are listed in the second row, with the associated values of M_{ik} above in the first row. The possible values of I_j are listed in the second column, with the associated values of M_{ji} (the converse of M_{ij}) in the first column. Each entry in the table shows the necessary and sufficient condition for $I_j \cap I_k = \emptyset$. Many entries are blank, because in those cases we necessarily have $I_j \cap I_k \neq \emptyset$.

	$M_{ik} \rightarrow$	1'	$b+1$'	$\breve{b}+1$'	1	b	$\breve{b}$	$b+\breve{b}$
$M_{ji} \downarrow$		$\{a_k\}$	$(0,a_k]$	$[a_k,1)$	$(0,1)$	$(0,a_k)$	$(a_k,1)$	$\overline{\{a_k\}}$
1'	$\{a_j\}$	$a_j \neq a_k$	$a_j > a_k$	$a_j < a_k$		$a_j \geq a_k$	$a_j \leq a_k$	$a_j = a_k$
$\breve{b}+1$'	$(0,a_j]$	$a_j < a_k$		$a_j < a_k$			$a_j \leq a_k$	
$b+1$'	$[a_j,1)$	$a_j > a_k$	$a_j > a_k$			$a_j \geq a_k$		
1	$(0,1)$							
$\breve{b}$	$(0,a_j)$	$a_j \leq a_k$		$a_j \leq a_k$			$a_j \leq a_k$	
b	$(a_j,1)$	$a_j \geq a_k$	$a_j \geq a_k$			$a_j \geq a_k$		
$b+\breve{b}$	$\overline{\{a_j\}}$	$a_j = a_k$						

What remains is to show that every condition in the table leads to a contradiction.

Consider the condition $a_j < a_k$, which occurs three times in the table. Since $a_0, \ldots, a_{i-1}$ satisfy the constraints of M, we have $\langle a_j, a_k\rangle \in R^{PA}(M_{jk})$, hence $b \leq M_{jk}$. M is closed, so $M_{jk} \leq M_{ji};M_{ik}$, hence $b \leq M_{ji};M_{ik}$. The entries for M_{ji} and M_{ik} in the three cases are either $\breve{b}+1$' and 1', or 1' and $\breve{b}+1$', or $\breve{b}+1$' and $\breve{b}+1$'. In each of these cases the product $M_{ji};M_{ik}$ is $\breve{b}+1$', which does not contain b, a contradiction.

Next consider the condition $a_j \leq a_k$, which occurs five times in the table. Since $\langle a_j, a_k\rangle \in R^{PA}(M_{jk})$, we have either 1' $\leq M_{jk}$ or $b \leq M_{jk}$, *i.e.*, $0 \neq (b+1') \cdot M_{jk}$. From $M_{jk} \leq M_{ji};M_{ik}$ we derive $0 \neq (b+1') \cdot M_{ji};M_{ik}$. But the entries for M_{ji} and M_{ik} in the five cases are either $\breve{b}$ and 1', or $\breve{b}$ and $\breve{b}+1$', or $\breve{b}$ and $\breve{b}$, or $\breve{b}+1$' and $\breve{b}$, or 1' and $\breve{b}$. In each of these cases the product $M_{ji};M_{ik}$ is $\breve{b}$, which is disjoint from $b+1$', a contradiction.

We leave it to the reader to derive a contradiction for the remaining entries in the table.

Consider the sets $I_0, \ldots, I_{i-1}$. Some of them are convex sets, while all the others have the form $\overline{\{r\}}$. Any two of the convex sets intersect, as we have seen, so the intersection of all of them must be a nonempty convex set, by an easy-to-prove special case of Helly's Theorem [11], [15]. If this intersection has two elements, then it has infinitely many because it is convex, so that when it is intersected with the finitely many sets $\overline{\{a_0\}}, \ldots, \overline{\{a_{i-1}\}}$, the result still contains infinitely many elements, and we let a_i be any one of them.

Suppose that the intersection of the convex sets in $I_0, \ldots, I_{i-1}$ contains exactly one element. This can occur in two ways. First, suppose one of the convex sets among $I_0, \ldots, I_{i-1}$ has the form $\{r\}$, say $I_j = \{a_j\}$ for some $j < i$. Then $M_{ij} = 1'$ by the definition of I_j. No two sets among $I_0, \ldots, I_{i-1}$ are disjoint, so $\overline{\{a_j\}}$ does not occur among $I_0, \ldots, I_{i-1}$. Thus $\{a_j\} = \bigcap_{j=0}^{i-1} I_j$, and we must set $a_i = a_j$. The constraints on $a_0, \ldots, a_i$ are satisfied. The special proviso is also satisfied, for if $k < i$ and $a_i = a_k$, then $a_j = a_i = a_k$, hence $M_{jk} = 1'$ (by the proviso applied to $a_0, \ldots, a_{i-1}$), so $M_{ik} \leq M_{ij} ; M_{jk} = 1' ; 1' = 1'$, but $1'$ is an atom and $M_{ik} \neq 0$, so $M_{ik} = 1'$.

So we may assume that there is no singleton among $I_0, \ldots, I_{i-1}$, which brings us to the second way that the intersection of the convex sets in $I_0, \ldots, I_{i-1}$ can contain exactly one element, namely, that $(0, r]$ and $[r, 1)$ are among $I_0, \ldots, I_{i-1}$, for some $r \in \{a_0, \ldots, a_{i-1}\}$. Suppose $(0, r] = I_j$ and $[r, 1) = I_k$, for some $j, k < i$. It follows that $M_{ij} = b + 1'$, $M_{ik} = \breve{b} + 1'$, and $a_j = a_k$. By the special proviso, we also have $M_{jk} = 1'$. But then $\breve{b} + 1' = M_{ik} \leq M_{ij} ; M_{jk} = (b + 1') ; 1' = b + 1'$, a contradiction. ∎

Theorem 5.10 was inspired by Theorem 4 of [65]. The following ***PA***-matrix was presented in [62] as a counterexample to Theorem 4 of [65].

$$M = \begin{pmatrix} 1' & \breve{b} + 1' & \breve{b} + 1' & \breve{b} + 1' \\ b + 1' & 1' & b + 1' & b + 1' \\ b + 1' & \breve{b} + 1' & 1' & b + \breve{b} \\ b + 1' & \breve{b} + 1' & b + \breve{b} & 1' \end{pmatrix}.$$

It is straightforward to check that M is closed. M shows the need for the special proviso in the proof of Theorem 5.10, for if a_0, a_1, a_2 are chosen to be equal (and thus violate the special proviso) then there is no a_3 such that $\langle a_0, a_1, a_2, a_3 \rangle$ is a solution of M.

Some asymptotic complexity observations Assume that a single element composition counts as one operation, as does a single element intersection. For a finite relation algebra, this assumption is justified for asymptotic calculations, since in principle the entire (finite) composition table can be stored. For example, the composition table for the ***IA*** is of size $2^{13} \cdot 2^{13}$ (if entries for 1' are omitted, one has $(2 \cdot 2^{13} - 1)$ fewer entries). In practice, code for iterative local algorithms on finite relation algebras may store the composition table for atoms, but calculate compositions of non-atoms dynamically [55]. For the ***IA*** the atom composition table has 13^2 entries (again one may omit entries for 1'and use a table of size 12^2), so storage requirements are significantly smaller. There is also scope for clever trade-offs between store and compute.

The assumption of constant time for composition may be inappropriate in general for infinite algebras. In this case, composition may be an asymptotically non-constant operation.

Consider the $\boldsymbol{IA}$. Obtaining an $R^{\boldsymbol{IA}}$-solution from a basic $\boldsymbol{IA}$-matrix is relatively easy.

Corollary 5.11 *All basic $\boldsymbol{IA}$-matrices are $R^{\boldsymbol{IA}}$-solvable, in time quadratic in the order of the matrix.*

Proof: Every basic $\boldsymbol{IA}$-matrix M corresponds to a basic $\boldsymbol{PA}$-matrix of order $2 \cdot \mathrm{ord}(M)$ [27] [63]. Basic $\boldsymbol{PA}$-matrices are $R^{\boldsymbol{PA}}$-solvable by Theorem 5.10, and $R^{\boldsymbol{PA}}$-solutions for $R^{\boldsymbol{PA}}$-solvable basic $\boldsymbol{PA}$-matrices may be found in quadratic time [64]. ∎

Corollary 5.12 *Obtaining a proto-solution of a closed $\boldsymbol{IA}$-matrix is NP-complete in the order.*

Proof: $R^{\boldsymbol{IA}}$-solvability of $\boldsymbol{IA}$-matrices is NP-complete in the order [66]. Closure is cubic time, and by Corollary 5.11, solutions can be obtained in quadratic time from proto-solutions. It follows that it is NP-complete to obtain a proto-solution of a closed $\boldsymbol{IA}$-matrix. ∎

The complexity of detecting proto-solvability, and of constructing a minimal labeling may differ. Detecting solvability and constructing a solution for $\boldsymbol{PA}$-matrices is $O(n^2)$ [63], but calculating a minimal labeling is only known to be $O(n^4)$ [62]. Finding a minimal labeling for a constraint matrix over the $\boldsymbol{IA}$ is in NP, roughly because one may guess the labelling ($O(n^2)$), then for each atom in a label (constant number of atoms, so also $O(n^2)$), guess a protosolution involving that label ($O(n^2)$), and finally check that the protosolution is a solution ($O(n^3)$). The existence of a minimal labelling implies that there is a solution, and solvability is NP-complete [66], so finding a minimal labelling is also NP-hard. Thus finding minimal labellings for constraint matrices over the $\boldsymbol{IA}$ is NP-complete.

6 A closure algorithm scheme

The closure of a constraint matrix M is a necessary condition for the existence of solutions of M, but it is not sufficient [1]. For algebras of relations on a finite base, closure algorithms run in serial cubic time, and algorithms in the literature must perform cubically many operations on any constraint problem [35], [47]. Allen adapted such a closure algorithm to run on the $\boldsymbol{IA}$, which is representable over an infinite base. We characterize these algorithms as *iterative local* algorithms[2], and provide a scheme, `PCon`, defining them. `PCon` computes on RA matrices, and its behavior may be considered on infinite algebras, as well as finite ones. We will note that the closure of an $\mathfrak{A}$-matrix can be computed in parallel $O(n^2 \log n)$ time on finite algebras. We also note experimental evidence that, on an algebra of relations on a finite base, practical problems harder than

[2] Our thanks to a referee for suggesting this terminology.

parallel linear time are uncommon [57], [58]. However, we shall exhibit $\boldsymbol{HA}$-matrices in Section 7 on which iterative local algorithms, whether parallel or serial, must take $\Omega(n^2)$ time.

Let M be an n-by-n constraint matrix over $\mathfrak{A}$. If $\mathfrak{A}$ is complete and atomic, we have shown, under the condition that all proto-solutions of M are solvable, that among constraint matrices equivalent to M there is a least one, which is also closed (Theorem 5.8). If $\mathfrak{A}$ is an algebra of relations on a finite base, it is known that among constraint matrices equivalent to M which are reductions of M, there is also a greatest closed one [49, Theorem 4.1]. This greatest fixed point is the collection of largest possible relations R_{ij} which satisfy the following equations for all $i, j < n$:

$$\begin{aligned}
&R_{ij} \subseteq M_{ij},\\
&R_{ii} \subseteq Id,\\
&R_{ij} = (R_{ji})\breve{},\\
&R_{ij} \subseteq R_{i0}\,;R_{0j} \cap \cdots \cap R_{i,n-1}\,;R_{n-1,j}.
\end{aligned}$$

This theorem can be generalized to RA matrices. Define a *most general closed reduction* of M to be a closed reduction N of M such that all closed reductions of M are reductions of N. If there is such a reduction, it follows that it is unique, and we denote it by $MGR(M)$. We consider the existence of $MGR(M)$, by considering the termination of a certain algorithm scheme, `PCon`.

The algorithm scheme `PCon` is shown in Figure 1. Given a constraint matrix M, `PCon` computes a reduction M' by side-effect. It is easy to see that any solution of M is also a solution of $M \cdot \breve{M} \cdot M^2$, and the converse is trivial. Thus at any step in the execution of I, the current matrix is equivalent to M. The final matrix M' is therefore a constraint matrix equivalent to M. The scheme `PCon` may be regarded as a specification of a class of closure algorithms. We define an *iterative local* algorithm to be an algorithm that performs a computation described by `PCon`. An iterative local algorithm will also be called an *implementation* of `PCon`. We use the term *coding* to refer to program code that executes an algorithm.

All closure algorithms in the literature (*e.g.*, [1], [35], [47], [49], [58]) are iterative local algorithms, differing according to how the computations for each individual triangle are ordered, stored, and reused. We first show that any iterative local algorithm computes $MGR(M)$ whenever it terminates, then we consider termination of iterative local algorithms on RA matrices over various relation algebras.

Theorem 6.1 *If M' is the final RA matrix of any implementation of* `PCon` *on M that terminates, then $MGR(M)$ exists, and $M' = MGR(M)$.*

Proof. Let M be an RA matrix. Choose some fixed implementation I of `PCon`. Execute I on M, and if I terminates, let M' be the final RA matrix. The formulation of `PCon` shows I terminates only when the current matrix is closed, so M' must be closed.

Let N be a closed reduction of M. From $N \leq M$, it follows that $\breve{N} \leq \breve{M}$ and $N^2 \leq M^2$. But $N = \breve{N} = N^2$ since N is closed, so $N \leq M \cdot \breve{M} \cdot M^2$. Thus after one

```
repeat
    find i, j, k such that M_ij ≰ M̆_ij · M_ik ; M_kj
    M_ij := M_ij · M̆_ij · M_ik ; M_kj
until M = M̆ = M^2
```

Figure 1: The PCon algorithm scheme

iteration of PCon, N is still smaller than the current matrix. It follows by induction that $N \leq M'$. Thus M' is a closed reduction of M that has as reductions all closed reductions of M, so by definition $M' = MGR(M)$. ∎

Theorem 6.2 *All iterative local implementations of* PCon *terminate within $d \cdot n^2$ iterations of the loop on every n-by-n constraint matrix over a finite relation algebra $\mathfrak{A}$ with d atoms.*

Proof: The assignment statement in PCon can only leave a label unchanged or replace it with a smaller one. Each of the n^2 labels in M can change at most as many times as there are atoms of $\mathfrak{A}$, namely d times. Therefore the `repeat` loop can be executed at most $d \cdot n^2$ times. By then a state will be reached in which there is no triangle $\langle i, j, k\rangle$ with $M_{ij} \not\leq \breve{M}_{ij} \cdot M_{ik} ; M_{kj}$, that is, no violations of closure can be found, so the current matrix is closed. ∎

The next corollary follows from Theorems 5.1, 5.4, and 5.6 and the termination condition of PCon.

Corollary 6.3 *For every relation algebra $\mathfrak{A}$, every implementation of* PCon *terminates on every $\mathfrak{A}$-matrix M of order 3, because $((M \cdot \breve{M})^2)^2$ is closed.*

On the other hand, Theorem 6.4 shows that there are RA matrices of order 4 (the smallest possible), on which no implementation of PCon terminates.

Theorem 6.4 *There is a constraint matrix M of order 4 such that no iterative local algorithm terminates on M.*

Proof: Let U be a countable infinite set, arranged and labeled with 0,1,2, and 3 as shown in Figure 2. For all $i, j < 4$ let M_{ij} be the set of all pairs $\langle u, v\rangle$ of elements of U such that, in Figure 2, the label of u is i, the label of v is j, and either $u = v$ or else there is a horizontal, vertical, or diagonal line segment connecting u with v. Thus, for example, M_{00} consists of all those pairs $\langle u, u\rangle$ such that $u \in U$ and the label on u is 0. Elements of U connected by a line segment will be called *adjacent.* Hence M_{01} consists of those pairs $\langle u, v\rangle$ such that u and v are adjacent, u is labeled 0, and v is labeled 1.

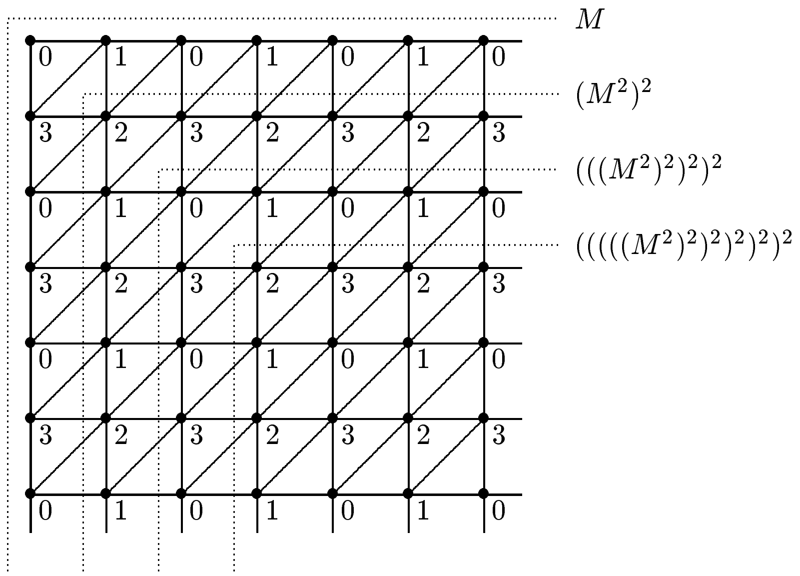


Figure 2: Relations in a 4-by-4 nonterminating matrix

It is straightforward to verify that successive squarings of M always produce smaller matrices, *i.e.*,

$$M > M^2 > (M^2)^2 > ((M^2)^2)^2 > (((M^2)^2)^2)^2 > ((((M^2)^2)^2)^2)^2 > \dots$$

To explain how this works, we give an intuitive idea of what happens during iterated squaring. All the adjacent pairs in the first row of Figure 2 belong to M_{01} or M_{10}. All the ones in M_{01} are excluded from either $M_{03}\,;M_{31}$ or $M_{02}\,;M_{21}$. But $(M^2)_{01} \subseteq M_{03}\,;M_{31} \cdot M_{02}\,;M_{21}$, so all such pairs are also excluded from $(M^2)_{01}$. Similarly, the adjacent pairs in the first row which belong to M_{10} do not belong to $(M^2)_{10}$. Thus none of the adjacent pairs in the first row survive into M^2. For similar reasons, the same is true for the adjacent pairs in the first column. All other adjacent pairs appear in M^2. Pairs of the form $\langle u, u\rangle$ also survive into M^2, except when u is the element in the upper left-hand corner of Figure 2. All pairs containing an element of U in the first row or first column disappear at the next squaring. Thus a diagram for $(M^2)^2$ can be obtained from Figure 2 by deleting all the elements in the first row and the first column. The restrictions which yield diagrams for $(M^2)^2$, $(((M^2)^2)^2)^2$, and $(((((M^2)^2)^2)^2)^2)^2$ are indicated in Figure 2. Thus as squaring proceeds, the array is gradually "eaten away" from the top. Since it is infinite below and to the right, there is no finite stage in this process at which any M_{ij} becomes empty, although every pair is eventually eliminated. ∎

$$
\begin{array}{l}
M := M \cdot \breve{M} \\
\texttt{repeat} \\
\qquad M := M^2 \\
\texttt{until } M = M^2
\end{array}
$$

Figure 3: The implementation `CMat` of `PCon`

A path-consistency algorithm. The condition $M = \breve{M} = M^2$ is equivalent to the closure of the constraint matrix M, and squares of symmetric constraint matrices are symmetric by Theorem 5.4. This enables an elegant definition of an implementation `CMat` of `PCon`, in Figure 3, which constructs $MGR(M)$, again by side-effect.

Practical use of `CMat`. `CMat` has been coded and used in an effective interval constraint solver, which finds a single solution (in our terminology, $R^{\boldsymbol{IA}}$-solution) to a random[3] interval constraint problem (*i.e.*, $\boldsymbol{IA}$-matrix) on the average in less than a second on a RISC workstation [28], and is also effective in finding solutions to solvable interval constraint problems of up to 200 variables (i.e. order 200) [55]. The coding of `CMat` embodies improvements over the algorithm as stated in Figure 3, including obvious ones such as only using and storing half of the matrix, and some non-obvious, validated by use and testing. Details are in the references.

Computing composition for algebras on finite bases For algebras of relations on (small) finite bases, it is practical to represent relations as adjacency matrices whose rows and columns are labeled by elements of the base, and compute composition as a Boolean-arithmetic operation on matrices. Boolean-arithmetic operations on 0 and 1 yield the same results as the usual arithmetic operations, except for $1 + 1 = 1$. Boolean product of 0's and 1's is the same as in usual arithmetic.

Let U be the base, and suppose that U contains s elements. The rows and columns of an adjacency matrix M_R of a relation R are labeled with the elements of U. M_R is a 0-1-valued matrix, with $(M_R)_{ab} = 1$ if and only if $\langle a, b \rangle \in R$. Let $\times$ be the Boolean-arithmetic multiplication of 0-1-matrices.

It is straightforward to see that the composition operation on finite relations may be formulated as multiplication of adjacency matrices, *i.e.*, for binary relations R and S, $M_{R;S} = M_R \times M_S$. This formulation of composition appears explicitly already in [52, p188] and [56, §4], and is used in [49] and [35].

Time complexity of path-consistency on finite algebras Let $\mathfrak{A}$ be a finite relation algebra. We may assume as before that a single element composition is one operation, as is a single element intersection. It is easy to construct a MIMD processor

[3]Each $\boldsymbol{IA}$ relation has equal probability of occurring as the constraint between a pair of variables.

array that squares an $\mathfrak{A}$-matrix in $O(\log n)$ time, where $n = \mathrm{ord}(M)$, with one relation-arithmetic operation per processor. This is not a practical scheme, since it requires $O(n^3)$ processors, but it will suffice to obtain an upper bound estimate for parallel iterative local algorithms that is close to the lower bound result in Section 7.

Pairwise element compositions will be performed in parallel, each product $M_{ij} ; M_{jk}$ computed on a different processor P_{ijk}. In the first step, each M_{ij} is broadcast to the $2n$ processors that use it, namely P_{ijk} and P_{kij} for $k < n$. In the second step, all the P_{ijk} compute their results. In succeeding steps, results are cumulatively intersected two at a time in parallel by $\lceil \log n \rceil$ consecutive rows of processors. Thus squaring takes $O(\log n)$ time.

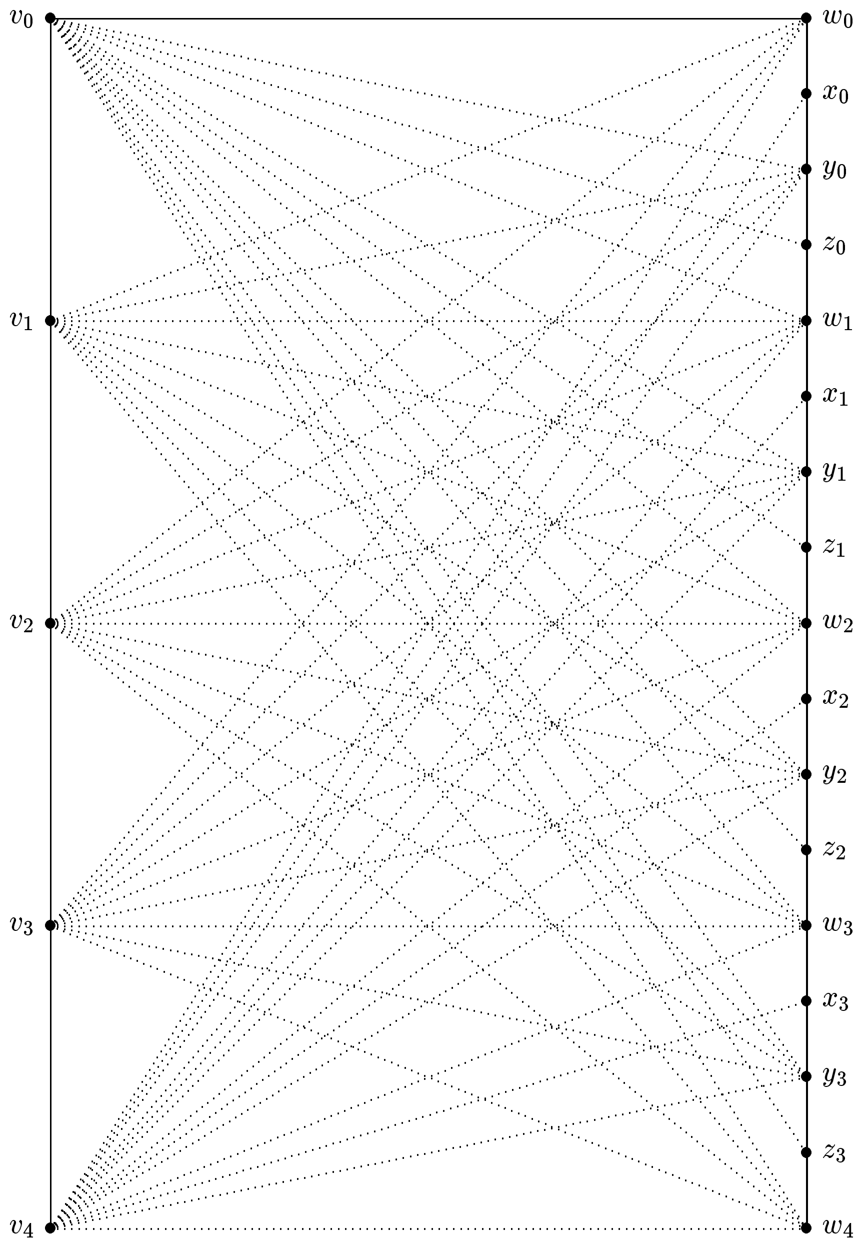
Let $\mathfrak{A}$ have d atoms. By Theorem 6.2, the loop in `CMat` iterates at most $d \cdot n^2$ times. Putting this estimate together with the estimate for squaring, we obtain a parallel upper bound of $O(n^2 \log n)$ time for the computation of $MGR(M)$ over finite algebras.

Path-consistency computation experiments on parallel machines such as Crays, Butterflies and Connection Machines have been reported in [57], [58], for relations on a finite set. They obtain roughly linear computation times (in the order) on their sample problem set. By contrast, we construct a class of $\Theta(n^2)$ examples below, where n is the order.

7 A Class of Hard Examples for Path-Consistency

For each $n \geq 2$ we define a symmetric $\boldsymbol{HA}$-matrix M_n of order $5n + 2$, and show that the computation of $MGR(M_n)$ by any iterative local algorithm takes $\Theta(n^2)$ time. This result, along with the upper bound for `CMat`, bounds the asymptotic worst-case parallel complexity of iterative local algorithms for MGR within a range of "width" $\log(n)$. Our construction of M_n is such that, in any state before termination, there is just one unstable triangle in the M_n, and by stabilizing this triangle, a new unstable triangle will be created. The most efficient iterative local algorithm will visit the "right" (unstable) triangle first, then the new unstable triangle next, and so on, and will need $\Theta(n^2)$ triangle operations to obtain $MGR(M_n)$. Since each unstable triangle is created by stabilizing the previous one, any iterative local algorithm operates essentially serially on M_n, and both parallel and serial algorithms of this type take the same time to stabilize the entire network.

We define M_n by describing its network. $\boldsymbol{HA}$ is a symmetric algebra so we may use an undirected graph, since the labels read correctly in either direction. The $5n + 2$ nodes are divided into five groups: $v_0, \ldots, v_n$, $w_0, \ldots, w_n$, $x_0, \ldots, x_{n-1}$, $y_0, \ldots, y_{n-1}$, and $z_0, \ldots, z_{n-1}$. For any nodes s, t we denote the label on the edge between s and t by



All the solid edges are labeled with a.
All the dotted edges are labeled with $a + c$.
All other edges are labeled with $a + b$.

Figure 4: The Hard Algebra matrix M_4

$M_n(s,t)$. For every node s, let $M_n(s,s) = 1'$. For every $i < n$ and every $j < n+1$ let

$$\begin{aligned} M_n(v_i, v_{i+1}) &= a, \\ M_n(w_i, x_i) &= a, \\ M_n(x_i, y_i) &= a, \\ M_n(y_i, z_i) &= a, \\ M_n(z_i, w_{i+1}) &= a, \\ M_n(w_0, v_0) &= a, \\ M_n(w_i, v_j) &= a + c \quad \text{if } i \neq 0 \text{ or } j \neq 0, \\ M_n(y_i, v_j) &= a + c, \\ M_n(x_i, v_n) &= a + c, \\ M_n(z_i, v_0) &= a + c. \end{aligned}$$

For all other distinct nodes s, t, let $M_n(s,t) = a + b$. The network for M_4 is shown in Figure 4. The edges drawn with solid lines are labeled with a. The dotted edges are labeled $a + c$. Edges from a node to itself are labeled with 1' and are not shown. All other edges are labeled with $a + b$, and are also not shown.

Every triangle in M_n is stable with one exception, namely $\langle v_1, v_0, w_0 \rangle$, which has two edges labeled a, and one edge labeled $a + c$. It follows that M_n^2 agrees with M_n except along edge $\langle v_1, w_0 \rangle$, which is now labeled a. M_n^2 has exactly one unstable triangle, namely $\langle v_2, v_1, w_0 \rangle$, which is isomorphic to the original unstable triangle $\langle v_1, v_0, w_0 \rangle$ in M_n. Hence every iteration of `CMat` stabilizes a single unstable triangle in the network and by doing so (except for the last iteration) creates a new unstable neighboring triangle. The unstable triangles propagate through the network. On the first iteration, the label on $\langle v_1, w_0 \rangle$ changes from $a + c$ to a; on the next, the label on $\langle v_2, w_0 \rangle$ changes from $a + c$ to a, and so on. Furthermore, since each iteration except the last introduces a new unstable triangle, the most efficient parallel iterative local algorithm is only as efficient as the most efficient serial one. They both take time $\Theta(n^2)$. In Figure 4 there are 52 edges whose labels change from $a + c$ to a in the following order:

1.	$\langle v_1, w_0 \rangle$	14.	$\langle v_2, w_1 \rangle$	27.	$\langle v_3, w_2 \rangle$	40.	$\langle v_4, w_3 \rangle$
2.	$\langle v_2, w_0 \rangle$	15.	$\langle v_3, w_1 \rangle$	28.	$\langle v_4, w_2 \rangle$	41.	$\langle v_4, x_3 \rangle$
3.	$\langle v_3, w_0 \rangle$	16.	$\langle v_4, w_1 \rangle$	29.	$\langle v_4, x_2 \rangle$	42.	$\langle v_4, y_3 \rangle$
4.	$\langle v_4, w_0 \rangle$	17.	$\langle v_4, x_1 \rangle$	30.	$\langle v_4, y_2 \rangle$	43.	$\langle v_3, y_3 \rangle$
5.	$\langle v_4, x_0 \rangle$	18.	$\langle v_4, y_1 \rangle$	31.	$\langle v_3, y_2 \rangle$	44.	$\langle v_2, y_3 \rangle$
6.	$\langle v_4, y_0 \rangle$	19.	$\langle v_3, y_1 \rangle$	32.	$\langle v_2, y_2 \rangle$	45.	$\langle v_1, y_3 \rangle$
7.	$\langle v_3, y_0 \rangle$	20.	$\langle v_2, y_1 \rangle$	33.	$\langle v_1, y_2 \rangle$	46.	$\langle v_0, y_3 \rangle$
8.	$\langle v_2, y_0 \rangle$	21.	$\langle v_1, y_1 \rangle$	34.	$\langle v_0, y_2 \rangle$	47.	$\langle v_0, z_3 \rangle$
9.	$\langle v_1, y_0 \rangle$	22.	$\langle v_0, y_1 \rangle$	35.	$\langle v_0, z_2 \rangle$	48.	$\langle v_0, w_4 \rangle$
10.	$\langle v_0, y_0 \rangle$	23.	$\langle v_0, z_1 \rangle$	36.	$\langle v_0, w_3 \rangle$	49.	$\langle v_1, w_4 \rangle$
11.	$\langle v_0, z_0 \rangle$	24.	$\langle v_0, w_2 \rangle$	37.	$\langle v_1, w_3 \rangle$	50.	$\langle v_2, w_4 \rangle$
12.	$\langle v_0, w_1 \rangle$	25.	$\langle v_1, w_2 \rangle$	38.	$\langle v_2, w_3 \rangle$	51.	$\langle v_3, w_4 \rangle$
13.	$\langle v_1, w_1 \rangle$	26.	$\langle v_2, w_2 \rangle$	39.	$\langle v_3, w_3 \rangle$	52.	$\langle v_4, w_4 \rangle$

In the general case, the number of edges whose labels change from $a+c$ to a is $2n^2+5n$. Summarizing, we have

Theorem 7.1 *There is a class of binary constraint problems for which path-consistency is $\Theta(n^2)$ for iterative local algorithms, whether they execute operations in parallel or serially. In particular, for every integer $n \geq 1$, M_n is an* $\boldsymbol{HA}$*-matrix of order $5n+2$ such that* `CMat` *iterates $2n^2+5n+1$ times before terminating, i.e.,*

$$M_n > (M_n \overbrace{)^2}^{1} > ((M_n \overbrace{)^2)^2}^{2} > \ldots > (\cdots(M_n \overbrace{)^2)^2\cdots)^2}^{2n^2+5n} = (\cdots(M_n \overbrace{)^2)^2\cdots)^2}^{2n^2+5n+1}.$$

References

[1] Allen, J.F., *Maintaining Knowledge about Temporal Intervals*, Comm. A.C.M. 26 (11), November 1983, pp832–843.

[2] Allen J.F. and Hayes, P. J., *Moments and Points in an Interval-Based Temporal Logic*, Technical Report TR180, Dept. of Computer Science, University of Rochester, December 1987.

[3] Burris, S., and Sankappanavar, H.P., *A Course in Universal Algebra*, Springer Verlag, 1981.

[4] Chang, C.C., and Keisler, H.J., *Model Theory*, North-Holland, 1973.

[5] Chin, L.H., and Tarski, A., *Distributive and Modular Laws in the Arithmetic of Relation Algebras*, Univ. Calif. Publications in Mathematics, New Series, 1 (9), 1951, pp341–384.

[6] Dechter, R., Meiri, I., and Pearl, J., *Temporal Constraint Networks.* Artificial Intelligence 49, 1991, pp61–95.

[7] Dechter, R., and Pearl, J., *Network-Based Heuristics for Constraint-Satisfaction Problems*, Artificial Intelligence 34, 1988, pp1–38.

[8] De Morgan, A., *On the Syllogism, no. IV, and on the Logic of Relations*, Trans. Cambridge Philos. Soc. 10, 1864, pp331–358. (Written 1859, presented to C.P.S. 4/23/1860, published as above, reprinted in [9]).

[9] De Morgan, A., *On the Syllogism and other Logical Writings*, Yale University Press, New Haven, 1966.

[10] Eilenberg, S., and Mac Lane, S., *Relations between Homology and Homotopy Groups of Spaces*, Annals of Mathematics 46, 1945, pp480–509.

[11] Eggleston, H.G., *Convexity*, Cambridge University Press, 1969.

[12] Freuder, E.C., *Synthesizing Constraint Expressions*, Communications of the ACM 21 (11), Nov 1978, pp958–966.

[13] Golumbic, M.C., and Shamir, R., *Complexity and Algorithms for Reasoning about Time: A Graph-Theoretic Approach*, RUTCOR Research Report RRR # 22–91, Rutgers Center for Operations Research, New Brunswick, NJ, 1991.

[14] Güsgen, H.W., *CONSAT: A System for Constraint Satisfaction*, Pitman and Morgan Kaufmann, 1989.

[15] Helly, E., *Über Mengen convexer Körper mit gemeinschaftlichen Punkten*, Jahresberichte der Deutsch. Math. Verein. 32, 1923, pp175–176.

[16] Imielinski, T., and Lipski, W., Jr., *The Relational Model of Data and Cylindric Algebras*, Jour. of Computer and System Sciences 28, 1984, pp80–102.

[17] Jónsson, B., *Varieties of Relation Algebras*, Algebra Universalis 15, 1982, pp273–298.

[18] Jónsson, B., *The Theory of Binary Relations*, in *Algebraic Logic*, Colloquia Mathematica Societis János Bolyai 54, North-Holland, Amsterdam, 1991, pp245–292.

[19] Jónsson, B. and Tarski, A., *Boolean Algebras with Operators II*, American J. Mathematics 74, 1952, pp127–162.

[20] Kautz, H.A. and Ladkin, P.B., *Integrating Metric and Qualitative Temporal Reasoning*, Proceedings of AAAI-91, the 9th National Conference on AI, AAAI Press 1991, pp241–246.

[21] Ladkin, P.B., *Time Representation: A Taxonomy of Interval Relations*, Proceedings of the 5th National Conference on AI (AAAI-86), 1986, pp360-366.

[22] Ladkin, P.B., *Models of Axioms for Time Intervals*, Proceedings of AAAI-87, the Sixth National Conference on Artificial Intelligence, Morgan Kaufmann 1987, pp234–239; Kestrel Institute Technical Report KES.U.87.4. (long version), Palo Alto CA, 1987.

[23] Ladkin, P.B., *The Logic of Time Representation*, dissertation, University of California, Berkeley, 1987, pp130; Kestrel Institute Technical Report KES.U.87.13, Palo Alto CA, 1987.

[24] Ladkin, P.B., *Satisfying First-Order Constraints about Time Intervals*, Proceedings of the 7th National Conference on AI, (AAAI-88), Morgan Kaufmann 1988, pp512–517; *Constraint Satisfaction in Time Interval Structures I: Convex Intervals*, Kestrel Institute Technical Report KES.U.87.11 (long version), Palo Alto CA 1987.

[25] Ladkin, P.B., *Constraint Reasoning With Intervals: A Tutorial, Survey, and Bibliography*, Technical Report TR-90-059, International Computer Science Institute, Berkeley CA, Sept. 1990.

[26] Ladkin, P.B. and Maddux, R.D., *The Algebra of Convex Time Intervals*, Kestrel Institute Technical Report KES.U.87.2, 1987 (short version); *Representation and Reasoning with Convex Time Intervals*, Kestrel Institute Technical Report KES.U.88.2, Palo Alto CA, 1988 (long version).

[27] Ladkin, P.B., and Maddux, R.D., *On Binary Constraint Networks*, Kestrel Institute Technical Report KES.U.88.8, Palo Alto CA, 1988.

[28] Ladkin, P.B., and Reinefeld, A., *How to Solve Interval Constraint Networks: The Definitive Answer – Probably*, Technical Report TR-91-063, International Computer Science Institute, Berkeley CA, Nov. 1991.

[29] Ligozat, G., and Bestougeff, H., *On Relations Between Intervals*, Information Processing Letters 32, 1989, pp177–182.

[30] Ligozat, G., *Weak Representations of Interval Algebras*, Proceedings of AAAI-90, the Eight National Conference on Artificial Intelligence, AAAI Press 1990, pp715–720.

[31] Ligozat, G., *Intervalles généralisés I, II* Comptes Rendues de l'Academie des Sciences de Paris, t. 310, Série I, 1990, pp225–228 and pp299–302.

[32] Ligozat, G., *On Generalized Interval Calculi*, Proceedings of AAAI-91, the Ninth National Conference on Artificial Intelligence, AAAI Press 1991, pp234–240.

[33] Lyndon, R.C., *The Representation of Relational Algebras*, Annals of Math. Series 2, 51, 1950, pp707–729.

[34] Lyndon, R.C., *The Representation of Relational Algebras, II*, Annals of Math. Series 2, 63, 1956, pp294–307.

[35] Mackworth, A.K., *Consistency in Networks of Relations*, Artificial Intelligence 8, 1977, pp99–118.

[36] Mackworth, A.K., *Constraint Satisfaction*, in the *Encyclopedia of Artificial Intelligence*, ed. S. Shapiro, Wiley Interscience 1987.

[37] Mackworth, A.K., and Freuder, E.C., *The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems*, Artificial Intelligence 25, pp65–74, 1985.

[38] Maddux, R.D., *Some Varieties Containing Relation Algebras*, Transactions of the American Mathematical Society 272, 1982, pp501–526.

[39] Maddux, R.D., *A Sequent Calculus For Relation Algebras*, Annals of Pure and Applied Logic 25, 1983, pp73–101.

[40] Maddux, R.D., *Nonfinite Axiomatizability Results for Cylindric and Relation Algebras*, The Journal of Symbolic Logic 54, 1989, pp951–974.

[41] Maddux, R.D., *Pair-dense Relation Algebras*, Transactions of the American Mathematical Society 328, 1991, pp83–131.

[42] Maddux, R.D., *Introductory course on relation algebras, finite-dimensional cylindric algebras, and their interconnections*, in *Algebraic Logic*, Colloquia Mathematica Societis János Bolyai 54, North-Holland, Amsterdam, 1991, pp361–392.

[43] Maddux, R.D., *The Neat Embedding Problem and the Number of Variables Required in Proofs*, Proc. Am. Math. Soc. 112, 1991, pp195–202.

[44] Maddux, R.D., *The Origin of Relation Algebras in the Development and Axiomatization of the Calculus of Relations*, to appear, Studia Logica.

[45] Meiri, I., *Combining Qualitative and Quantitative Constraints in Temporal Reasoning*, Proceedings of AAAI-91, the 9th National Conference on AI, AAAI Press 1991, pp260–267.

[46] Meiri, I., *Temporal Reasoning: A Constraint-Based Approach*, Ph.D. Thesis, Technical Report R-173, Cognitive Systems Laboratory, UCLA, 1992.

[47] Mohr, R., and Henderson, T.C., *Arc and Path Consistency Revisited*, Artificial Intelligence 28, 1986, pp225–233.

[48] Monk, J. D., *On Representable Relation Algebras*, Michigan Math. J. 11, 1964, pp207–210.

[49] Montanari, U., *Networks of Constraints: Fundamental Properties and Applications to Picture Processing*, Inform. Sci. 7, 1974, pp95–132.

[50] Nökel, K., *Temporally Distributed Symptoms in Technical Diagnosis*, Lecture Notes in Artificial Intelligence 517, Springer Verlag 1991.

[51] Peirce, C. S., *Description of a notation for the logic of relatives, resulting from an amplification of the conceptions of Boole's calculus of logic*, Memoirs of the American Academy of Sciences 9, 1870, pp317–378. Reprint by Welch, Bigelow and Co., Cambridge, Mass., 1870, pp1–62. Also reprinted in [53].

[52] Peirce, C. S., *Note B: the logic of relatives*, in *Studies in Logic by Members of the Johns Hopkins University*, edited by C. S. Peirce, Little, Brown, and Co., Boston, 1883, pp187–203. Reprinted in [53] and in [54].

[53] Peirce, C.S., *Collected Papers, Volume III*, ed. Charles Hartshorne and Paul Weiss, Harvard University Press, Cambridge, 1933.

[54] Peirce, C. S., *Studies in Logic by Members of the Johns Hopkins University*, edited by C. S. Peirce, Introduction by M. H. Fisch, Preface by A. Eschbach, John Benjamins Publishing Co., Amsterdam and Philadelphia, 1983.

[55] Reinefeld, A., and Ladkin, P.B., *Fast Solution of Large Interval Constraint Networks*, J. Glasgow and R. Hedley, eds., Proceedings of the Ninth Canadian Conference on Artificial Intelligence, AI'92, Morgan Kaufkann, San Mateo, CA, 1992, pp156-162.

[56] Schröder, F.W.K.E., *Vorlesungen über die Algebra der Logik (exacte Logik), Dritter Band, Algebra und Logik der Relative, Erste Abteilung*, Leipzig, 1895. Reprinted Chelsea Publishing Co., N.Y., 1966.

[57] Susswein, S., *Parallel Path Consistency*, MS Thesis, University of Utah, Department of Computer Science, 1991.

[58] Susswein, S., Henderson, T.C., Zachary, J., Hinker, P., Hansen, C., and Marsden, G., *Parallel Path Consistency*, University of Utah, Technical Report UUCS-91-010, July 30, 1991.

[59] Tarski, A., *On the Calculus of Relations*, Journal of Symbolic Logic 6, 1941, pp73–89.

[60] Tarski, A., *Contributions to the Theory of Models, III*, Koninklijkle Nederlandse Akademie van Wetenschappen. Proceedings. Series A. Mathematical Sciences (Indagationes Mathematicae 17) 58, 1955, pp56–64.

[61] Tarski, A., and Givant, S.R., *A Formalization of Set Theory without Variables*, Colloquium Publications in Math. 41, Amer. Math. Soc., 1987.

[62] van Beek, P.G., and Cohen, R., *Approximation Algorithms for Temporal Reasoning*, Proceedings of IJCAI89, the 11th Joint Conference on Artifical Intelligence, 1291–1296, Morgan Kaufmann 1989 (short version); *Exact and Approximate Reasoning about Temporal Relations*, Computational Intelligence 6, 1990, pp132–144 (long version).

[63] van Beek, P.G., *Exact and Approximate Reasoning about Qualitative Temporal Relations*, Ph.D. Thesis (U. Waterloo), Technical Report TR 90–29, University of Alberta, Edmonton, 1990.

[64] van Beek, P.G., *Reasoning about Qualitative Temporal Information*, Proceedings of AAAI-90, the Eighth National Conference on Artificial Intelligence, AAAI Press 1990, pp728–734.

[65] Vilain, M., and Kautz, H., *Constraint Propagation Algorithms for Temporal Reasoning*, Proceedings of AAAI-86, Morgan Kaufmann, 1986, pp377-382.

[66] Vilain, M., Kautz, H., and van Beek, P.G., *Constraint Propagation Algorithms for Temporal Reasoning*, in Weld and de Kleer, eds., *Readings in Qualititative Reasoning About Physical Systems*, Morgan Kaufmann 1989, pp373-381, revised version of [65].

[67] Waltz, D.L., *Generating Semantic Desciptions from Drawings of Scenes with Shadows*, Technical Report AI-TR-271, Massachusetts Institute of Technology, 1972.