Cardinality of Relations and Relational Approximation Algorithms

Rudolf Berghammer[a,*], Peter Höfner[b], Insa Stucke[a]

[a] *Institut für Informatik*
Christian-Albrechts-Universität zu Kiel
Olshausenstraße 40, 24098 Kiel, Germany
[b] *NICTA, Australia, and*
Computer Science and Engineering, University of New South Wales, Australia

Abstract

First, we discuss three specific classes of relations, which allow to model the essential constituents of graphs, such as vertices and (directed or undirected) edges. Based on Kawahara's characterisation of the cardinality of relations we then derive fundamental properties on their cardinalities. As main applications of these results, we formally verify four relational programs, which implement approximation algorithms by using the assertion-based method and relation-algebraic calculations.

Keywords: Relation algebra, cardinality operation, point, atom, edge, approximation algorithm, assertion-based program verification.

1. Introduction

Formal program verification means to show with mathematical rigour that a program is correct with respect to a formal problem specification. In the case of imperative programs, and when using the assertion-based Floyd-Hoare-approach, specifications usually consist of pre- and post-conditions. Program verification is then mainly based on loop invariants. To support formal program verification, adequate algebraic frameworks are very helpful since they support calculational reasoning and often even equational reasoning.

Relation algebra (as introduced in [26] and further developed in [17, 24, 25, 27]) plays a prominent role for computational problems on discrete structures such as (directed or undirected) graphs. This is due to the fact that relations and many discrete structures are essentially the same or closely related. For instance, a directed graph is nothing else than a relation on a set of vertices and also for other classes of graphs there are simple and elegant ways to model them relation-algebraically, as shown in [24], for example. With regard to proofs, the use of relation algebra has the advantage to frequently clarify the proof structure, to reduce the danger of wrong proof steps and to open the possibility for proof mechanisation, e.g., by automated theorem provers or proof assistant tools. Examples for the latter can be found in [7, 8, 9, 14] using the automated theorem prover Prover9 and the proof assistants Coq and Isabelle/HOL, respectively. Finally, the relation-algebraic framework supports prototyping and validation tasks in a significant manner, for instance, by computer systems like RelView; see e.g., [5, 9, 29].

In [15] Kawahara acknowledges the considerable influence of the Schmidt and Ströhlein's textbook [24] to the formal study of graphs from a relational point of view. However, he also mentions that "... the cardinality of relations is treated rather implicitly or intuitively ..." [15, Page 251]. To close this gap, he develops a cardinality operation on relations and demonstrates that the corresponding laws can be used to reason about cardinalities in a calculational and algebraic manner.

The present paper is a continuation of Kawahara's work. Whereas he considers applications in basic graph theory (theorems of Hall and König), we are interested in applications concerning relational programs and their formal development/verification, as initiated in [2]. We concentrate on approximation algorithms, where cardinalities of

*Corresponding author: Tel.: ++49 431 880 7272 Fax: ++49 431 880 7613 Email: rub@informatik.uni-kiel.de

certain sets play an important role when proving the desired approximation bound. To the best of our knowledge such algorithms have not yet been analysed using relation algebra, and hence we extend the area of applications for relation algebra. Experience has shown that a relation-algebraic approach to algorithms and programming besides the basic constants, operations and tests of relation algebra frequently requires the use of further operations. In our previous work especially choice operations for relational points and single-pair-relations (corresponding to the choice of a vertex or an edge in graph theory) play an important role as, for example, demonstrated in [2, 3, 5, 6]. In [2] these concepts are defined via certain laws and hence can be seen as an axiomatic extension of relation algebra. This extension is in line with other extensions such as projection relations [24] and embedding relations [6]. Therefore, the laws of [15] constitute a further axiomatic extension of relation algebra by a cardinality operation. Because of the importance of the choice operations, a natural task is to investigate the cardinalities of the objects they compute.

The remainder of the paper is organised as follows: in Section 2 we present the relation-algebraic preliminaries. Next, in Section 3, we introduce the concepts of relational points, atoms and edges, i.e., of the objects the choice functions compute, and prove some of their fundamental properties. In Section 4 we use Kawahara's characterisation of the cardinality of relations as an axiomatisation of a cardinality operation and present fundamental properties concerning the cardinality of the composition of univalent relations and mappings, as well as of points with specific types, atoms and edges. We use the cardinality axioms and properties in the following Sections 5 to 8 to formally verify relational variants of four well-known approximation algorithms by combining relation-algebraic reasoning with the assertion-based program verification method. We conclude in Section 9 with a short summary and some remarks concerning tool support and future work.

2. Relation-Algebraic Preliminaries

In this section we recall the fundamentals of relation algebra based on the heterogeneous approach of [24, 25]. Set-theoretic relations form the standard model of relation algebras. We assume the reader to be familiar with the basic operations on them, viz. R^T (transposition, conversion), $\overline{R}$ (complementation, negation), $R \cup S$ (union), $R \cap S$ (intersection), $R;S$ (composition), the predicates $R \subseteq S$ (inclusion) and $R = S$ (equality), as well as the special relations O (empty relation), L (universal relation) and I (identity relation). The Boolean operations, the inclusion and the constants O and L form Boolean lattices. Further well-known properties are the distributivity laws $Q;(R \cup S) = Q;R \cup Q;S$, $(R \cup S)^\mathsf{T} = R^\mathsf{T} \cup S^\mathsf{T}$ and $(R \cap S)^\mathsf{T} = R^\mathsf{T} \cap S^\mathsf{T}$, the sub-distributivity law $Q;(R \cap S) \subseteq Q;R \cap Q;S$, the laws $\overline{R^\mathsf{T}} = \overline{R}^\mathsf{T}$, $(R^\mathsf{T})^\mathsf{T} = R$ and $(R;S)^\mathsf{T} = S^\mathsf{T};R^\mathsf{T}$, and the monotonicity of transposition, union, intersection and composition.

The theoretical framework for these laws (and many others) to hold is that of a (heterogeneous) *relation algebra* in the sense of [24, 25], with typed relations as elements. This implies that each relation has a source and a target. We write $R : X \leftrightarrow Y$ to express that X is the source, Y is the target and $X \leftrightarrow Y$ is the type of R. In case of set-theoretic relations $R : X \leftrightarrow Y$ means that R is a subset of $X \times Y$. As constants and operations of a relation algebra we have those of set-theoretic relations, where we frequently overload the symbols O, L and I, i.e., avoid the binding of types to them. Only when necessary we use indices such as L_{XY} for the universal relation of type $X \leftrightarrow Y$ and I_{XX} for the identity relation of type $X \leftrightarrow X$. The axioms of a relation algebra are

(1) the axioms of a Boolean lattice for all relations of the same type under the Boolean operations, the inclusion, the empty relation and the universal relation,

(2) the associativity of composition and that identity relations are neutral elements with respect to composition,

(3) that $Q;R \subseteq S$, $Q^\mathsf{T};\overline{S} \subseteq \overline{R}$ and $\overline{S};R^\mathsf{T} \subseteq \overline{Q}$ are equivalent, for all relations Q, R and S (with appropriate types),

(4) that $R \neq \mathsf{O}$ implies $\mathsf{L};R;\mathsf{L} = \mathsf{L}$, for all relations R and all universal relations (with appropriate types).

In [24] the laws of (3) are called the *Schröder rules* and (4) is called the *Tarski rule*. In the relation-algebraic proofs of this paper we will only mention applications of (3), (4) and 'non-obvious' consequences of the axioms, like

$$Q;R \cap S \subseteq (Q \cap S;R^\mathsf{T});(R \cap Q^\mathsf{T};S) ,$$

for all relations $Q : X \leftrightarrow Y$, $R : Y \leftrightarrow Z$ and $S : X \leftrightarrow Z$. In [24] this inequality is called the *Dedekind rule*. Furthermore, we will assume that complementation and transposition bind stronger than composition, composition binds stronger

than union and intersection, and that all relation-algebraic expressions and formulae are well-typed. The latter assumption allows us to suppress many type annotations, since types of relations can be derived from other relations with known types, using the typing rules of the relational operations.

In the following we recapitulate some well-known classes of relations used in the remainder of this paper. For more details see e.g. [24, 25].

A relation R is called *univalent* if $R^\mathsf{T};R \subseteq \mathsf{I}$, and *total* if $R;\mathsf{L} = \mathsf{L}$, which is equivalent to $\mathsf{I} \subseteq R;R^\mathsf{T}$. A *mapping* is a univalent and total relation. For a univalent R we have $R;(Q \cap S) = R;Q \cap R;S$ and $R;\overline{S} \subseteq \overline{R;S}$ and for a total R we have $R;\overline{S} \supseteq \overline{R;S}$, for all Q and S. A relation R is called *injective* if R^T is univalent and *surjective* if R^T is total. Hence, if S is injective, then $\overline{R};S \subseteq \overline{R;S}$, and if S is surjective, then $\overline{R};S \supseteq \overline{R;S}$, for all R. Relations of type $X \leftrightarrow X$ are called *homogeneous*. A homogenous relation R is called *reflexive* if $\mathsf{I} \subseteq R$, *irreflexive* if $R \subseteq \overline{\mathsf{I}}$, *symmetric* if $R = R^\mathsf{T}$ and *transitive* if $R;R \subseteq R$.

By definition, a *vector* is a relation v with $v = v;\mathsf{L}$. Usually vectors are denoted by lower-case letters. For $v : X \leftrightarrow Y$ the condition $v = v;\mathsf{L}$ means that v can be written in the form $v = Z \times Y$ with a subset Z of X. We say that *v models the subset Z* of X. For this purpose the target of a vector is irrelevant. Therefore, we often use the specific singleton set $\mathbf{1}$ as target. In the Boolean matrix model of relations vectors correspond to Boolean matrices with only 1-entries or only 0-entries in each row. Thus, a vector of type $X \leftrightarrow \mathbf{1}$ corresponds to a Boolean column vector.

3. Relational Points, Atoms and Edges

In this section we introduce three further classes of relations and show some important properties. We start with the formal definitions.

Definition 3.1. *A* point *is a vector p such that $p \neq \mathsf{O}$ and $p;p^\mathsf{T} \subseteq \mathsf{I}$. A relation a is a (relational)* atom*, if $a \neq \mathsf{O}$, $a^\mathsf{T};\mathsf{L};a \subseteq \mathsf{I}$ and $a;\mathsf{L};a^\mathsf{T} \subseteq \mathsf{I}$. A relation e is an* edge*, if there exists an atom a such that $e = a \cup a^\mathsf{T}$.*

The concept of a point is introduced in [23] and used for proving a representation theorem for relation algebras. As a vector, a point models a singleton subset of its source, or an element of it if a singleton set is identified with the only element it contains. Since points allow to model elements, they are frequently used in relational algorithms, e.g., to model single vertices of graphs or elements of ordered sets. See [2, 3, 5, 6] for some applications. Our axiomatisation of atoms stems from [16], but generalises the definition of [16] from the homogeneous to the heterogeneous case. A similar axiomatisation of atoms can be found in [2]. As we will show in the next section, an atom is a relation that consists of precisely one element in the set-theoretic model. So, in relational algorithms atoms can be used to model single directed edges of directed graphs or pairs of incomparable elements of ordered sets, as e.g. done in [2, 5]. An edge is a relation of the specific form $\{(x, y), (y, x)\}$, i.e., an edge can be represented as the union of an atom and its transpose. In graph-theoretic applications edges are appropriate models for single edges of undirected graphs.

Each point is an atom in the lattice-theoretic sense among the vectors of its type; see [24, Proposition 2.4.5]. As a consequence, if p and q are points such that $p \neq q$, then $p \cap q = \mathsf{O}$. Furthermore, if p is a point, then p is injective and surjective and hence $(R \cap S);p = R;p \cap S;p$ of p and $\overline{R;p} = \overline{R};p$ holds for all relations R and S. Surjectivity is a consequence of $p \neq \mathsf{O}$, the Tarski rule and $p = p;\mathsf{L}$.

In this paper we only consider set-theoretic relations. However, we do not argue point-wisely but treat such relations in a calculational, algebraic manner only. This approach is based on some fundamental properties of the operations, predicates and constant relations we have presented in Section 2, which are taken as axioms, and laws, which can be derived from the axioms. So far, we have introduced the four axioms of a relation algebra. Now, we introduce a further axiom. For set-theoretic relations the following *point axiom* of [12] holds, where for a given vector v we denote by $\mathcal{P}_v$ the set of all points p such that $p \subseteq v$.

Axiom 3.1. *For all sets X we have $\mathsf{L}_{X\mathbf{1}} = \bigcup_{p \in \mathcal{P}_{\mathsf{L}_{X\mathbf{1}}}} p$.*

In words, the point axiom states that each universal vector with target $\mathbf{1}$ is the union of the points it contains. The following lemma states that this property even holds for all vectors with target $\mathbf{1}$. It is shown in [12] as Proposition 3.3.4.

Lemma 3.1. *If $v : X \leftrightarrow \mathbf{1}$ is a vector, then $v = \bigcup_{p \in \mathcal{P}_v} p$.* □

As a consequence, each non-empty vector contains a point. Having presented some fundamental properties of points, we now do the same for atoms and edges. As each non-empty vector contains a point, each non-empty relation contains an atom and each non-empty symmetric relation contains an edge. Some of the following properties can already be found in [2].

Lemma 3.2. *If $a : X \leftrightarrow Y$ is an atom, then the following properties hold, where in (3) and (4) it is assumed that $X = Y$:*

(1) $a^\mathsf{T};a \subseteq \mathsf{I}$ *(univalence)* (2) $a;a^\mathsf{T} \subseteq \mathsf{I}$ *(injectivity)* (3) $a;a \subseteq a$ *(transitivity)* (4) $a;a \subseteq \mathsf{I}$

Proof. Univalence of a is shown by

$$a^\mathsf{T};a \subseteq a^\mathsf{T};\mathsf{L};a \subseteq \mathsf{I} .$$

In the same way injectivity of a follows. The proof of transitivity is as follows:

$$\begin{array}{lll} a;a &= a;a \cap a;a & \\ &\subseteq a;\mathsf{L} \cap \mathsf{L};a & \\ &\subseteq (a \cap \mathsf{L};a;\mathsf{L}^\mathsf{T});(\mathsf{L} \cap a^\mathsf{T};\mathsf{L};a) & \text{Dedekind rule} \\ &\subseteq a;a^\mathsf{T};\mathsf{L};a & \\ &\subseteq a & \text{as } a^\mathsf{T};\mathsf{L};a \subseteq \mathsf{I} \end{array}$$

The last property is shown by the calculation

$$a;a = a;a \cap a \subseteq (a \cap a;a^\mathsf{T});(a \cap a^\mathsf{T};a) \subseteq a;a^\mathsf{T};a^\mathsf{T};a \subseteq \mathsf{I} ,$$

using transitivity, the Dedekind rule, injectivity and univalence of a. □

In [24], Proposition 4.2.2 iv), it is shown that for all univalent relations R and all relations Q satisfying $Q \subseteq R$ and $R;\mathsf{L} \subseteq Q;\mathsf{L}$ for some non-empty universal relation L it follows $Q = R$. In combination with (1) of Lemma 3.2 this allows us to prove that an atom in the sense of Definition 3.1 is in fact an atom in the lattice-theoretic sense, such that different atoms have an empty intersection.

Theorem 3.1. *If $a : X \leftrightarrow Y$ is an atom, then it is an atom in the lattice-theoretic sense among the relations of type $X \leftrightarrow Y$.*

Proof. We consider the vector $a;\mathsf{L} : X \leftrightarrow Y$ with $\mathsf{L} : Y \leftrightarrow Y$. This vector is injective due to

$$(a;\mathsf{L});(a;\mathsf{L})^\mathsf{T} = a;\mathsf{L};a^\mathsf{T} \subseteq \mathsf{I} .$$

Furthermore, we have $a;\mathsf{L} \neq \mathsf{O}$, since from $a;\mathsf{L} = \mathsf{O}$ we get

$$\mathsf{L}_{\mathbf{1}Y} = \mathsf{L}_{\mathbf{1}X};a;\mathsf{L}_{YY} = \mathsf{L}_{\mathbf{1}X};\mathsf{O}_{XY} = \mathsf{O}_{\mathbf{1}Y}$$

because of $a \neq \mathsf{O}$ and the Tarski rule. $a \neq \mathsf{O}$ implies $Y \neq \emptyset$ and, as a consequence, $\mathsf{L}_{\mathbf{1}Y} = \mathsf{O}_{\mathbf{1}Y}$ is a contradiction.

Now, we prove the desired result. By definition we have $a \neq \mathsf{O}$. Assume an arbitrary relation $R : X \leftrightarrow Y$ such that $R \neq \mathsf{O}$ and $R \subseteq a$. Then we have

$$\mathsf{O} \neq R = R;\mathsf{I} \subseteq R;\mathsf{L} \subseteq a;\mathsf{L}$$

and this implies $R;\mathsf{L} = a;\mathsf{L}$, since the point $a;\mathsf{L} : X \leftrightarrow Y$ is an atom in the lattice-theoretic sense among the vectors of type $X \leftrightarrow Y$. Now, Lemma 3.2 (1) and [24], Proposition 4.2.2 iv), show $R = a$. □

For an atom $a : X \leftrightarrow Y$ not every universal relation L leads to a point $a;\mathsf{L}$: for $\mathsf{L} : Y \leftrightarrow \emptyset$ we get $a;\mathsf{L} = \mathsf{O}$ since there is only one relation of type $Y \leftrightarrow \emptyset$. If, however, the target of L is non-empty, then $a;\mathsf{L}$ is a point and the same holds for $a^\mathsf{T};\mathsf{L}$, since atoms are closed under transposition. Based on Lemma 3.2, we now list some fundamental properties of edges.

Lemma 3.3. *If $e : X \leftrightarrow X$ is an edge, then $e = e^\mathsf{T}$ (symmetry) and $e;e \subseteq \mathsf{I}$. In particular, e is univalent and injective.*

Proof. By definition there exists an atom a such that $e = a \cup a^\mathsf{T}$. This yields $e = e^\mathsf{T}$ and, in combination with (1), (2) and (4) of Lemma 3.2, also

$$e;e = (a \cup a^\mathsf{T});(a \cup a^\mathsf{T}) = a;a \cup a;a^\mathsf{T} \cup a^\mathsf{T};a \cup (a;a)^\mathsf{T} \subseteq \mathsf{I} \,. \qquad \square$$

Different edges are also disjoint. This is a consequence of the following result that characterises edges as lattice-theoretic atoms, too.

Theorem 3.2. *If $e : X \leftrightarrow X$ is an edge, then it is an atom in the lattice-theoretic sense among the symmetric relations of type $X \leftrightarrow X$.*

Proof. By definition there exists an atom a such that $e = a \cup a^\mathsf{T}$, hence $e \neq \mathsf{O}$. Next, assume an arbitrary relation R such that $R = R^\mathsf{T}$, $R \neq \mathsf{O}$ and $R \subseteq e$. To show $R = e$ we distinguish two cases. First, assume $a = a^\mathsf{T}$. With the help of Theorem 3.1 we get

$$R \subseteq e \iff R \subseteq a \implies R = a \iff R = e \,.$$

Next, assume $a \neq a^\mathsf{T}$. Since both a and a^T are atoms, Theorem 3.1 yields $a \cap a^\mathsf{T} = \mathsf{O}$. So, from $R \neq \mathsf{O}$ we get $R \not\subseteq a$ or $R \not\subseteq a^\mathsf{T}$. In the first case we have

$$\begin{array}{lll} R \not\subseteq a & \iff R \cap \overline{a} \neq \mathsf{O} & \\ & \iff R \cap \overline{a} = a^\mathsf{T} & \text{by Theorem 3.1, } R \subseteq a \cup a^\mathsf{T} \\ & \iff R \cap \overline{a}^\mathsf{T} = a & \text{as } R = R^\mathsf{T} \end{array}$$

and this yields

$$R = R \cap \mathsf{L} = R \cap \overline{a \cap a^\mathsf{T}} = R \cap (\overline{a} \cup \overline{a}^\mathsf{T}) = (R \cap \overline{a}) \cup (R \cap \overline{a}^\mathsf{T}) = a^\mathsf{T} \cup a = e \,.$$

In the same way it can be shown that $R = e$ follows from $R \not\subseteq a^\mathsf{T}$. $\square$

As already mentioned in the introduction, the choice of a point from a non-empty vector and of an atom from a non-empty relation are fundamental for relational algorithms. Therefore, we assume two operations *point* and *atom* to be at hand such that *point*(v) is a point with *point*(v) $\subseteq v$, for all non-empty vectors v, and *atom*(R) is an atom with *atom*(R) $\subseteq R$, for all non-empty relations R. Both operations are assumed to be (deterministic) functions in the usual mathematical sense. Hence, each call *point*(v) yields the same point in the non-empty vector v and each call *atom*(R) yields the same atom in the non-empty relation R such that *point*(v) = *point*(v) and *atom*(R) = *atom*(R) are true. This property allows us to realise the choice of an edge by *edge*(R) = *atom*(R) $\cup$ *atom*(R)$^\mathsf{T}$. The above requirements for *point* and *atom* enable different realisations. For instance, those in the programming language of RelView (where *point* and *atom* are pre-defined operations) use that the tool deals only with relations on finite sets, which are linearly ordered by an internal enumeration. In RelView the call *point*(v) returns the point that models the least element of the set modelled by v and the call *atom*(R) yields the subrelation of R that consists of its lexicographically least pair.

4. Cardinality of Relations

In [15] Kawahara discusses the cardinality of set-theoretic relations. The main results are a formula, called Dedekind inequality, that allows a calculational treatment of cardinalities of compositions of relations and, based on it, an algebraic characterisation of cardinalities of finite relations. If the properties of this characterisation (Theorem 2 of [15]) are considered as axiomatic specification of a cardinality operation $|\cdot|$ that assigns to all relations R a cardinality $|R|$, then this leads to the following five axioms. Together with the four axioms of a relation algebra and the point axiom they constitute the list of ten axioms on which the results of this paper are based upon.

Axioms 4.1. *For all relations Q, R and S with appropriate types it holds:*

(C1) If R is finite, then $|R| \in \mathbb{N}$ and $|R| = 0$ iff $R = \mathsf{O}$.

(C2) $|R| = |R^\mathsf{T}|$.

(C3) If R and S are finite, then $|R \cup S| = |R| + |S| - |R \cap S|$.

(C4) If Q is univalent, then $|R \cap Q^\mathsf{T};S| \leq |Q;R \cap S|$ *and* $|Q \cap S;R^\mathsf{T}| \leq |Q;R \cap S|$.

(C5) $|\mathbb{1}_{\mathbf{1}}| = 1$.

In the cardinality axioms (C1) and (C3) the relations in question are assumed to be finite such that the cardinality $|R|$ can be regarded as a natural number and cardinality arithmetic can be avoided; in the cardinality axioms (C2) and (C4) the notation $|R| = |S|$ (respectively $|R| \leq |S|$) is equivalent to the fact that there exists a bijection between R and S (respectively an injection from R to S) and cardinality axiom (C5) says that the identity relation on the singleton set $\mathbf{1}$ consists of precisely one pair. To simplify the presentation and to avoid additional pre-conditions in lemmas and theorems, throughout this paper we assume in case of an expression $|R|$ the sets of R's type to be finite and thereby $|R| \in \mathbb{N}$. This suffices for our applications.

The cardinality axioms (C1) and (C3) imply monotonicity of $|\cdot|$, i.e., $R \subseteq S$ implies $|R| \leq |S|$, for all R and S. Even *strict monotonicity* holds, i.e., $R \subset S$ implies $|R| < |S|$, for all R and S. Based on the above cardinality axioms (C1) to (C5), many algebraic laws are derived in a purely calculational manner in [15]. In this paper we make use of the following one (see [15, Corollary 1 (c)]):

Lemma 4.1. *If $R : X \leftrightarrow Y$ is univalent and $S : Y \leftrightarrow Z$ is a mapping, then* $|R;S| = |R|$. □

Furthermore, we need that $|Q \cap R;S| = |Q;S^\mathsf{T} \cap R|$ if R and S are univalent relations (in [15] shown as Corollary 1 (a)) for proving the following auxiliary result:

Lemma 4.2. *Given $R : X \leftrightarrow X$ and $P, Q : X \leftrightarrow Z$ such that R is symmetric, P is injective and Q is univalent, we have* $|R \cap P;Q^\mathsf{T}| = |R;P \cap Q|$.

Proof. We get the desired result by the following calculation using cardinality axiom (C2), then [15, Corollary 1 (a)] (P^T is univalent) and, finally, $R = R^\mathsf{T}$:

$$|R \cap P;Q^\mathsf{T}| = |R^\mathsf{T} \cap Q;P^\mathsf{T}| = |R^\mathsf{T};(P^\mathsf{T})^\mathsf{T} \cap Q| = |R;P \cap Q|$$ □

Next, we consider the cardinalities of points, atoms and edges, which are derived only from the cardinality axioms and the presented consequences. We start with points having the singleton set $\mathbf{1}$ as target.

Lemma 4.3. *If $p : X \leftrightarrow \mathbf{1}$ is a point, then* $|p| = 1$.

Proof. The statement follows from

$$|p| = |p^\mathsf{T}| = |\mathbb{1}_{\mathbf{1}};p^\mathsf{T}| = |\mathbb{1}_{\mathbf{1}}| = 1 \ ,$$

using cardinality axioms (C2) and (C5) and Lemma 4.1 ($\mathbb{1}_{\mathbf{1}}$ is univalent and $p^\mathsf{T} : \mathbf{1} \leftrightarrow X$ is a mapping). □

As generalisation we get $|p| = |Y|$ if p is a point of type $X \leftrightarrow Y$, but in the following sections we only will apply Lemma 4.3. This lemma also allows us to show that the cardinality of a vector with target $\mathbf{1}$ equals the cardinality of the set of all points it contains.

Lemma 4.4. *For all $v : X \leftrightarrow \mathbf{1}$ we have* $|v| = |\mathcal{P}_v|$.

Proof. Because of Lemma 3.1, cardinality axioms (C3) and (C1) (the points of $\mathcal{P}_v$ are pair-wise disjoint) and Lemma 4.3 we obtain the claim by

$$|v| = |\bigcup_{p \in \mathcal{P}_v} p| = \sum_{p \in \mathcal{P}_v} |p| = |\mathcal{P}_v| \ .$$ □

For vectors $v : X \leftrightarrow Y$ this lemma generalises to $|v| = |\mathcal{P}_v| \cdot |Y|$. The next lemma states that atoms contain precisely one pair.

Lemma 4.5. *If $a : X \leftrightarrow Y$ is an atom, then $|a| = 1$.*

Proof. We consider the point $a;\mathsf{L}_{Y\mathbf{1}} : X \leftrightarrow \mathbf{1}$. Since the atom a is univalent by Lemma 3.2 (1) and $\mathsf{L}_{Y\mathbf{1}} : Y \leftrightarrow \mathbf{1}$ is a mapping, Lemma 4.1 is applicable and yields in conjunction with Lemma 4.3 that $|a| = |a;\mathsf{L}_{Y\mathbf{1}}| = 1$. □

In case of edges e we have to investigate the cardinality of the union of two atoms. If $e = a \cup a^\mathsf{T}$ and $a = a^\mathsf{T}$, then $|e| = 1$ follows immediately from the previous lemma. If a relation is interpreted as a graph, this may happen if the only pair of a forms a loop. If $e = a \cup a^\mathsf{T}$ and $a \neq a^\mathsf{T}$, then $|e| = 2$ can be shown as follows.

Lemma 4.6. *If $e : X \leftrightarrow Y$ is an irreflexive edge, then $|e| = 2$.*

Proof. By definition there exists an atom a such that $e = a \cup a^\mathsf{T}$. Because of Lemma 3.2 (3) and (4), and a Schröder rule we have

$$e \subseteq \bar{\mathsf{I}} \implies a \subseteq \bar{\mathsf{I}} \implies a;a \subseteq \bar{\mathsf{I}} \implies a;a \subseteq \mathsf{O} \iff a^\mathsf{T};\mathsf{L} \subseteq \overline{a}\,.$$

This yields $a \neq a^\mathsf{T}$, as $a = a^\mathsf{T}$ would lead to $a \subseteq a;\mathsf{L} \subseteq \overline{a}$, i.e., to the contradiction $a = \mathsf{O}$. So, Theorem 3.1 shows $a \cap a^\mathsf{T} = \mathsf{O}$. Finally, to conclude the proof we apply cardinality axioms (C3), (C1) and (C2), and Lemma 4.5:

$$|e| = |a \cup a^\mathsf{T}| = |a| + |a^\mathsf{T}| - |a \cap a^\mathsf{T}| = |a| + |a^\mathsf{T}| = |a| + |a| = 2$$ □

5. Relational Approximation of Minimum Vertex Covers

In this and the following three sections we use the notions and results of the last two sections to formally verify relational versions of approximation algorithms for *NP*-hard optimisation problems. We start with an approximation algorithm for minimum vertex covers as described e.g. in [10]. The authors of [10] attribute this algorithm to Gavril and Yannakakis.

In the remainder we assume an undirected (loop-free) graph g to be given, with a non-empty and finite set X of vertices and a set E of edges. Each edge is a set that consists of precisely two (different) vertices. In this section we model $g = (X, E)$ by an *adjacency relation* $R : X \leftrightarrow X$, that is defined by $x\,R\,y$ iff $\{x, y\} \in E$, for all $x, y \in X$. By this definition R becomes irreflexive and symmetric.

We take the adjacency relation R as input for the relational program we want to verify; the pre-condition $Pre(R)$ of the program is the conjunction of the following two formulae, which specify R as irreflexive and symmetric:

$$R \subseteq \bar{\mathsf{I}} \qquad\qquad R = R^\mathsf{T}$$

A *vertex cover* of g is a set C of vertices such that $C \cap e \neq \emptyset$, for all $e \in E$. A point-wise calculation shows that a vector $c : X \leftrightarrow \mathbf{1}$ models a vertex cover of g iff $R \subseteq c;\mathsf{L} \cup (c;\mathsf{L})^\mathsf{T}$, where $\mathsf{L} : \mathbf{1} \leftrightarrow X$. The approximation algorithm of Garvil and Yannakakis always returns a vertex cover whose cardinality is guaranteed to be at most twice the cardinality of any vertex cover (or, equivalently, of any minimum vertex cover). This leads to the conjunction of the following two formulae as post-condition $Post(R, c)$ of the program:

$$R \subseteq c;\mathsf{L} \cup (c;\mathsf{L})^\mathsf{T} \qquad\qquad \forall\, d : X \leftrightarrow \mathbf{1} \bullet R \subseteq d;\mathsf{L} \cup (d;\mathsf{L})^\mathsf{T} \Rightarrow |c| \leq 2 \cdot |d|$$

As a relational while-program, the approximation algorithm of Garvil and Yannakakis looks as follows, where we consider the let-clause as syntactical sugar.

$$\begin{array}{l} c, M, S := \mathsf{O}_{X\mathbf{1}}, \mathsf{O}, R; \\ \textbf{while } S \neq \mathsf{O} \textbf{ do} \\ \quad \textbf{let } e = edge(S); \\ \quad c, M, S := c \cup e;\mathsf{L}, M \cup e, S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e} \textbf{ od} \end{array} \tag{GY}$$

From the initialisation we see that c is of type $X \leftrightarrow \mathbf{1}$. The typing $M, S, e : X \leftrightarrow X$ can be derived from R's type $X \leftrightarrow X$, the initialisation of S and the typing rules of union and *edge*. Similarly, the types of the constants O and L can be derived. For example, from $c : X \leftrightarrow \mathbf{1}$ and $e : X \leftrightarrow X$ we get $\mathsf{L} : X \leftrightarrow \mathbf{1}$ for the L of $c \cup e;\mathsf{L}$ and $\mathsf{L} : X \leftrightarrow X$ for both L of $e;\mathsf{L} \cup \mathsf{L};e$.

Informally, in each run of the loop the program (GY) selects an edge from the graph S, adds its vertices to the present vector c and removes all edges that are incident to the selected one. M serves as an auxiliary variable since it is neither used in the pre- , the post-condition, the guard of the loop nor in assignments to the other variables c and S. It collects all edges that were selected by the calls $edge(S)$. In graph-theoretic terms M models a *matching* of g (a set of pair-wise non-incident edges) that has no incident edge with the subgraph of g modelled by S. Relation-algebraically this fact can be described as follows:

$$(1)\ M \subseteq R \qquad (2)\ M = M^\mathsf{T} \qquad (3)\ M;M \subseteq \mathsf{I} \qquad (4)\ M;\mathsf{L} \cap S = \mathsf{O}$$

We take the conjunction of formulae (1) to (4) as first part of the loop invariant $Inv(R, c, M, S)$. Note that M contains for each edge $\{x, y\}$ of the matching two pairs, viz. (x, y) and (y, x); hence M is symmetric. The remaining four formulae of the conjunction that defines $Inv(R, c, M, S)$ are:

$$(5)\ R \cap \overline{S} \subseteq c;\mathsf{L} \cup (c;\mathsf{L})^\mathsf{T} \qquad (6)\ S \subseteq R \qquad (7)\ S = S^\mathsf{T} \qquad (8)\ |c| \leq |M|$$

Formula (5) generalises the formula $R \subseteq c;\mathsf{L} \cup \mathsf{L};c^\mathsf{T}$ of the post-condition $Post(R, c)$, (6) and (7) are auxiliary formulae and formula (8) specifies the decisive property for proving the approximation bound 2 for the program (GY).

To show that the program (GY) is totally correct with respect to the pre-condition $Pre(R)$ and the post-condition $Post(R, c)$, we have to verify the four well-known proof obligations of assertion-based program verification; see e.g., [11, 13]. We start with the following lemma. It says that the loop invariant is established by the initialisation if the pre-condition holds.

Lemma 5.1. *If $R : X \leftrightarrow X$ satisfies $Pre(R)$, then we have $Inv(R, \mathsf{O}_{X\mathbf{1}}, \mathsf{O}, R)$.*

Proof. Formula (7) of $Inv(R, c, M, S)$ follows from $Pre(R)$ and formula (8) from the cardinality axiom (C1); the proofs of the remaining formulae are trivial. □

By Lemma 5.1 we have verified the first proof obligation. As usual, the hardest task is the verification of the second proof obligation, i.e., to verify that the loop invariant is maintained by every execution of the body of the loop. The corresponding statement reads as follows.

Lemma 5.2. *Assume $R, M, S : X \leftrightarrow X$ and $c : X \leftrightarrow \mathbf{1}$ such that $Inv(R, c, M, S)$ is satisfied and $S \neq \mathsf{O}$. Then we have $Inv(R, c \cup e;\mathsf{L}, M \cup e, S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e})$, for all edges $e : X \leftrightarrow X$ with $e \subseteq S$.*

Proof. We show that the eight formulae (1) to (8) of the loop invariant $Inv(R, c, M, S)$ hold for the new values of c, M and S. Using (1), the assumption $e \subseteq S$ and (6) we obtain the first formula by

$$M \cup e \subseteq R \cup S \subseteq R \cup R = R\,.$$

The relation M is symmetric due to (2) and edges are symmetric by Lemma 3.3, thus

$$(M \cup e)^\mathsf{T} = M^\mathsf{T} \cup e^\mathsf{T} = M \cup e$$

shows the claim for the second formula. To verify it for the third formula, we calculate

$$\begin{array}{lll} (M \cup e);(M \cup e) &= M;M \cup M;e \cup e;M \cup e;e & \\ &\subseteq \mathsf{I} \cup M;e \cup e;M \cup \mathsf{I} & \text{by formula (3), Lemma 3.3} \\ &\subseteq \mathsf{I} \cup M;S \cup S;M & \text{as } e \subseteq S \\ &= \mathsf{I} \cup M;S \cup (M;S)^\mathsf{T} & \text{by formulae (2) and (7)} \end{array}$$

It suffices to prove $M;S = \mathsf{O}$. This is a consequence of (4) and (2), since these formulae in combination with a Schröder rule show

$$M;\mathsf{L} \cap S = \mathsf{O} \iff M;\mathsf{L} \subseteq \overline{S} \iff M^\mathsf{T};S \subseteq \mathsf{O} \iff M;S = \mathsf{O}.$$

For the fourth formula the claim is shown by the following calculation.

$$\begin{aligned}(M \cup e);\mathsf{L} \cap S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e} &= (M;\mathsf{L} \cup e;\mathsf{L}) \cap S \cap \overline{e;\mathsf{L}} \cap \overline{\mathsf{L};e} \\ &\subseteq ((M;\mathsf{L} \cap S) \cup (e;\mathsf{L} \cap S)) \cap \overline{e;\mathsf{L}} \\ &= e;\mathsf{L} \cap S \cap \overline{e;\mathsf{L}} && \text{by formula (4)} \\ &= \mathsf{O}\end{aligned}$$

A proof for the fifth formula is given by

$$\begin{aligned}R \cap \overline{S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e}} &= R \cap (\overline{S} \cup e;\mathsf{L} \cup \mathsf{L};e) \\ &= (R \cap \overline{S}) \cup (R \cap (e;\mathsf{L} \cup \mathsf{L};e)) \\ &\subseteq (R \cap \overline{S}) \cup e;\mathsf{L} \cup \mathsf{L};e \\ &\subseteq c;\mathsf{L} \cup (c;\mathsf{L})^{\mathsf{T}} \cup e;\mathsf{L} \cup \mathsf{L};e && \text{by formula (5)} \\ &= c;\mathsf{L} \cup e;\mathsf{L} \cup (c;\mathsf{L})^{\mathsf{T}} \cup (e;\mathsf{L})^{\mathsf{T}} && \text{as } e = e^{\mathsf{T}} \\ &= (c \cup e;\mathsf{L});\mathsf{L} \cup ((c \cup e;\mathsf{L});\mathsf{L})^{\mathsf{T}}\,.\end{aligned}$$

By means of (6), we show it for the sixth formula:

$$S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e} \subseteq S \subseteq R$$

The claim for the seventh formula, the symmetry of $S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e}$, is proven by the following calculation:

$$\begin{aligned}(S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e})^{\mathsf{T}} &= S^{\mathsf{T}} \cap \overline{(e;\mathsf{L})^{\mathsf{T}} \cup (\mathsf{L};e)^{\mathsf{T}}} \\ &= S \cap \overline{\mathsf{L}^{\mathsf{T}};e^{\mathsf{T}} \cup e^{\mathsf{T}};\mathsf{L}^{\mathsf{T}}} && \text{by formula (7)} \\ &= S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e} && \text{as } e = e^{\mathsf{T}}\end{aligned}$$

The proof of the claim for the last formula is given by

$$\begin{aligned}|c \cup e;\mathsf{L}| &\leq |c| + |e;\mathsf{L}| && \text{by cardinality axiom (C3)} \\ &\leq |M| + |e;\mathsf{L}| && \text{by formula (8)} \\ &= |M| + |e| && \text{by Lemma 4.1} \\ &= |M| + |e| - |M \cap e| && \text{as } M \cap e = \mathsf{O}, \text{and by cardinality axiom (C1)} \\ &= |M \cup e| && \text{by cardinality axiom (C3)}\end{aligned}$$

Lemma 4.1 is applicable since edges are univalent (see Lemma 3.3) and $\mathsf{L} : X \leftrightarrow \mathbf{1}$ is a mapping and $M \cap e = \mathsf{O}$ is shown by $M \cap e \subseteq M;\mathsf{L} \cap S = \mathsf{O}$ using the assumption $e \subseteq S$ and (4). □

The choice of e and formula (6) of the loop invariant $\mathit{Inv}(R, c, M, S)$ implies $e \subseteq S \subseteq R$. In particular, the edge e is irreflexive, since R is irreflexive. From $M \cap e = \mathsf{O}$ and Lemma 4.6, we get that the cardinality of M increases by 2 in every run through the loop. The third proof obligation is to show that the program terminates without an error. Since each call of the partial operation *edge* is defined due to the guard $S \neq \mathsf{O}$ of the loop, it suffices to show that the loop terminates. This follows from the next lemma in combination with the finiteness of the set X.

Lemma 5.3. *Given $S : X \leftrightarrow X$ with $S \neq \mathsf{O}$, we have $S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e} \subset S$, for all edges $e : X \leftrightarrow X$ with $e \subseteq S$.*

Proof. Since $S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e} \subseteq S$ it remains to show that $S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e} \neq S$. Here we use contradiction and assume $S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e} = S$. Then we have

$$S = S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e} \subseteq S \cap \overline{e} \subseteq S$$

and this yields $S \subseteq \overline{e}$, i.e., $e \subseteq \overline{S}$. In combination with the assumption $e \subseteq S$ we get $e = \mathsf{O}$. This contradicts the fact that edges are non-empty. □

So, we have $|S \cap \overline{e;\mathsf{L} \cup \mathsf{L};e}| < |S|$ by the strict monotonicity of the cardinality operation and we can take $|S|$ as measure for a termination proof. The fourth proof obligation is to verify that, after termination, i.e., if $S = \mathsf{O}$, the loop invariant implies the post-condition. It is shown by the following lemma.

Lemma 5.4. *If $R, M, S : X \leftrightarrow X$ and $c : X \leftrightarrow \mathbf{1}$ satisfy $Inv(R, c, M, S)$ and $S = \mathsf{O}$, then $Post(R, c)$ holds.*

Proof. From $S = \mathsf{O}$ and formula (5) of $Inv(R, c, M, S)$ we get

$$R = R \cap \mathsf{L} = R \cap \overline{S} \subseteq c;\mathsf{L} \cup (c;\mathsf{L})^{\mathsf{T}},$$

i.e., the first formula of $Post(R, c)$. To prove the second formula of $Post(R, c)$, let $d : X \leftrightarrow \mathbf{1}$ be an arbitrary vector such that $R \subseteq d;\mathsf{L} \cup (d;\mathsf{L})^{\mathsf{T}}$. Using one of the Schröder rules we have

$$d^{\mathsf{T}};\overline{d} \subseteq \mathsf{O} \iff d;\mathsf{L} \subseteq d \iff d \subseteq d$$

and thus

$$R;\overline{d} \subseteq (d;\mathsf{L} \cup (d;\mathsf{L})^{\mathsf{T}});\overline{d} = d;\mathsf{L};\overline{d} \cup \mathsf{L}^{\mathsf{T}};d^{\mathsf{T}};\overline{d} = d;\mathsf{L};\overline{d} \subseteq d;\mathsf{L} = d.$$

Using this estimation, we prove the approximation $|M| \leq 2 \cdot |d|$.

$$\begin{array}{lll}
|M| &= |M;(d \cup \overline{d})| & M \text{ univalent}, d \cup \overline{d} : X \leftrightarrow \mathbf{1} \text{ mapping, Lemma 4.1} \\
&= |M;d \cup M;\overline{d}| & \\
&\leq |M;d| + |M;\overline{d}| & \text{by cardinality axiom (C3)} \\
&\leq |M;d| + |R;\overline{d}| & \text{by formula (1) of } Inv(R, c, M, S), \text{ monotonicity of cardinality} \\
&\leq |M;d| + |d| & \text{as } R;\overline{d} \subseteq d, \text{ monotonicity of cardinality} \\
&= |\mathsf{L} \cap M^{\mathsf{T}\mathsf{T}};d| + |d| & \\
&\leq |M^{\mathsf{T}};\mathsf{L} \cap d| + |d| & M^{\mathsf{T}} \text{ univalent, by cardinality axiom (C4)} \\
&\leq |d| + |d| & \text{monotonicity of cardinality} \\
&= 2 \cdot |d|\,, &
\end{array}$$

where the univalence of M and M^{T} is a consequence of the formulae (2) and (3) of $Inv(R, c, M, S)$.

Now, formula (8) of $Inv(R, c, M, S)$ yields $|c| \leq |M| \leq 2 \cdot |d|$ and this concludes the proof of the second formula of $Post(R, c)$. □

The auxiliary variable M and its properties (1) to (4) in the loop invariant $Inv(R, c, M, S)$ allowed us a formal assertion-based correctness proof of the program (GY). However, M is not needed to determine c; therefore, it should be eliminated from (GY) to increase efficiency. By transforming proof outlines, in [4] it is shown that auxiliary variables always can be eliminated without affecting the correctness of programs. As a consequence, we obtain the following result:

Theorem 5.1. *For all relations $R : X \leftrightarrow X$ with X being non-empty and finite we have: If in the relational program* (GY) *all assignments to the auxiliary variable M are removed, then the resulting program is totally correct with respect to the pre-condition $Pre(R)$ and the post-condition $Post(R, c)$.* □

6. Adaptation to Hitting Sets

In this section we generalise the approximation algorithm of the previous section to hypergraphs $h = (X, H)$, where X is again a non-empty and finite set of vertices but, in contrast to undirected graphs, now the set H of hyperedges consists of non-empty subsets of X. As shown in [24], an *incidence relation* $I : X \leftrightarrow H$ is an adequate means for modelling h, where $x\,I\,e$ iff $x \in e$, for all $x \in X$ and $e \in H$.

The expression $\max\{|I;p| \mid p : H \leftrightarrow \mathbf{1} \text{ point}\}$ denotes the cardinality of all maximal hyperedges of h. This number is also called the *rank* of h. We adapt the relational program (GY) to incidence relations as inputs in such a way that it computes a vector $c : X \leftrightarrow \mathbf{1}$, which models a vertex cover – in this context usually called a *hitting set* – of h such that $|c|$ is no greater than k-times the cardinality of any hitting set of h, where k is the maximum cardinality of all hyperedges of h. As pre-condition $Pre(I, k)$ we use the conjunction of the following two formulae:

$$\mathsf{I} \subseteq I^{\mathsf{T}};I \qquad\qquad k = \max\{|I;p| \mid p : H \leftrightarrow \mathbf{1} \text{ point}\}$$

The first formula specifies the incidence relation I to be surjective. Surjectivity of I is equivalent to the fact that all hyperedges are non-empty sets of vertices. The second formula says that k is the cardinality of all maximum hyperedges of h. As in the case of adjacency relations a short point-wise calculation shows that $c : X \leftrightarrow \mathbf{1}$ models a hitting set of h iff $\mathsf{L} = I^{\mathsf{T}};c$. Together with the desired approximation bound k this leads to the conjunction of the following formulae as post-condition $Post(I, k, c)$:

$$\mathsf{L} = I^{\mathsf{T}};c \qquad\qquad \forall\, d : X \leftrightarrow \mathbf{1} \bullet \mathsf{L} = I^{\mathsf{T}};d \Rightarrow |c| \leq k \cdot |d|$$

The adaptation of the program (GY) to incidence relations and hitting sets is as follows, where we consider the let-clause again as syntactical sugar:

$$\begin{array}{l} c, m, s := \mathsf{O}_{X\mathbf{1}}, \mathsf{O}, \mathsf{L}; \\ \textbf{while } s \neq \mathsf{O} \textbf{ do} \\ \quad \textbf{let } p \;= point(s); \\ \quad c, m, s := c \cup I;p, m \cup p, s \cap \overline{I^{\mathsf{T}};I;p} \textbf{ od} \end{array} \tag{HS}$$

The typing $m, s, p : H \leftrightarrow \mathbf{1}$ and $\mathsf{L}, \mathsf{O} : H \leftrightarrow \mathbf{1}$ can be derived from the type $X \leftrightarrow H$ of I, the initialisation of c and the typing rules of the relational operations. Instead of the two relations $M, S : X \leftrightarrow X$ in (GY), the program (HS) uses two vectors $m, s : H \leftrightarrow \mathbf{1}$ to model a (hypergraph) matching and the set of hyperedges still to be treated, respectively. Consequently, instead of $e = edge(S)$ now $p = point(s)$ is used to select a new hyperedge. This leads, as simple point-wise considerations show, to $s \cap \overline{I^{\mathsf{T}};I;p}$ as relation-algebraic specification of the removal of all hyperedges incident to the selected one from the set of hyperedges modelled by s.

The use of a vector of type $H \leftrightarrow \mathbf{1}$ allows us to specify the (hypergraph) matching property relation-algebraically by a single formula – formula (1) in the list below. Informally it says that hyperedges from the matching modelled by m, which possess a common vertex are equal.

$$(1)\;\; m;m^{\mathsf{T}} \cap I^{\mathsf{T}};I \subseteq \mathsf{I} \qquad (2)\;\; \overline{s} \subseteq I^{\mathsf{T}};c \qquad (3)\;\; m;s^{\mathsf{T}} \cap I^{\mathsf{T}};I = \mathsf{O} \qquad (4)\;\; |c| \leq k \cdot |m|$$

Formula (2) again generalises the first formula of the post-condition $Post(I, k, c)$. The auxiliary formula (3) describes that no hyperedge from the matching modelled by m is incident to a hyperedge from the set modelled by s. Finally, formula (4) specifies the decisive property for proving the approximation bound k.

In the remainder of the section we use the conjunction of the formulae (1) to (4) as loop invariant $Inv(I, k, c, m, s)$ for proving the total correctness of the program (HS) with respect to the pre-condition $Pre(I, k)$ and the post-condition $Post(I, k, c)$. We proceed as in the case of (GY) and verify step-by-step the four proof obligations. The first one, saying that the loop invariant is established if the pre-condition holds, is given by the following lemma.

Lemma 6.1. *If $I : X \leftrightarrow H$ and $k \in \mathbb{N}$ satisfy $Pre(I, k)$, then $Inv(I, k, \mathsf{O}_{X\mathbf{1}}, \mathsf{O}, \mathsf{L})$ holds.*

Proof. Formula (4) of $Inv(I, k, c, m, s)$ holds because of cardinality axiom (C1); the proofs of the remaining formulae are trivial. □

Note that for the establishment of the invariant the pre-condition is not even necessary. As second step we now treat the second proof obligation and verify that the loop invariant is maintained by every execution of the body of the loop. In contrast to Lemma 5.2, in the corresponding Lemma 6.2 the pre-condition is used.

Lemma 6.2. *Assume $I : X \leftrightarrow H$, $k \in \mathbb{N}$, $c : X \leftrightarrow \mathbf{1}$ and $m, s : H \leftrightarrow \mathbf{1}$ such that $Pre(I, k)$ and $Inv(I, k, c, m, s)$ are satisfied and $s \neq \mathsf{O}$. Then we have $Inv(I, k, c \cup I;p, m \cup p, s \cap \overline{I^{\mathsf{T}};I;p})$, for all points $p : H \leftrightarrow \mathbf{1}$ with $p \subseteq s$.*

Proof. We show that the four formulae (1) to (4) of the loop invariant $Inv(I, k, c \cup I;p, m \cup p, s \cap \overline{I^{\mathsf{T}};I;p})$ hold for the new values of c, m and s. In case of the first formula we proceed as follows, where we use (1) and the injectivity of points in the third step and the assumption $p \subseteq s$ and (3) to get $m;p^{\mathsf{T}} \cap I^{\mathsf{T}};I \subseteq m;s^{\mathsf{T}} \cap I^{\mathsf{T}};I = \mathsf{O}$.

$$\begin{array}{rcl} (m \cup p);(m \cup p)^{\mathsf{T}} \cap I^{\mathsf{T}};I &=& (m;m^{\mathsf{T}} \cup m;p^{\mathsf{T}} \cup p;m^{\mathsf{T}} \cup p;p^{\mathsf{T}}) \cap I^{\mathsf{T}};I \\ &=& (m;m^{\mathsf{T}} \cap I^{\mathsf{T}};I) \cup (m;p^{\mathsf{T}} \cap I^{\mathsf{T}};I) \cup (p;m^{\mathsf{T}} \cap I^{\mathsf{T}};I) \cup (p;p^{\mathsf{T}} \cap I^{\mathsf{T}};I) \\ &\subseteq& \mathsf{I} \cup (m;p^{\mathsf{T}} \cap I^{\mathsf{T}};I) \cup (m;p^{\mathsf{T}} \cap I^{\mathsf{T}};I)^{\mathsf{T}} \cup \mathsf{I} \\ &=& \mathsf{I} \end{array}$$

To verify the claim for the second formula, we use (2) and calculate as follows:

$$\overline{s \cap \overline{I^\mathsf{T};I;p}} = \overline{s} \cup I^\mathsf{T};I;p \subseteq I^\mathsf{T};c \cup I^\mathsf{T};I;p = I^\mathsf{T};(c \cup I;p)$$

The claim for the third formula is shown by the following calculation:

$$\begin{array}{lll}
(m \cup p);(s \cap \overline{I^\mathsf{T};I;p})^\mathsf{T} \cap I^\mathsf{T};I &= (m;(s \cap \overline{I^\mathsf{T};I;p})^\mathsf{T} \cup p;(s \cap \overline{I^\mathsf{T};I;p})^\mathsf{T}) \cap I^\mathsf{T};I \\
&\subseteq (m;s^\mathsf{T} \cup p;(s \cap \overline{I^\mathsf{T};I;p})^\mathsf{T}) \cap I^\mathsf{T};I \\
&= (m;s^\mathsf{T} \cap I^\mathsf{T};I) \cup (p;(s \cap \overline{I^\mathsf{T};I;p})^\mathsf{T} \cap I^\mathsf{T};I) \\
&= p;(s \cap \overline{I^\mathsf{T};I;p})^\mathsf{T} \cap I^\mathsf{T};I & \text{by formula (3)} \\
&\subseteq p;\overline{I^\mathsf{T};I;p}^\mathsf{T} \cap I^\mathsf{T};I \\
&= p;\overline{p^\mathsf{T};I^\mathsf{T};I} \cap I^\mathsf{T};I \\
&\subseteq p;p^\mathsf{T};\overline{I^\mathsf{T};I} \cap I^\mathsf{T};I & \text{as } p^\mathsf{T} \text{ is univalent} \\
&\subseteq \overline{I^\mathsf{T};I} \cap I^\mathsf{T};I & \text{as } p \text{ is injective} \\
&= \mathsf{O}
\end{array}$$

It remains to show that the last formula of the loop invariant holds for the new values. Starting with (3) and using the first formula of the pre-condition, i.e., that the incidence relation I is surjective, we can first derive $p \cap m = \mathsf{O}$:

$$\begin{array}{lll}
m;s^\mathsf{T} \cap I^\mathsf{T};I = \mathsf{O} &\implies m;s^\mathsf{T} \cap \mathsf{I} = \mathsf{O} & \text{as } \mathsf{I} \subseteq I^\mathsf{T};I \\
&\iff m;s^\mathsf{T} \subseteq \overline{\mathsf{I}} & \\
&\iff \mathsf{I};s \subseteq \overline{m} & \text{Schröder rule} \\
&\iff s \cap m = \mathsf{O} & \\
&\implies p \cap m = \mathsf{O} & \text{as } p \subseteq s
\end{array}$$

This preparatory step allows us to prove the claim for the last formula by

$$|c \cup I;p| \leq |c| + |I;p| \leq k \cdot |m| + k \cdot |p| = k \cdot (|m| + |p|) = k \cdot |m \cup p|$$

that uses cardinality axiom (C3), then formula (4), the pre-condition $|I;p| \leq k$ and Lemma 4.3 and, finally, cardinality axioms (C3) and (C1) in combination with $p \cap m = \mathsf{O}$. □

Note that our second proof obligation is a variant of the second proof obligation that is usually used in assertion-based program verification. The latter is a direct consequence of the while-rule of the Hoare calculus and demands to verify the maintainance of the loop invariant by the body of the loop without assuming the pre-condition. Our version follows from the usual one by taking the pre-condition as part of the loop invariant. If programs do not change the input, which generally is assumed in program verification, then the maintainance of this part is obvious. Later we also will use a variant of the usual fourth proof obligation that additionally assumes the pre-condition to be true.

The third proof obligation shows again that the program terminates without an error. Since $s \neq \mathsf{O}$ each call of the partial operation *point* is defined. Hence it suffices to show that the loop terminates. This follows from the next lemma in combination with the finiteness of the set X.

Lemma 6.3. *Given $s : H \leftrightarrow \mathbf{1}$ such that $s \neq \mathsf{O}$, then $s \cap \overline{I^\mathsf{T};I;p} \subset s$, for all points $p : H \leftrightarrow \mathbf{1}$ with $p \subseteq s$.*

Proof. As we have $s \cap \overline{I^\mathsf{T};I;p} \subseteq s$, it remains to show $s \cap \overline{I^\mathsf{T};I;p} \neq s$. Analogous to the proof of (GY) we use contradiction and assume $s \cap \overline{I^\mathsf{T};I;p} = s$. Then we have $s \subseteq \overline{I^\mathsf{T};I;p}$, which is the same as $I^\mathsf{T};I;p \subseteq \overline{s}$. Now, the surjectivity of I yields

$$p = \mathsf{I};p \subseteq I^\mathsf{T};I;p \subseteq \overline{s}$$

such that, together with $p \subseteq s$, we obtain the contradiction $p = \mathsf{O}$. □

The fourth proof obligation is to verify that, after termination, the loop invariant implies the post-condition.

Lemma 6.4. *If $I : X \leftrightarrow H$, $k \in \mathbb{N}$, $c : X \leftrightarrow \mathbf{1}$ and $m, s : H \leftrightarrow \mathbf{1}$ satisfy $Inv(I, k, c, m, s)$ and $s = \mathsf{O}$, then $Post(I, k, c)$ holds.*

Proof. From formula (2) of $Inv(I, k, c, m, s)$ and $s = \mathsf{O}$ we get

$$\mathsf{L} = \overline{s} \subseteq I^{\mathsf{T}};c \,,$$

i.e., the first formula of $Post(I, k, c)$. To prove the second formula of $Post(I, k, c)$, let $d : X \leftrightarrow \mathbf{1}$ be an arbitrary vector such that $\mathsf{L} = I^{\mathsf{T}};d$.

First, we prove $I;p \cap d \neq \mathsf{O}$, for all points $p : H \leftrightarrow \mathbf{1}$. The proof is by contradiction. Assume that there exists a point $p : H \leftrightarrow \mathbf{1}$ with $I;p \cap d = \mathsf{O}$. Then a Schröder rule and $\overline{p} \subset \mathsf{L}$ (which follows from $p \neq \mathsf{O}$) allow to derive a contradiction to $\mathsf{L} = I^{\mathsf{T}};d$ as follows:

$$I;p \cap d = \mathsf{O} \iff I;p \subseteq \overline{d} \iff I^{\mathsf{T}};d \subseteq \overline{p} \implies I^{\mathsf{T}};d \neq \mathsf{L}$$

In combination with cardinality axiom (C1) and Lemma 4.3 we get as first auxiliary result that $|I;p \cap d| \geq 1 = |p|$, for all points $p : H \leftrightarrow \mathbf{1}$.

Next, we show that for all points p and q with $p \subseteq m$, $q \subseteq m$ and $p \neq q$ it follows $I;p \cap I;q = \mathsf{O}$. We start the proof with the calculation

$$\begin{array}{rcll} m \cap I^{\mathsf{T}};I;q & \subseteq & m \cap (\overline{m;m^{\mathsf{T}}} \cup \mathsf{I});q & \text{by formula (1)} \\ & = & m \cap (\overline{m;m^{\mathsf{T}}};q \cup q) & \\ & = & (m \cap \overline{m;m^{\mathsf{T}}};q) \cup (m \cap q) & \\ & = & (m \cap \overline{m;m^{\mathsf{T}}};q) \cup q & \text{as } q \subseteq m \\ & = & q & \text{as } \overline{m;m^{\mathsf{T}}};q \subseteq \overline{m}, \end{array}$$

where $\overline{m;m^{\mathsf{T}}};q \subseteq \overline{m}$ follows from $m;q^{\mathsf{T}} \subseteq m;m^{\mathsf{T}}$ (a consequence of $q \subseteq m$) and a Schröder rule. Via this estimation we can complete the proof:

$$\begin{array}{rcll} I;p \cap I;q & \subseteq & (I \cap I;q;p^{\mathsf{T}});(p \cap I^{\mathsf{T}};I;q) & \text{Dedekind rule} \\ & = & (I \cap I;q;p^{\mathsf{T}});(p \cap m \cap I^{\mathsf{T}};I;q) & \text{as } p \subseteq m \\ & \subseteq & (I \cap I;q;p^{\mathsf{T}});(p \cap q) & \text{see above} \\ & = & \mathsf{O} & \text{as } p \neq q \text{ implies } p \cap q = \mathsf{O} \end{array}$$

In combination with the cardinality axioms (C1) and (C3) and the finiteness of the set $\mathcal{P}_m$, hence, we get as second auxiliary result that

$$|\bigcup_{p \in \mathcal{P}_m} (I;p \cap d)| = \sum_{p \in \mathcal{P}_m} |I;p \cap d|.$$

With the help of these two auxiliary results, we now prove that $|m| \leq |d|$:

$$\begin{array}{rcll} |m| & = & |\bigcup_{p \in \mathcal{P}_m} p| & \text{by Lemma 3.1} \\ & \leq & \sum_{p \in \mathcal{P}_m} |p| & \mathcal{P}_m \text{ finite, cardinality axiom (C3)} \\ & \leq & \sum_{p \in \mathcal{P}_m} |I;p \cap d| & \text{first auxiliary result} \\ & = & |\bigcup_{p \in \mathcal{P}_m} (I;p \cap d)| & \text{second auxiliary result} \\ & = & |(\bigcup_{p \in \mathcal{P}_m} I;p) \cap d)| & \\ & \leq & |d| & \text{monotonicity of cardinality} \end{array}$$

From this and formula (4) of $Inv(I, k, c, m, s)$ we get $|c| \leq k \cdot |m| \leq k \cdot |d|$. This concludes the proof of the second formula of $Post(I, k, c)$. □

In the program (HS) the auxiliary variable m is again only used for the formal assertion-based correctness proof. If we use the result of [4] and remove m without affecting correctness, we get the following result:

Theorem 6.1. *For all relations $I : X \leftrightarrow H$ with X being non-empty and finite and $k \in \mathbb{N}$ we have: If in the relational program* (HS) *all assignments to the auxiliary variable m are removed, then the resulting program is totally correct with respect to the pre-condition $Pre(I, k)$ and the post-condition $Post(I, k, c)$.* □

7. Relational Approximation of Maximum Independent Sets

Using the adjacency-relation model of Section 5 and assertions, we now formally verify a relational program for approximating maximum independent sets for undirected (loop-free) graphs. A formal verification of a similar (non-relational) program can be found in [4]. In contrast to [4], however, we do not use an auxiliary variable for proving the desired approximation bound, and calculate purely relation-algebraically. Furthermore, we show that our program immediately generalises to the computation of independent sets in hypergraphs if the incidence-relation model of Section 6 is used.

As before, we assume an undirected graph $g = (X, E)$, where the set of vertices X is non-empty and finite. As described in Section 5, we model the graph by a symmetric and irreflexive adjacency relation $R : X \leftrightarrow X$. Thus, we have the same pre-condition as for the program (GY) of Section 5 and, as the approximation bound will depend on the degrees of the vertices, additionally that the maximum degree of g is $k \in \mathbb{N}$. This yields the conjunction of the following three formulae as pre-condition $Pre(R, k)$:

$$R \subseteq \overline{\mathsf{I}} \qquad R = R^{\mathsf{T}} \qquad k = \max\{|R;p| \mid p : X \leftrightarrow \mathbf{1} \text{ point}\}$$

An *independent set* (or *stable set*) of g is a set of vertices S such that $\{x, y\} \notin E$, for all $x, y \in S$. A vector $s : X \leftrightarrow \mathbf{1}$ models an independent set with respect to the adjacency relation R iff $R;s \subseteq \overline{s}$. We will show that our program has approximation bound $\frac{1}{k+1}$, i.e., the cardinality of the computed independent set is guaranteed to be at least $\frac{1}{k+1}$-times the cardinality of any independent set; hence, also of any maximum independent set. So, the post-condition $Post(R, k, s)$ is the conjunction of the following two formulae:

$$R;s \subseteq \overline{s} \qquad \forall\, t : X \leftrightarrow \mathbf{1} \bullet R;t \subseteq \overline{t} \Rightarrow |t| \leq |s| \cdot (k + 1)$$

As we will show in a moment, the following relational program (which is based on Wei's approximation algorithm described in [28]) is correct with respect to the pre-condition $Pre(R, k)$ and the post-condition $Post(R, k, s)$:

$$\begin{array}{l} s, v := \mathsf{O}, \mathsf{O}_{X\mathbf{1}}; \\ \textbf{while } v \neq \mathsf{L} \textbf{ do} \\ \quad \textbf{let } p = point(\overline{v}); \\ \quad s, v := s \cup p, v \cup p \cup R;p \textbf{ od} \end{array} \tag{W}$$

A little reflection shows that the initialisation of v leads to the typing $s, v, p : X \leftrightarrow \mathbf{1}$ and also to $X \leftrightarrow \mathbf{1}$ as type of the constant L in the guard of the loop. In the program (W) the vector v is used to collect the vertices contained in the present independent set, that is modelled by the vector s, and also their neighbours.

We use the conjunction of the following two formulae as loop invariant $Inv(R, k, s, v)$ for proving that the program (W) is totally correct with respect to the pre-condition $Pre(R, k)$ and the post-condition $Post(R, k, s)$:

$$(1)\ R;s \subseteq \overline{s} \qquad (2)\ R;s \cup s = v$$

Formula (1) is part of the post-condition and formula (2) is an auxiliary formula saying that the vector v models the union of the set modelled by the vector s with the neighbours of this set.

To reach our goal, we prove the four proof obligations of assertion-based verification with respect to the above specified pre- and post-condition, as usual. We start with the establishment of the loop invariant by the initialisation of the variables.

Lemma 7.1. *If $R : X \leftrightarrow X$ and $k \in \mathbb{N}$ satisfy $Pre(R, k)$, then $Inv(R, k, \mathsf{O}, \mathsf{O}_{X\mathbf{1}})$ holds.* □

We omit the trivial proof. Note that for the establishment of the invariant the pre-condition is again not necessary. The next lemma says that the loop invariant is maintained by each execution of the body of the loop. As in the case of Lemma 6.2 we need the pre-condition for this result.

Lemma 7.2. *Assume $R : X \leftrightarrow X$, $s, v : X \leftrightarrow \mathbf{1}$ and $k \in \mathbb{N}$ such that $Pre(R, k)$ and $Inv(R, k, s, v)$ are satisfied and $v \neq \mathsf{L}$. Then we have $Inv(R, k, s \cup p, v \cup p \cup R;p)$, for all points $p : X \leftrightarrow \mathbf{1}$ with $p \subseteq \overline{v}$.*

Proof. First, we verify that the first formula of the loop invariant holds for the new value of s, i.e., that

$$R;(s \cup p) \subseteq \overline{s \cup p}\,.$$

Since $\overline{s \cup p} = \overline{s} \cap \overline{p}$, it is sufficient to show that $R;s \subseteq \overline{s} \cap \overline{p}$ and $R;p \subseteq \overline{s} \cap \overline{p}$. Because of (2) we have $R;s \subseteq v$ and with the assumption $p \subseteq \overline{v}$ we have $R;s \subseteq \overline{p}$. Due to this and (1) we have $R;s \subseteq \overline{s} \cap \overline{p}$. Furthermore, we obtain the equivalences

$$R;s \subseteq \overline{p} \iff R^{\mathsf{T}};p \subseteq \overline{s} \iff R;p \subseteq \overline{s}$$

using one of the Schröder rules in the first step, and the second formula of the pre-condition in the second step. Since the point p is injective and R is irreflexive due to the first formula of the pre-condition, we obtain

$$p;p^{\mathsf{T}} \subseteq \mathsf{I} \implies p;p^{\mathsf{T}} \subseteq \overline{R} \iff R;p \subseteq \overline{p}\,,$$

using one of the Schröder rules in the last step; whereby $R;p \subseteq \overline{s} \cap \overline{p}$ holds.

The maintenance of the second formula of the loop invariant is similarly easy to prove, since by (2) we have

$$R;(s \cup p) \cup (s \cup p) = R;s \cup R;p \cup s \cup p = v \cup p \cup R;p\,. \qquad \square$$

For the third proof obligation we verify the error-free termination of the program (W). A consequence of the guard of the loop is that each call of the partial operation *point* is defined. For this reason and the assumed finiteness of the set X, it suffices to show that the loop terminates, i.e., that v is strictly enlarged by each execution of the loop's body.

Lemma 7.3. *Given $v : X \leftrightarrow \mathbf{1}$ with $v \neq \mathsf{L}$, we have $v \subset v \cup p \cup R;p$, for all points $p : X \leftrightarrow \mathbf{1}$ with $p \subseteq \overline{v}$.*

Proof. Since $v \subseteq v \cup p \cup R;p$ holds, we show $v \neq v \cup p \cup R;p$, again using contradiction. We start the proof with the assumption $v = v \cup p \cup R;p$ and calculate as follows:

$$v = v \cup p \cup R;p \iff p \cup R;p \subseteq v \implies p \subseteq v.$$

The last inclusion and the assumption $p \subseteq \overline{v}$ imply $p = \mathsf{O}$, which contradicts the fact that points are non-empty. $\square$

Finally, we consider the last proof obligation, i.e., that if $v = \mathsf{L}$ holds then the loop invariant implies the post-condition. In contrast to Section 5 and Section 6 we also need the pre-condition, in particular the maximum-degree condition, for the proof. Since neither the variable R nor the variable k are changed by the program, the addition of $\mathit{Pre}(R, k)$ as assumption is eligible.

Lemma 7.4. *If $R : X \leftrightarrow X$, $k \in \mathbb{N}$ and $s, v : X \leftrightarrow \mathbf{1}$ satisfy $\mathit{Pre}(R, k)$ and $\mathit{Inv}(R, k, s, v)$, and $v = \mathsf{L}$, then $\mathit{Post}(R, k, s)$ holds.*

Proof. The first formula of $\mathit{Post}(R, k, s)$ holds since it correspond to formula (1) of the loop invariant $\mathit{Inv}(R, k, s, v)$.

To verify the second formula of $\mathit{Post}(R, k, s)$, let $t : X \leftrightarrow \mathbf{1}$ be an arbitrary vector such that $R;t \subseteq \overline{t}$.

$$\begin{array}{lll}
|t| & \leq & |\mathsf{L}_{X\mathbf{1}}| & \text{typing } t : X \leftrightarrow \mathbf{1}\text{, monotonicity of cardinality} \\
& = & |v| & \text{since } v = \mathsf{L}_{X\mathbf{1}} \\
& = & |R;s \cup s| & \text{by formula (2) of } \mathit{Inv}(R, k, s, v) \\
& \leq & |R;s| + |s| & \text{by cardinality axiom (C3)} \\
& = & |R;\bigcup_{p \in \mathcal{P}_s} p| + |s| & \text{by Lemma 3.1} \\
& = & |s| + |\bigcup_{p \in \mathcal{P}_s} R;p| & \\
& \leq & |s| + \sum_{p \in \mathcal{P}_s} |R;p| & \mathcal{P}_s \text{ is finite, cardinality axiom (C3)} \\
& \leq & |s| + \sum_{p \in \mathcal{P}_s} k & \text{second formula of } \mathit{Pre}(R, k) \\
& = & |s| + k \cdot |s| & \text{by Lemma 4.4} \\
& = & (k + 1) \cdot |s| &
\end{array}$$

$\square$

In sum we have proven the four proof obligations of the assertion-based program verification method, such that the following theorem holds.

Theorem 7.1. *For all relations $R : X \leftrightarrow X$, where X is finite and non-empty, and $k \in \mathbb{N}$ the relational program* (W) *is totally correct with respect to the pre-condition* $\mathrm{Pre}(R, k)$ *and the post-condition* $\mathrm{Post}(R, k, s)$. □

We conclude this section by briefly explaining how a slight modification of the program (W) can be used to approximate maximum independent sets in hypergraphs. To this end, let $h = (X, H)$ be a hypergraph and let $I : X \leftrightarrow H$ be its incidence relation. An *independent set* of h is a set of vertices Y such that no hyperedge contains more than one vertex of Y. The latter means that $x \neq y$ implies $x\,\overline{I;I^\mathsf{T}}\,y$, for all $x, y \in Y$. If we define an undirected graph $g = (X, E)$ with $E := \{\{x, y\} \mid x, y \in X \wedge x\,(\overline{\mathsf{I}} \cap I;I^\mathsf{T})\,y\}$ as set of edges, i.e., with adjacency relation $\overline{\mathsf{I}} \cap I;I^\mathsf{T} : X \leftrightarrow X$, then a little reflection shows for all sets of vertices Y that Y is an independent set of h iff it is an independent set of g. So, if in the program (W) the relation $R : X \leftrightarrow X$ is replaced by the relation $\overline{\mathsf{I}} \cap I;I^\mathsf{T} : X \leftrightarrow X$, then the resulting program approximates an independent set of h with approximation bound $\frac{1}{k+1}$, where $k = \max\{|I;I^\mathsf{T};p| \mid p : X \leftrightarrow \mathbf{1} \text{ point}\}$. This equation states that the maximum degree of a vertex of h is k.

8. Relational Approximation of Maximum Cuts

In this section we present yet another relational program; this time a program that approximates maximum cuts in undirected (loop-free) graphs. As before we assume an undirected graph $g = (X, E)$ such that the set of vertices X is non-empty and finite, and we model the graph by a symmetric and irreflexive adjacency relation $R : X \leftrightarrow X$. Thus, the pre-condition $\mathit{Pre}(R)$ is again the conjunction of the following two formulae:

$$R \subseteq \overline{\mathsf{I}} \qquad\qquad R = R^\mathsf{T}$$

A *cut* of g is just a partition of the set of vertices into two disjoint subsets. Obviously, a vector $s : X \leftrightarrow \mathbf{1}$ and its complement always model such a partition with respect to the adjacency relation R. In the following we present a relational approximation algorithm for maximum cuts and show that our program maximises the number of edges between s and $\overline{s}$, i.e., the cardinality $|R \cap (s;\overline{s}^\mathsf{T} \cup \overline{s};s^\mathsf{T})|$, in such a way that we get an approximation bound of $\frac{1}{2}$. This desired bound is specified by the following formula that we take as the post-condition $\mathit{Post}(R, s)$:

$$\forall\, c : X \leftrightarrow \mathbf{1} \bullet |R \cap (c;\overline{c}^\mathsf{T} \cup \overline{c};c^\mathsf{T})| \leq 2 \cdot |R \cap (s;\overline{s}^\mathsf{T} \cup \overline{s};s^\mathsf{T})|$$

This means that the number of edges between the computed cut and its complement is guaranteed to be at least half of the number of edges between any cut and its complement, hence, also at least half of the number of edges between any maximum cut and its complement.

In the remainder of this section we show that the following relational program (MC) is totally correct with respect to the pre-condition $\mathit{Pre}(R)$ and the post-condition $\mathit{Post}(R, s)$. In contrast to the programs we have presented so far, (MC) also uses a conditional and the cardinality operation within its guard.

$$\begin{array}{l} v, s, t := \mathsf{L}_{\mathbf{x1}}, \mathsf{O}, \mathsf{O}; \\ \textbf{while } v \neq \mathsf{O} \textbf{ do} \\ \quad \textbf{let } p = \mathit{point}(v); \\ \quad \textbf{if } |R;p \cap s| < |R;p \cap t| \textbf{ then } v, s := v \cap \overline{p}, s \cup p \\ \qquad\qquad\qquad\qquad\quad \textbf{else } \; v, t := v \cap \overline{p}, t \cup p \textbf{ od} \end{array} \qquad \text{(MC)}$$

Due to the initialisation of v we get v, s, t and p of type $X \leftrightarrow \mathbf{1}$. Informally, the program (MC) partitions the vertices of X by checking for each of them the number of neighbours in the sets modelled by the current vectors s and t. Each vertex is added to the set with the smaller number of neighbours. This approach is, for example, mentioned in [20] and is a specialisation of an approximation algorithm published in [22] for the maximum cut problem.

To show correctness of the program (MC) we use the conjunction of the following three formulae as loop invariant $\mathit{Inv}(R, v, s, t)$:

$$(1)\;\; s \cap t = \mathsf{O} \qquad (2)\;\; s \cup t = \overline{v} \qquad (3)\;\; |R \cap (s;s^\mathsf{T} \cup t;t^\mathsf{T})| \leq |R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})|$$

Formulae (1) and (2) specify that s and t form a partition of the vector $\overline{v}$. Formula (3) states that the number of edges connecting vertices of the set modelled by s or of the set modelled by t is smaller than the number of edges between these two sets. For this reason (3) is an auxiliary formula used to prove the desired approximation bound.

As in the previous three sections we prove the four proof obligations with respect to the above specified pre- and post-condition. The establishment of the loop invariant by the initialisation of the variables is formulated in the following lemma; we omit the trivial proof that again does not use the pre-condition.

Lemma 8.1. *If $R : X \leftrightarrow X$ satisfies Pre(R), then Inv($R, \mathsf{L}_{X\mathbf{1}}, \mathsf{O}, \mathsf{O}$) holds.* □

In the next lemma we treat the maintenance of the loop invariant. In contrast to the already presented programs of Section 5 to 7 in the program (MC) a conditional occurs in the body of the loop. Due to this, the second proof obligation for (MC) slightly differs from the previous ones. The new values for s and t, respectively, depend on the guard. Consequently, we have to prove that the loop invariant is maintained in both cases. Thus, the next lemma claims two implications. We need the pre-condition for the proofs.

Lemma 8.2. *Assume $R : X \leftrightarrow X$ and $v, s, t : X \leftrightarrow \mathbf{1}$ such that Pre(R) and Inv(R, v, s, t) are satisfied and $v \neq \mathsf{O}$. Then the following two implications hold for all points $p : X \leftrightarrow \mathbf{1}$ with $p \subseteq v$:*

(a) If $|R;p \cap s| < |R;p \cap t|$, then we have Inv($R, v \cap \overline{p}, s \cup p, t$).

(b) If $|R;p \cap t| \leq |R;p \cap s|$, then we have Inv($R, v \cap \overline{p}, s, t \cup p$).

Proof. First of all, we state that $p \subseteq v = \overline{s \cup t} = \overline{s} \cap \overline{t}$ holds by the assumption $p \subseteq v$ and formula (2) of Inv(R, v, s, t). Thus, we have $p \subseteq \overline{s}$ and $p \subseteq \overline{t}$. Next, we have to show that in both cases the three formulae of the loop invariant hold for the new values of v, s and t. We only do this for (a) since the proof of (b) can be done analogously.

So, assume $|R;p \cap s| < |R;p \cap t|$. The proof for the first formula of the loop invariant follows by (1) and $p \subseteq \overline{t}$:

$$(s \cup p) \cap t = (s \cap t) \cup (p \cap t) = p \cap t = \mathsf{O} .$$

The claim that the second formula holds for the new values is shown as follows, where we use (2) for the second equation:

$$(s \cup p) \cup t = (s \cup t) \cup p = \overline{v} \cup p = \overline{v \cap \overline{p}}$$

To prove the claim for the third formula of the loop invariant we need the fact that $|R;p \cap s| < |R;p \cap t|$ holds. Using that p is a point and the first formula of the pre-condition we have

$$R \cap p;p^{\mathsf{T}} \subseteq R \cap \mathsf{I} = \mathsf{O}$$

as first auxiliary result. Using again that p is a point and both Schröder rules we obtain

$$p \subseteq \overline{t} \implies p;\mathsf{L} \subseteq \overline{t} \iff p^{\mathsf{T}};t \subseteq \mathsf{O} \implies t;p^{\mathsf{T}};t \subseteq \mathsf{O} \implies t;p^{\mathsf{T}};t \subseteq \overline{p} \iff p;t^{\mathsf{T}} \subseteq \overline{t;p^{\mathsf{T}}} \iff p;t^{\mathsf{T}} \cap t;p^{\mathsf{T}} = \mathsf{O} .$$

From this and the cardinality axioms (C1) and (C3) we get

$$|(R \cap p;t^{\mathsf{T}}) \cup (R \cap t;p^{\mathsf{T}})| = |R \cap p;t^{\mathsf{T}}| + |R \cap t;p^{\mathsf{T}}|$$

as second auxiliary result. Similar to the above calculation we can show that $s;t^{\mathsf{T}} \cap t;p^{\mathsf{T}} = \mathsf{O}$, and $s;t^{\mathsf{T}} \cap p;t^{\mathsf{T}} = (s \cap p);t^{\mathsf{T}} = \mathsf{O}$, which follows from the injectivity of t^{T} and $s \cap p = \mathsf{O}$. This yields $(s;t^{\mathsf{T}} \cup t;s^{\mathsf{T}}) \cap (p;t^{\mathsf{T}} \cup t;p^{\mathsf{T}}) = \mathsf{O}$. From the cardinality axioms (C1) and (C3) we get

$$|(R \cap (s;t^{\mathsf{T}} \cup t;s^{\mathsf{T}})) \cup (R \cap p;t^{\mathsf{T}}) \cup (R \cap t;p^{\mathsf{T}})| = |R \cap (s;t^{\mathsf{T}} \cup t;s^{\mathsf{T}})| + |(R \cap p;t^{\mathsf{T}}) \cup (R \cap t;p^{\mathsf{T}})|$$

as third and last auxiliary result. After these preparatory proofs, the following calculation shows the desired claim:

$$
\begin{array}{lll}
|R \cap ((s \cup p);(s \cup p)^\mathsf{T} \cup t;t^\mathsf{T})| &= |(R \cap (s;s^\mathsf{T} \cup t;t^\mathsf{T} \cup s;p^\mathsf{T} \cup p;s^\mathsf{T} \cup p;p^\mathsf{T})| & \\
&= |(R \cap (s;s^\mathsf{T} \cup t;t^\mathsf{T})) \cup (R \cap s;p^\mathsf{T}) \cup (R \cap p;s^\mathsf{T})| & \text{first auxiliary result} \\
&\leq |R \cap (s;s^\mathsf{T} \cup t;t^\mathsf{T})| + |R \cap s;p^\mathsf{T}| + |R \cap p;s^\mathsf{T}| & \text{by cardinality axiom (C3)} \\
&\leq |R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})| + |R \cap s;p^\mathsf{T}| + |R \cap p;s^\mathsf{T}| & \text{by formula (3)} \\
&= |R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})| + |R \cap p;s^\mathsf{T}| + |R \cap p;s^\mathsf{T}| & \text{by cardinality axiom (C2), } R = R^\mathsf{T} \\
&= |R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})| + |R;p \cap s| + |R;p \cap s| & \text{by Lemma 4.2} \\
&< |R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})| + |R;p \cap t| + |R;p \cap t| & \text{as } |R;p \cap s| < |R;p \cap t| \\
&= |R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})| + |R \cap p;t^\mathsf{T}| + |R \cap p;t^\mathsf{T}| & \text{by Lemma 4.2} \\
&= |R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})| + |R \cap p;t^\mathsf{T}| + |R \cap t;p^\mathsf{T}| & \text{by cardinality axiom (C2), } R = R^\mathsf{T} \\
&= |R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})| + |(R \cap p;t^\mathsf{T}) \cup (R \cap t;p^\mathsf{T})| & \text{second auxiliary result} \\
&= |(R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})) \cup (R \cap p;t^\mathsf{T}) \cup (R \cap t;p^\mathsf{T})| & \text{third auxiliary result} \\
&= |R \cap ((s \cup p);t^\mathsf{T} \cup t;(s \cup p)^\mathsf{T})| &
\end{array}
$$

Note that Lemma 4.2 is applicable since $R = R^\mathsf{T}$ due to the pre-condition, p is injective and s and t are univalent as a consequence of $s, t : X \leftrightarrow \mathbf{1}$. □

To verify the third proof obligation it again suffices to show that the loop terminates, since each call of the partial operation *point* is defined because of the guard $v \neq \mathsf{O}$. Since we assume the set of vertices X to be finite, the termination of the program (MC) follows from the next lemma.

Lemma 8.3. *Given $v : X \leftrightarrow \mathbf{1}$ with $v \neq \mathsf{O}$, we have $v \cap \overline{p} \subset v$, for all points $p : X \leftrightarrow \mathbf{1}$ with $p \subseteq v$.* □

We omit the trivial proof. The next lemma states the last proof obligation, i.e., that after terminating the loop invariant implies the post-condition.

Lemma 8.4. *If $R : X \leftrightarrow X$ and $v, s, t : X \leftrightarrow \mathbf{1}$ satisfy $Inv(R, v, s, t)$ and $v = \mathsf{O}$, then $Post(R, s)$ holds.*

Proof. We consider an arbitrary vector $c : X \leftrightarrow \mathbf{1}$. Since $v = \mathsf{O}$ and $Inv(R, v, s, t)$ is true, we have $s \cup t = \overline{v} = \mathsf{L}$ and $s \cap t = \mathsf{O}$ and thereby $t = \overline{s}$. Now, the following calculation concludes the proof:

$$
\begin{array}{lll}
|R \cap (c;\overline{c}^\mathsf{T} \cup \overline{c};c^\mathsf{T})| &\leq |R| & \text{monotonicity of cardinality} \\
&= |R \cap \mathsf{L}| & \\
&= |R \cap (s \cup t);(s \cup t)^\mathsf{T}| & \text{by formula (2) of } Inv(R, v, s, t),\ v = \mathsf{O} \\
&= |(R \cap (s;s^\mathsf{T} \cup t;t^\mathsf{T})) \cup (R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})| & \\
&\leq |R \cap (s;s^\mathsf{T} \cup t;t^\mathsf{T})| + |R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})| & \text{by cardinality axiom } (C3) \\
&\leq |R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})| + |R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})| & \text{by formula (3) of } Inv(R, v, s, t) \\
&= 2 \cdot |R \cap (s;t^\mathsf{T} \cup t;s^\mathsf{T})| & \\
&= 2 \cdot |R \cap (s;\overline{s}^\mathsf{T} \cup \overline{s};s^\mathsf{T})| & \text{as } t = \overline{s}
\end{array}
$$

□

By the previous four lemmas we have shown the four proof obligations, which are sufficient for the total correctness of the program (MC). Summarising these results we have the following fact:

Theorem 8.1. *For all relations $R : X \leftrightarrow X$, where X is finite and non-empty, the relational program* (MC) *is totally correct with respect to the pre-condition $Pre(R)$ and the post-condition $Post(R, s)$.* □

To the best of our knowledge the maximum cut problem is not considered for hypergraphs in the literature, which is in contrast to the vertex cover problem and the independent set problem. A reason may be that there are too many different possibilities to translate the notion of a cut from undirected graphs to hypergraphs.

9. Concluding Remarks

In the present paper we have discussed points, atoms and edges as specific relations. They allow us to model vertices, directed edges and undirected edges of graphs, i.e., their essential constituents, and to reason about them in a relation-algebraic manner. We have presented some fundamental properties of these relations. For those concerning their cardinality we applied Kawahara's general characterisation of the cardinality of relations as axiomatic basis. To show how the concepts can be applied in practice, we have formulated four well-known approximation algorithms as relational programs and have shown that they are totally correct with respect to a problem specification given by relation-algebraic formulae. For the latter we used the assertion-based method and relation-algebraic calculations to verify its proof obligations.

Our investigations have always been supported by several tools. In particular we have used the system RelView for testing loop invariants as well as the automated theorem prover Prover9 (see [18]) and the proof assistant tool Isabelle/HOL (see [19]) for proof support. For the replication of proofs in this context we followed the approach presented in [8, 9]. Furthermore, we have made use of the counterexample generator Mace4 (see again [18]) for identifying missing assumptions in cases we had difficulties to prove a conjecture and for falsifying statements. For instance, Mace4 has shown that the Tarski rule is necessary to obtain the decisive result that in relation algebras each point is an atom in the lattice-theoretic sense among the vectors of its type. If the Tarski rule is not required as an axiom of relation algebra, as it is done in equational axiomatisations (see e.g., [17, 26, 27]), then the four relations

$$\mathsf{O} := \emptyset \qquad R := \{(a,a)\} \qquad S := \{(b,b)\} \qquad \mathsf{L} := \{(a,a),(b,b)\}$$

of type $\{a,b\} \leftrightarrow \{a,b\}$ constitute a counterexample, as shown by Mace4. With the identity relation I defined as L they satisfy the axioms (1) to (3) of Section 2, but not the axiom (4). Furthermore, all relations are vectors and R, S and L are points. But L is not an atom in the lattice $(\{\mathsf{O}, R, S, \mathsf{L}\}, \cup, \cap)$ of vectors.

In case of set-theoretic relations the points of type $X \leftrightarrow Y$ are precisely the lattice-theoretic atoms among the vectors of type $X \leftrightarrow Y$, i.e., for Proposition 2.4.5 of [24] also the reverse implication holds. And also for Theorem 3.1 and Theorem 3.2 hold the reverse directions in case of set-theoretic relations. However, all this cannot be shown by relation-algebraic means since these facts are not valid in all relation algebras. Mace4 delivered the four relations

$$\mathsf{O} := \emptyset \qquad \mathsf{I} := \{(a,a),(b,b)\} \qquad \bar{\mathsf{I}} := \{(a,b),(b,a)\} \qquad \mathsf{L} := \{(a,a),(b,b),(a,b),(b,a)\}$$

of type $\{a,b\} \leftrightarrow \{a,b\}$ as counterexample. Both O and L are the only vectors, L is the only lattice-theoretic atom of the lattice $(\{\mathsf{O}, \mathsf{L}\}, \cup, \cap)$ of vectors, but it is not a point. Furthermore, the lattice-theoretic atoms of the lattice $(\{\mathsf{O}, \mathsf{I}, \bar{\mathsf{I}}, \mathsf{L}\}, \cup, \cap)$ are I and $\bar{\mathsf{I}}$, but they are not atoms in the sense of Definition 3.1. This shows that the reverse direction of Theorem 3.1 does not hold in this relation algebra. Since O, I, $\bar{\mathsf{I}}$ and L are symmetric, this is also a counterexample for the reverse direction of Theorem 3.2.

As future work we plan to investigate the axiomatisation of the cardinality operation further. In doing so, we hope to find useful laws for important classes of relations and for relation-algebraic expressions that frequently appear in applications. We also want to explore our relational approach to approximation algorithms by treating further examples to understand the pros and cons of the approach. To be able to support all this by theorem-proving tools we plan to extend existing libraries for relation algebra such as [1] for Isabelle/HOL and [21] for Coq, by the axioms of and facts about cardinalities.

Acknowledgement. We are indepted to Y. Kawahara, M. Winter and W. Guttmann for reading a draft version of the paper and for giving valuable hints for its improvement. We also thank the three referees for their helpful comments. R. Berghammer and I. Stucke gratefully acknowledge support by the DAAD. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

[1] Armstrong, A., Foster, S., Struth, G., Weber, T.: Relation algebra. Archive of Formal Proofs, 2014. `http://afp.sf.net/entries/Relation_Algebra.shtml`

[2] Berghammer, R.: Combining relational calculus and the Dijkstra-Gries method for deriving relational programs. Information Sciences 119, 155-171 (1999)

[3] Berghammer, R., Hoffmann, T.: Relational depth-first-search with applications. Information Sciences 139, 167-186 (2001)

[4] Berghammer, R., Müller-Olm, M.: Formal development and verification of approximation algorithms using auxiliary variables. In: Bruynooghe, M. (ed.): Logic Based Program Development and Transformation. LNCS, vol. 3018, pp. 59-74. Springer (2004)

[5] Berghammer, R.: Applying relation algebra and RelView to solve problems on orders and lattices. Acta Informatica 45, 211-236 (2008)
[6] Berghammer, R., Winter, M.: Embedding mappings and splittings with applications. Acta Informatica 47, 77-110 (2010)
[7] Berghammer, R., Struth, G.: On automated program construction and verification. In: Bolduc, C., Desharnais, J., Ktari, B. (eds.) Mathematics of Program Construction. LNCS, vol. 6120, pp. 22-41. Springer (2010)
[8] Berghammer, R., Höfner, P., Stucke, I.: Automated verification of relational while-programs. In: Höfner, P., Jipsen, P., Kahl, W., Müller, M.E. (eds.) Relational and Algebraic Methods in Computer Science. LNCS, vol. 8248, pp. 309-326. Springer (2014)
[9] Berghammer, R., Höfner, P., Stucke, I.: Tool-based verification of a relational vertex coloring program. In: Kahl, W., Winter, M., Oliveira, J.N. (eds.) Relational and Algebraic Methods in Computer Science. LNCS, vol. 9348, pp. 275-292. Springer (2015)
[10] Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to algorithms. The MIT Press (1990)
[11] Francez, N.: Program verification. Addison-Wesley (1992)
[12] Furusawa, H.: Algebraic formalisations of fuzzy relations and their representation theorems. Ph.D. thesis, Department of Informatics, Kyushu University (1998)
[13] Gries, D.: The science of programming. Springer (1981)
[14] Höfner, P., Struth, G.: On automating the calculus of relations. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) Automated Reasoning. LNAI, vol. 5195, pp. 50-66. Springer (2008)
[15] Kawahara, Y.: On the cardinality of relations. In: Schmidt, R.A. (ed.): Relations and Kleene Algebra in Computer Science. LNCS, vol. 4136, pp. 251-265. Springer (2006)
[16] Maddux, R.: Relation algebras. In: Brink, C., Kahl, W., Schmidt, G. (eds.): Relational Methods in Computer Science. Advances in Computing Science, pp. 22-38. Springer (1997)
[17] Maddux, R.: Relation algebras. Studies in Logic and the Foundations of Mathematics, vol. 150. Elsevier (2006)
[18] McCune, W.W.: Prover9 and Mace4. `http://www.cs.unm.edu/~mccune/prover9`
[19] Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL: A proof assistant for higher-order logic. LNCS, vol. 2283. Springer (2002)
[20] Papadimitriou C.H., Yannakakis M.: Optimization, approximation and complexity classes. Journal of Computer and System Sciences 43, 425-440 (1991)
[21] Pous, D.: Relation algebra and KAT in Coq. `http://perso.ens-lyon.fr/damien.pous/ra/`
[22] Sahni S., Gonzalez T.: *P*-complete approximation problems. Journal of the ACM 23, 555-565 (1976)
[23] Schmidt, G., Ströhlein, T.: Relation algebras: Concept of points and representability. Discrete Mathematics 34, 83-97 (1985)
[24] Schmidt, G., Ströhlein, T.: Relations and graphs, Discrete mathematics for computer scientists, EATCS Monographs on Theoretical Computer Science. Springer (1993)
[25] Schmidt, G.: Relational mathematics. Encyclopedia of Mathematics and its Applications, vol. 132. Cambridge University Press (2010)
[26] Tarski, A.: On the calculus of relations. Journal of Symbolic Logic 6, 73-89 (1941)
[27] Tarski, A., Givant, S.: A formalization of set theory without variables. Colloquium Publications 41, American Mathematical Society (1987)
[28] Wei, V.K.: A lower bound for the stability number of a simple graph. Bell Lab. Tech. Memor. 81-11217-9 (1981)
[29] RelView-homepage: `http://www.informatik.uni-kiel.de/~progsys/relview/`