Journal of Logical and Algebraic Methods in Programming 85 (2016) 269–286

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Cardinality of relations and relational approximation algorithms

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

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

ARTICLE INFO

Article history:
Received 13 July 2015
Received in revised form 30 November 2015
Accepted 4 December 2015
Available online 8 December 2015

Keywords:
Relation algebra
Cardinality operation
Approximation algorithm
Assertion-based program verification

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.

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 of frequently clarifying the proof structure, reducing the danger of wrong proof steps and opening the possibility for proof mechanisation, e.g., by automated theorem provers or proof assistant tools. Examples for the latter can be found in [7–9,14] 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].

* Corresponding author. Tel.: +49 431 880 7272; fax: +49 431 880 7613.
E-mail address: rub@informatik.uni-kiel.de (R. Berghammer).

http://dx.doi.org/10.1016/j.jlamp.2015.12.001
2352-2208/

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, p. 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 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].