Journal of Logical and Algebraic Methods in Programming 83 (2014) 194–211

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Multirelations with infinite computations

Walter Guttmann

Department of Computer Science and Software Engineering, University of Canterbury, New Zealand

A R T I C L E I N F O

Article history:
Available online 12 February 2014

Keywords:
Approximation order
Median
Multirelations
Program semantics
Relations
RelView
Sequential computations

A B S T R A C T

Multirelations model computations with both angelic and demonic non-determinism. We extend multirelations to represent finite and infinite computations independently. We derive an approximation order for multirelations assuming only that the endless loop is its least element and that the lattice operations are isotone. We use relations, relation algebra and RelView for representing and calculating with multirelations and for finding the approximation order.

1. Introduction

Computation models that support both angelic and demonic choice are useful for modelling contracts between agents, interaction, games and protocols [1,20]. Finite computations in such models can be represented by multirelations [22–24]. However, multirelations have no means for representing computations that fail to terminate independently of finite computations; this is similar to relational computation models such as those of [17]. The first goal of this paper is to extend multirelations to represent finite and infinite computations independently.

The basic idea to achieve this is the same as for relations, namely to extend the state space. The main challenge is to provide a suitable approximation order, which is needed to define the semantics of recursion. In some relational models, the Egli–Milner order is used for approximation. But already in the relational setting it is not clear which approximation order to use when more precise models are considered; these require variations of the Egli–Milner order [11,14]. It is even less obvious how to generalise the Egli–Milner order to multirelations. Thus the second goal of this paper is to find means of defining approximation which are independent of particular computation models.

We apply relational methods to achieve the two goals. First, we express the basic definitions, operations and properties of multirelations in terms of relations. This involves transforming logical expressions to relational formulas and algebraic calculations with these formulas. Second, we derive relational programs for investigating the approximation order in RelView [3]. From particular instances of approximation we distill a relational definition that is suitable for multirelations; its properties are again shown by applying algebra.

The main contributions of this paper are as follows:

- We introduce strict multirelations in Section 5.2 as a reaction to the failure of having an approximation order for all up-closed multirelations.

E-mail address: walter.guttmann@canterbury.ac.nz.

http://dx.doi.org/10.1016/j.jlap.2014.02.008
2352-2208/

- We give an approximation order for strict, up-closed multirelations, which is expressed by Theorem 15 in terms of relations. It turns out to be the same as an order in a previous study of pointed distributive lattices [10]. We thus obtain many useful properties including representations of least fixpoints.
- In Theorem 18 we show that our approximation order is the $\subseteq$-least partial order which has the endless loop as least element and isotone lattice operations. This is a new characterisation with very weak assumptions expected to hold in many computation models.
- We give definitions, operations and properties of multirelations, multirelations with a special state, the endless loop, up-closed multirelations and strict multirelations in terms of relations. We implement programs using such multirelations in RelView.

Section 2 recalls heterogeneous relations. Section 3 expresses multirelations in terms of relations and shows how they represent computations. Section 4 extends the state space to represent infinite computations. Section 5 applies RelView to derive an approximation order, shows that it is suitable for multirelations, expresses it in terms of relations, characterises it based on weak assumptions, and instantiates previous results for the representation of fixpoints.

Isotone predicate transformers are isomorphic to up-closed multirelations [22,1,23,16]. Many basic properties of multirelations shown in Section 3 are known from existing research on these models; here we give a relational development. Sections 4 and 5 extend these models by representing finite and infinite computations independently. For example, a non-deterministic choice between the endless loop and the program that does not change the state will be different from both of these programs. Existing approaches that make this distinction do not support both angelic and demonic choice [18,21,15,8].

2. Relations

In this section we recall basic definitions, operations and properties of heterogeneous relations [28,27,26]; see also [4,25,2].

2.1. Basic operations

Given sets A and B, a relation R of type $A \leftrightarrow B$ is a subset of the Cartesian product $A \times B$; we write $R : A \leftrightarrow B$ and abbreviate $(x, y) \in R$ by R_{xy}. The relations of type $A \leftrightarrow B$ form a complete Boolean algebra with union $\cup$, arbitrary union $\bigcup$, intersection $\cap$, arbitrary intersection $\bigcap$, complement $\overline{\ }$ and partial order $\subseteq$ bounded by the empty relation $\mathsf{O} = \emptyset$ and the universal relation $\mathsf{T} = A \times B$. The composition of two relations $Q : A \leftrightarrow B$ and $R : B \leftrightarrow C$ is the relation $QR : A \leftrightarrow C$ defined by $(QR)_{xz} \Leftrightarrow \exists y \in B : Q_{xy} \wedge R_{yz}$. The identity relation $\mathsf{I} : A \leftrightarrow A$ is defined by $\mathsf{I}_{xy} \Leftrightarrow x = y$. The converse of a relation $R : A \leftrightarrow B$ is the relation $R^{\smile} : B \leftrightarrow A$ defined by $R^{\smile}{}_{xy} \Leftrightarrow R_{yx}$. Union, intersection and composition are defined only if the types of the involved relations match as indicated above; we tacitly assume this in the remainder of this paper. Composition has higher precedence than union and intersection.

Union, intersection, composition and converse are $\subseteq$-isotone; complement is $\subseteq$-antitone. We furthermore use the following properties:

- Composition distributes over $\bigcup$ and converse distributes over $\bigcup$, $\bigcap$ and $\overline{\ }$.
- $(QR)^{\smile} = R^{\smile} Q^{\smile}$ and $R^{\smile\smile} = R$ and $\mathsf{O}^{\smile} = \mathsf{O}$ and $\mathsf{I}^{\smile} = \mathsf{I}$ and $\mathsf{T}^{\smile} = \mathsf{T}$.
- $\mathsf{O}R = \mathsf{O}$ and $R\mathsf{O} = \mathsf{O}$ and $\mathsf{I}R = R = R\mathsf{I}$ and $\mathsf{T}\mathsf{T} = \mathsf{T}$.
- $PQ \subseteq R \Leftrightarrow P^{\smile}\overline{R} \subseteq \overline{Q} \Leftrightarrow \overline{R}Q^{\smile} \subseteq \overline{P}$; these are the Schröder equivalences.
- $\mathsf{T}R\mathsf{T} = \mathsf{T}$ if $R \neq \mathsf{O}$; this is the Tarski rule.
- $R = R^{\smile}$ if $R \subseteq \mathsf{I}$.

A relation R is total if $R\mathsf{T} = \mathsf{T}$, univalent if $R^{\smile}R \subseteq \mathsf{I}$, surjective if $\mathsf{T}R = \mathsf{T}$, injective if $RR^{\smile} \subseteq \mathsf{I}$, a mapping if R is total and univalent, bijective if R is injective and surjective, a vector if $R\mathsf{T} = R$, reflexive if $\mathsf{I} \subseteq R$, transitive if $RR \subseteq R$, antisymmetric if $R \cap R^{\smile} \subseteq \mathsf{I}$, a preorder if R is reflexive and transitive, and a partial order if R is an antisymmetric preorder. A vector $R : A \leftrightarrow B$ represents a set which contains those elements of A that are related by R to every element of B. We use the following properties:

- $(\bigcap_{i \in I} Q_i)R = \bigcap_{i \in I}(Q_i R)$ if R is injective and $\overline{QR} = \overline{Q}R$ if R is bijective.
- $PQ \subseteq R \Leftrightarrow Q \subseteq P^{\smile}R$ and $QP^{\smile} \subseteq R \Leftrightarrow Q \subseteq RP$ if P is bijective.
- $(P\mathsf{T} \cap Q)R = P\mathsf{T} \cap QR$ and $Q(R \cap \mathsf{T}P) = QR \cap \mathsf{T}P$ and $(Q \cap \mathsf{T}P^{\smile})R = Q(P\mathsf{T} \cap R)$.
- Vectors are closed under $\bigcup$, $\bigcap$ and $\overline{\ }$.