Available online at www.sciencedirect.com
ScienceDirect
Science of Computer Programming 65 (2007) 140–158
Science of Computer Programming
www.elsevier.com/locate/scico

Modelling angelic and demonic nondeterminism with multirelations

C.E. Martin[a,*], S.A. Curtis[a], I. Rewitzky[b]

[a] *Department of Computing, Oxford Brookes University, United Kingdom*
[b] *Department of Mathematics, University of Stellenbosch, South Africa*

Received 31 January 2005; received in revised form 15 January 2006; accepted 30 January 2006
Available online 28 November 2006

Abstract

This paper presents an introduction to a calculus of binary multirelations, which can model both angelic and demonic kinds of non-determinism. The isomorphism between up-closed multirelations and monotonic predicate transformers allows a different view of program transformation, and program transformation calculations using multirelations are easier to perform in some circumstances. Multirelations are illustrated by modelling both kinds of nondeterministic behaviour in games and resource-sharing protocols.

Keywords: Multirelation; Predicate transformer; Agent; Strongest postcondition; Angelic nondeterminism; Demonic nondeterminism; Resource sharing

1. Introduction

For many years the realm of total functions was considered to be the natural framework in which to reason about functional programs [6]. The limitation of this framework is that it can only be used to describe deterministic programs: those that deliver only one output for each input. More recently, the development of the relational calculus for program derivation [7] has allowed programmers to reason about programs together with their specifications, which may be nondeterministic: they may offer a choice of outputs for each input. Specifications expressed in this calculus can be manipulated and refined into functions that can be translated into an appropriate functional programming language. We now propose to go one step further, since relations can be used to describe only angelic or demonic nondeterminism, but not both.

Angelic nondeterminism occurs when the choice is made by an 'angel': it is assumed that the angel will choose the best possible outcome. Demonic nondeterminism occurs when the choice is made by a 'demon': no assumption can be made about the choice made by the demon, so one must be prepared for the worst possible outcome. It is well known that both these kinds of behaviour can be described in the domain of monotonic predicate transformers [2,14], but this is usually associated with the derivation of imperative, rather than functional programs.

* Corresponding author.
E-mail address: cemartin@brookes.ac.uk (C.E. Martin).

0167-6423/$ - see front matter
doi:10.1016/j.scico.2006.01.007

An equivalent relational model that has recently been proposed consists of up-closed multirelations [16]. Relational equivalents of predicate transformers have been introduced in the past (see [12] for example), but Rewitzky has extended this equivalence to show that any monotonic function over a boolean algebra has an alternative representation as a multirelation.

This paper is devoted to showing how specifications involving both angelic and demonic nondeterminism can be expressed and manipulated as multirelations. Such specifications can be refined until they contain only one kind of nondeterministic choice. Then they can be translated into programs in which those remaining choices are made interactively by an external agent at run time.

This is not the first attempt to introduce angelic and demonic nondeterminism into a calculus for the derivation of functional programs. Ward [18] developed a refinement calculus for this purpose, in a similar style to that of [2,14], and with a corresponding predicate transformer semantics. However, we argue that multirelations will provide a better model for several reasons. First, one of the beauties of the relational calculus (for examples of its use see [3,7,17]) is that specifications are simply relations: there is no separate command language. As such, they can be manipulated by all the familiar operations on relations, such as composition and inverse, as well as those developed specifically for the derivation of functional programs. In contrast, in the refinement calculus of both [2,14] and [18] the command language is distinct from its predicate transformer semantics. So specifications can only be manipulated using laws that were previously derived from the semantics. This places a burden on the developer to memorise all the laws, and can complicate the reasoning about operations as fundamental as composition, which can be expressed simply for predicate transformers, but much less so for specifications in the command language. One solution to this problem could be to represent specifications directly as predicate transformers, but this might be counter-intuitive because they model programs in reverse, mapping postconditions to preconditions. In contrast, multirelations can be used to relate inputs, or initial states, to outputs, or final states, so we propose to blur the distinction between multirelations and the specifications they represent.

As an illustration of our theory, we will give examples demonstrating how multirelations can be used to specify and manipulate some games and resource-sharing protocols. For example, in a two-player game the choices of our player could be modelled by angelic nondeterminism, and those of our opponent by demonic nondeterminism. The resulting specification would allow us to reason about the game as a whole, and then derive an implementation that achieves some goal, such as an optimal strategy for the player. The implementation would then consist of a computerised player, who would play according to the strategy against a human opponent who could make demonic choices in an attempt to defeat the machine.

The rest of this paper is organised as follows. Section 2 introduces multirelations and shows how they can be used to model contracts between two agents. Section 3 discusses refinement, together with the associated issues of correctness and feasibility. Two applications are considered in Section 4, which is followed by a description of the isomorphism between the categories of multirelations and predicate transformers in Section 5. Some concluding remarks are given in Section 6.

2. Nondeterministic specifications

2.1. Agents

We are primarily interested in specifying systems that contain both angelic and demonic choice. One way to think of such specifications is as a contract [2] between two agents, both of whom are free to make various choices. One of the agents represents our interests, but the other may not. In this context, the angelic choices are interpreted as those made by our agent, since we assume that he will always make the choice that results in the best outcome for us. Conversely, the demonic choices are those made by the other agent, and since we have no control over them we must be prepared for all possibilities, including the worst.

For example, the specification of a simple vending machine from a customer's viewpoint might contain an angelic choice of coin insertion and chocolate bar selection, followed by a demonic choice of coins returned by the machine as change. This can be viewed as a contract where our agent is the customer and the other agent is the machine designer who can choose how the machine gives out change.

Specifications like this, featuring both angelic and demonic nondeterminism, have been modelled previously using monotonic predicate transformers (see [2,14] for example), but until now there has been no equivalent relational