Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 319 (2015) 271–287

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Sequential Algorithms for Unbounded Nondeterminism

J. Laird[1]

Department of Computer Science, University of Bath, UK

Abstract

We give extensional and intensional characterizations of higher-order functional programs with unbounded nondeterminism: as stable and monotone functions between the biorders of states of ordered concrete data structures, and as *sequential algorithms* (states of an exponential ocds) which compute them. Our fundamental result establishes that these representations are equivalent, by showing how to construct a unique sequential algorithm which computes a given stable and monotone function.

We illustrate by defining a denotational semantics for a functional language with countable nondeterminism ("fair PCF"), with an interpretation of fixpoints which allows this to be proved to be computationally adequate. We observe that our model contains functions which cannot be computed in fair PCF, by identifying a further property of the definable elements, and so show that it is not fully abstract.

Keywords: Sequential Algorithms, Nondeterminism, Fairness, Biorders

1 Introduction

This paper develops a model of higher-order computation with unbounded nondeterminism. In this setting we may write programs which will always return a value but may take an unbounded number of steps to do so, corresponding to the notion of *fairness* [6]. A major challenge for capturing such programs is that they do not correspond to continuous functions, in general. In domain theory, this may be resolved by weakening the continuity properties required (e.g. to ω_1-continuity [1]), although this admits many undefinable functions and leaves fewer principles with which to reason about program behaviour. A more *intensional* representation of programs (for example, as strategies in a games model) offers the possibility of studying unbounded nondeterminism in computation more directly, although traditional representations of strategies as collections of finite sequences of moves are insufficient to capture the distinction between infinite interactions and finite, but unbounded ones [7].

[1] Research supported by UK EPSRC grant EP/K037633/1

http://dx.doi.org/10.1016/j.entcs.2015.12.017
1571-0661/© 2015 The Author. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

We take an approach which relates extensional and intensional representations of programs with unbounded nondeterminism: our main result is an equivalence between stable and monotone functions and sequential algorithms on ordered concrete data structures. We show that these equivalent representations may be used to interpret a simple functional programming language with unbounded nondeterminism (fair PCF). We show that this model contains elements which are not definable as terms, leading to a failure of full abstraction and suggesting how it could be further refined.

1.1 Related Work

Our model is based on an intensional description of stable and monotone functions on *biorders* generated from ordered concrete data structures. Biorders, which combine some intensional information, in the form of the stable order, with the extensional (Scott) order, were introduced by Berry [2]. In previous work, the author has shown that stable and continuous functions on biorders with a (extensionally) greatest element are (Milner-Vuillemin) sequential, and used them to construct models of sequential languages such as the lazy λ-calculus [12], as well as λ-calculi with nondeterminism [11]. However, although these models technically carry information about program behaviour, they do not do so in a transparent way.

Concrete data structures were introduced by Kahn and Plotkin [9], as part of a definition of sequentiality for higher-order functionals, but the more intensional notion of *sequential algorithm* (a state of a "function-space" CDS) introduced by Berry and Curien [3] offers an appealing model of computation in its own right. On the one hand, concrete data structures correspond to a *positional* form of games, and sequential algorithms to positional strategies (see e.g. [8]). On the other, sequential algorithms may be related to purely extensional models: in the deterministic case, Cartwright, Curien and Felleisen [4] have established that they compute, and are equivalent to "observably sequential" functions; the author has given a more abstract representation of the latter as *bistable* functions on bistable biorders [12,10].

To interpret sequential, but nondeterministic programs (corresponding to stable and monotone functions on Berry-style biorders, which are sequential, but not *strongly sequential*) as sequential algorithms, we abandon the consistency condition on states (that any cell may be filled with at most one value). However, this also requires an ordering on cells and values (corresponding to game positions), to reflect the fact that (for example) any program which may diverge in response to a given argument may still diverge in response to an argument with a wider range of behaviours. This notion of an ordered concrete data structure was introduced in [13] in which stable and continuous functions were shown to correspond to *finite-branching* sequential algorithms. Interestingly, the stable order on non-deterministic sequential algorithms had been described by Roscoe [15] on processes in CSP — the "strong order" — in work approximately contemporaneous with the discoveries of bidomains and of sequential algorithms. Here, we extend the correspondence between stable functions and sequential algorithms to unbounded nondeterminism. This requires a new notion of ordinal-indexed interaction, to distinguish computations which are

Download English Version:

https://daneshyari.com/en/article/421634

Download Persian Version:

https://daneshyari.com/article/421634

Daneshyari.com