Abstract

Building on Kahn and Plotkin’s theory of concrete data structures and sequential functions, Berry and Curien defined an intensional model of sequential algorithms between concrete data structures. In this paper we report on an attempt to develop a similar intensional model of concurrent computation. We present a notion of parallel algorithm between concrete data structures, together with suitable application and currying operations. We define an intensional strictness ordering on parallel algorithms, with respect to which application is well behaved (at first order types). We define the input-output function computed by a parallel algorithm, and we show that every parallel algorithm computes a continuous function. Thus, a parallel algorithm may be viewed as a continuous function together with a parallel computation strategy. In contrast, a Berry-Curien sequential algorithm may be viewed as a sequential function together with a sequential computation strategy. The intensional strictness ordering on parallel algorithms corresponds to the pointwise ordering on the functions they compute, in the same sense that the set inclusion ordering used by Berry and Curien on sequential algorithms corresponds to the stable ordering on the functions they compute.

We believe that the ideas and results presented here constitute a first step towards a fuller understanding of the intensional semantics of parallelism, even though the model presented here is not yet general enough to provide a satisfactory account of higher order algorithms, and lacks a notion of composition for algorithms. We present some ideas for overcoming these deficiencies, and some directions for further research.

This paper is an expanded and improved version of [7].

1 Introduction

The search for a satisfactory syntactic and semantic account of sequential computation, in particular the desire to achieve full abstraction, has led to a considerable body of research. In the classic paper [4], Plotkin showed that under its standard interpretation the programming language PCF is inherently sequential, and that its standard continuous functions semantic model is not fully abstract because the model contains inherently parallel functions (such as parallel-or) that cannot be defined in PCF. The continuous functions model is, however, fully abstract for a parallel version of PCF obtained by including a parallel conditional primitive. A substantial body of work has been directed at obtaining a truly sequential model for the original PCF with a suitably restricted notion of function [15].

Milner [1], Sazonov [3], and Vuillemin [10] proposed notions of sequential functions; however, their constructions make essential use of the number of arguments to a function but do not adequately reflect the internal structure of these arguments, so that their notions of sequentiality are not general enough. Kahn and Plotkin [12] defined concrete data structures, or CDSs, together with