Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 347 (2005) 441 – 464

Theoretical
Computer Science

www.elsevier.com/locate/tcs

# A structural approach to reversible computation

Samson Abramsky

*Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, UK*

Received 4 May 2004; received in revised form 12 July 2005; accepted 12 July 2005

Communicated by J. Tiuryn

**Abstract**

Reversibility is a key issue in the interface between computation and physics, and of growing importance as miniaturization progresses towards its physical limits. Most foundational work on reversible computing to date has focussed on simulations of low-level machine models. By contrast, we develop a more structural approach. We show how high-level functional programs can be mapped *compositionally* (i.e. in a syntax-directed fashion) into a simple kind of automata which are immediately seen to be reversible. The size of the automaton is linear in the size of the functional term. In mathematical terms, we are building a concrete *model* of functional computation. This construction stems directly from ideas arising in Geometry of Interaction and Linear Logic—but can be understood without any knowledge of these topics. In fact, it serves as an excellent introduction to them. At the same time, an interesting logical delineation between reversible and irreversible forms of computation emerges from our analysis.

© 2005 Elsevier B.V. All rights reserved.

*Keywords:* Reversible computation; Linear combinatory algebra; Term-rewriting; Automata; Geometry of Interaction

## 1. Introduction

The importance of reversibility in computation, for both foundational and, in the medium term, for practical reasons, is by now well established. We quote from the excellent summary in the introduction to the recent paper by Buhrman et al. [19]:

---

*E-mail address:* samson@dcs.ed.ac.uk.

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2005.07.002

> *Reversible computation*: Landauer [41] has demonstrated that it is only the "logically irreversible" operations in a physical computer that necessarily dissipate energy by generating a corresponding amount of entropy for every bit of information that gets irreversibly erased; the logically reversible operations can in principle be performed dissipation-free. Currently, computations are commonly irreversible, even though the physical devices that execute them are fundamentally reversible. At the basic level, however, matter is governed by classical mechanics and quantum mechanics, which are reversible. This contrast is only possible at the cost of efficiency loss by generating thermal entropy into the environment. With computational device technology rapidly approaching the elementary particle level it has been argued many times that this effect gains in significance to the extent that efficient operation (or operation at all) of future computers requires them to be reversible …The mismatch of computing organization and reality will express itself in friction: computers will dissipate a lot of heat unless their mode of operation becomes reversible, possibly quantum mechanical.

The previous approaches of which we are aware (e.g. [43,17,18]) proceed by showing that some standard, low-level, irreversible computational model such as Turing machines can be simulated by a reversible version of the same model. Our approach is more "structural". We firstly define a simple model of computation which is directly reversible in a very strong sense—every automaton $\mathcal{A}$ in our model has a "dual" automaton $\mathcal{A}^{\mathsf{op}}$, defined quite trivially from $\mathcal{A}$, whose computations are exactly the time-reversals of the computations of $\mathcal{A}$. We then establish a connection to models of *functional computation*. We will show that our model gives rise to a *combinatory algebra* [33], and derive universality as an easy consequence. This method of establishing universality has potential significance for the important issue of how to *program* reversible computations. To quote from [19] again:

> Currently, almost no algorithms and other programs are designed according to reversible principles …To write reversible programs by hand is unnatural and difficult. The natural way is to compile irreversible programs to reversible ones.

Our approach can be seen as providing a simple, *compositional* (i.e. "syntax-directed") compilation from high-level functional programs into a reversible model of computation. This offers a novel perspective on reversible computing.

Our approach also has conceptual interest in that our constructions, while quite concrete, are based directly on ideas stemming from Linear Logic and Geometry of Interaction [25–29,45,21,22,15], and developed in previous work by the present author and a number of colleagues [5,6,2,3,9,10,4]. Our work here can be seen as a concrete manifestation of these more abstract and foundational developments. However, no knowledge of Linear Logic or Geometry of Interaction is required to read the present paper. In fact, it might serve as an introduction to these topics, from a very concrete point of view. At the same time, an interesting logical delineation between reversible and irreversible forms of computation emerges from our analysis.

*Related work*: Geometry of Interaction (GoI) was initiated by Girard in a sequence of papers [26–28], and extensively developed by Danos et al. see e.g. [45,21,22,15]. In particular, Danos and Regnier developed a computational view of GoI. In [22] they gave a compositional translation of the $\lambda$-calculus into a form of reversible abstract machine. We also

Download English Version:

https://daneshyari.com/en/article/10334729

Download Persian Version:

https://daneshyari.com/article/10334729

Daneshyari.com