Journal of Automata, Languages and Combinatorics **23** (2018) 1–3, 165 – 188

COMPUTATIONAL REQUIREMENTS HIDDEN BEHIND ENCODINGS

Maia Hoeberechts

Ocean Networks Canada & Department of Computer Science
University of Victoria, Victoria, British Columbia, Canada
maiah@uvic.ca

ABSTRACT

In formal language theory, we are accustomed to defining, discussing, and developing formal models of computation, often without considering their potential applicability as practical computing devices. In the real world, however, we expect our computers to do things for us: display images, solve differential equations, spell check documents, and so on.

It is frequently assumed that as long as an effective encoding of a problem in a machine's input/output alphabet exists and providing that the machine can compute a desired function, then that machine is capable of solving that problem. This approach hides requirements which should be explicitly included as part of the computation. In this paper, the gap between machine models and their encodings and one's ability to use those models to solve problems is explored, namely with respect to the computational requirements hidden behind encodings. The analysis provides some small steps in the direction of a model of computation which more closely resembles real-world computation.

Firstly, computational requirements for using machine descriptions for problem solving are analysed in detail. Definitions of computing capability for individual machines are introduced. Differences resulting from varied input alphabets are discussed, in addition to implications for universal computation, and consequences of using language acceptors and function calculators. Secondly, the distinction between formal descriptions and their interpretations in the problem domain is considered. The idea that machine descriptions are labels for functions is introduced. The trade-off between the succinctness of a description and its readiness for use as a computing device is discussed. Finally, some conclusions regarding the computing power required for solving problems are presented.

Keywords: formal models of computation, Turing machines as problem solving devices, machine descriptions as labels for functions

1. Introduction

1.1. Background and Motivation

What is the relationship between formal language theory and practical computation? Is theoretical computer science purely a formalist discipline, with no metaphysical

The ideas in this paper were inspired during my graduate studies by Helmut Jürgensen.