Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 322 (2016) 197–210

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Unbounded Recursion and Non-size-increasing Functions

S. Mazzanti[1]

Dipartimento di Culture del Progetto
Università Iuav di Venezia
Fondamenta delle Terese 2206,
30123 Venezia, Italy

Abstract

We investigate the computing power of function algebras defined by means of unbounded recursion on notation. We introduce two function algebras which contain respectively the regressive logspace computable functions and the non-size-increasing logspace computable functions. However, such algebras are unlikely to be contained in the set of logspace computable functions because this is equivalent to $\mathbf{L} = \mathbf{P}$. Finally, we introduce a function algebra based on simultaneous recursion on notation for the non-size-increasing functions computable in polynomial time and linear space.

Keywords: recursion on notation, logspace computable function, polynomial time computable function.

1 Introduction

Since the introduction of the Grzegorczyk's hierarchy, algebras of subrecursive functions have been defined by means of bounded recursion schemes, and from the early 1990s also by means of predicative recursion schemes.

However, both bounded and predicative recursion do not correspond to programming constructs of real-world programming languages, inhibiting the benefits of a major integration of complexity theory and programming language theory [4,5].

Function algebras mostly characterize function complexity classes containing polynomial growth functions, and either bounded recursion or predicative recursion is needed to avoid functions with exponential growth. On the other hand, computational complexity is mainly concerned with decision problems represented as languages or more generally as relations over a suitable data type.

In [7], the relations on natural numbers decidable in linear space and the languages decidable in logarithmic space have been characterized by means of two

[1] Email: mazzanti@iuav.it

http://dx.doi.org/10.1016/j.entcs.2016.03.014
1571-0661/

function algebras containing very simple base functions and closed with respect to substitution and unbounded simultaneous recursion.

One algebra contains a set of regressive functions computable in linear space; the other contains a set of non-size-increasing functions computable in logarithmic space.[2]

However, they contain all the characteristic functions of the decision problems solvable in linear and logarithmic space, respectively.

These results are instances of a general fact: if the base functions of a function algebra do not increase too rapidly, unbounded recursion can be used to describe interesting complexity classes. For example, both regressive and non-size-increasing functions are closed with respect to substitution and recursion on notation.

In [2], Predecessor Machines (Register Machines without increment instructions) have been introduced as a computation model for regressive functions. In particular, it has been shown that the total functions computable by Predecessor Machines are the regressive functions of Grzegorczyk's class $\mathcal{E}_2$.

Type systems for the non-size-increasing polynomial time functions have been extensively studied, see e.g. [3]. Such systems constitute an alternative to predicative recursion which rules out not only recursive definitions leading to exponential growth, but also natural algorithms computing non-size-increasing functions.

Therefore, it seems that regressive and non-size-increasing functions could play a remarkable role in computational complexity.

Let $\mathbf{R}(\mathbf{F})$ be the set of number-theoretic functions defined as the closure with respect to substitution and recursion on notation of a list $\mathbf{F}$ of basic functions, the constant functions and the projection functions.

Algebra $\mathbf{R}(P)$ where P is the predecessor function, has been studied in [8,9] following [7]. Algebra $\mathbf{R}(P)$ is a set of regressive logspace computable functions which contains all the sharply bounded logspace computable functions.[3] Moreover, $\mathbf{R}(P)$ is the set of functions computable by Regressive Machines, which are a polynomial time version of structured Predecessor Machines [2].

In this paper we study new function algebras which extend algebra $\mathbf{R}(P)$ in order to shed some light on some open questions suggested in [9]. For instance, we do not know whether $\mathbf{R}(P)$ equals the set of regressive logspace computable functions.

In Section 3, we consider the algebra $\mathbf{R}(bs_0, bs_1)$ where $s_i(x) = 2x + i$ and $bs_i(x, y) = s_i(x)$ if $s_i(x) \leq y$ and $bs_i(x, y) = x$ otherwise. Algebra $\mathbf{R}(bs_0, bs_1)$ contains the regressive logspace computable functions, but we give strong evidence that $\mathbf{R}(bs_0, bs_1)$ is unlikely to be contained in the set $\mathbf{FL}$ of logspace computable functions because $\mathbf{R}(bs_0, bs_1) \subseteq \mathbf{FL}$ is equivalent to $\mathbf{L} = \mathbf{P}$. In the same way, we show that if $\mathbf{R}(P)$ contained all the regressive logspace computable functions then $\mathbf{L} = \mathbf{P}$ would be true.

[2] The values of a regressive function are bounded by the maximum of its arguments and a constant whereas the size of the values of a non-size-increasing function is bounded by the maximum of the sizes of its arguments and a constant.

[3] A function is sharply bounded iff its values can be stored in logarithmic space.