Annals of Pure and Applied Logic 167 (2016) 335–369

Contents lists available at ScienceDirect

Annals of Pure and Applied Logic

www.elsevier.com/locate/apal

Cobham recursive set functions

Arnold Beckmann [a,1,2], Sam Buss [b,*,1,3], Sy-David Friedman [c,1,4], Moritz Müller [c,1], Neil Thapen [d,1,5]

[a] *Department of Computer Science, Swansea University, Swansea SA2 8PP, UK*
[b] *Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112, USA*
[c] *Kurt Gödel Research Center, University of Vienna, A-1090 Vienna, Austria*
[d] *Institute of Mathematics, Academy of Sciences of the Czech Republic, Prague, Czech Republic*

A R T I C L E   I N F O

Article history:
Received 8 August 2015
Received in revised form 12 December 2015
Accepted 14 December 2015
Available online 28 December 2015

MSC:
03D15
03D20
03E99
68Q15

Keywords:
Set function
Polynomial time
Cobham Recursion
Smash function
Hereditarily finite
Rudimentary function

A B S T R A C T

This paper introduces the Cobham Recursive Set Functions (CRSF) as a version of polynomial time computable functions on general sets, based on a limited (bounded) form of $\in$-recursion. This is inspired by Cobham's classic definition of polynomial time functions based on limited recursion on notation. We introduce a new set composition function, and a new smash function for sets which allows polynomial increases in the ranks and in the cardinalities of transitive closures. We bootstrap CRSF, prove closure under (unbounded) replacement, and prove that any CRSF function is embeddable into a smash term. When restricted to natural encodings of binary strings as hereditarily finite sets, the CRSF functions define precisely the polynomial time computable functions on binary strings. Prior work of Beckmann, Buss and Friedman and of Arai introduced set functions based on safe-normal recursion in the sense of Bellantoni-Cook. We prove an equivalence between our class CRSF and a variant of Arai's predicatively computable set functions.

* Corresponding author.
E-mail addresses: a.beckmann@swansea.ac.uk (A. Beckmann), sbuss@ucsd.edu (S. Buss), sdf@logic.univie.ac.at (S.-D. Friedman), moritz.mueller@univie.ac.at (M. Müller), thapen@math.cas.cz (N. Thapen).

[1] The initial results of this work were obtained at the Kurt Gödel Research Center at the University of Vienna, with funding provided by Short Visit Grant 4932 from the Humanities Research Networking Programmes of the European Science Foundation (ESF).
[2] Supported in part by the Simons Foundation (grant 208717 to S. Buss).
[3] Supported in part by NSF grants DMS-1101228 and CCR-1213151, and by the Simons Foundation, award 306202.
[4] Supported in part by Austrian Science Fund (FWF), Project Number I-1238 on "Definability and Computability".
[5] Supported in part by grant P202/12/G061 of GA ČR and grant RVO:67985840 of Academy of Sciences of the Czech Republic.

http://dx.doi.org/10.1016/j.apal.2015.12.005
0168-0072/© 2015 Published by Elsevier B.V.

1. Introduction

This paper presents a definition of “Cobham Recursive Set Functions” which is designed to be a version of polynomial time computability based on computation on sets. This represents an alternate (or, a competing) approach to the recent work of Beckmann, Buss and S. Friedman [3], who defined the Safe Recursive Set Functions (SRSF), and to the work of Arai [1], who introduced the Predicatively Computable Set Functions (PCSF). SRSF and PCSF were based on Bellantoni-Cook style safe-normal recursion, but using $\in$-recursion for computation on sets in place of recursion on strings. Both [3] and [1] were motivated by the desire to find analogues of polynomial time native to sets. For hereditarily finite sets, the class SRSF turned out to correspond to functions computable by Turing machines which use alternating exponential time with polynomially many alternations. For infinite sets, SRSF corresponds to definability at a polynomial level in the relativized L-hierarchy. For infinite binary strings of length ω, it corresponds to computation by infinite polynomial time Turing machines, which use time less than ω^n for some $n > 0$. The class PCSF, on the other hand, does correspond to polynomial time functions when restricted to appropriate encodings of strings by hereditarily finite sets. No characterization of PCSF for non-hereditarily finite sets is known.

In this paper, we give a different approach to polynomial time computability on sets, using an analogue of Cobham limited recursion on notation, inspired by one of the original definitions of polynomial time computable functions [7]. The class P (sometimes denoted FP) of polynomial time computable functions on binary strings can be defined as the smallest class of functions that (a) contains as initial functions the constant empty string ϵ, the two successor functions $s \mapsto s0$ and $s \mapsto s1$ and the projection functions, and (b) is closed under composition and limited recursion on notation. If g, h_0 and h_1 are functions, and p is a polynomial, then the following function f is said to be defined by *limited recursion on notation*:

$$
\begin{aligned}
f(\vec{a}, \epsilon) &= g(\vec{a}) \\
f(\vec{a}, s0) &= h_0(\vec{a}, f(\vec{a}, s), s) \\
f(\vec{a}, s1) &= h_1(\vec{a}, f(\vec{a}, s), s)
\end{aligned} \tag{1}
$$

provided that

$$
|f(a_1, \dots, a_n, s)| \;\leq\; p(|a_1|, \dots, |a_n|, |s|) \tag{2}
$$

always holds. Here $\vec{a}$ and s are (vectors of) binary strings; and $|a|$ denotes the length of the binary string a.

A slightly different version of limited recursion uses the smash ($\#$) function instead (cf. [10] and [6]). For this, the smash function is defined as $a\#b = 0^{|a|\cdot|b|}$ so that $a\#b$ is the string of 0’s with length $|a\#b|$ equal the product of the lengths of the binary strings a and b. The smash function can be included in the small set of initial functions, and then the bound (2) can be replaced by the condition that

$$
|f(\vec{a}, s)| \leq |k(\vec{a}, s)| \tag{3}
$$

where k is a function already known to be in P. In this version, f is said to be defined by *limited recursion on notation from g, h_0, h_1 and k.*

Section 3 defines the Cobham Recursive Set Functions (CRSF) via an analogue of the definition of polynomial time functions with limited recursion. CRSF uses $\in$-recursion instead of recursion on notation. In $\in$-recursion, the value of $f(x)$, for x a set, is defined in terms of the set of values $f(y)$ for all $y \in x$. This means that the recursive computation of $f(x)$ requires computing $f(y)$ for all y in the transitive closure, $\mathrm{tc}(x)$, of x. The depth of the recursion is bounded by the rank, $\mathrm{rank}(x)$, of x. Of course, the cardinality of the transitive closure of x, $|\mathrm{tc}(x)|$, can be substantially larger than the cardinality of the rank of x. The computational complexity of $f(x)$ is thus bounded by both the rank of x and by $|\mathrm{tc}(x)|$; however, the