Hereditarily Total Sequential Functionals of Finite Type

Bruce M. Kapron
University of Victoria

Abstract

We give a direct proof that the hereditarily total functionals definable in van Oosten's model for sequential functionals of finite type coincide with the Kleene/Kreisel continuous functionals.

1 Introduction

In a remarkable paper [Lon07], Longley proves that, when restricted to hereditarily total functionals over $\mathbb{N}$, many classes of higher-order functionals characterized by a "realizability-style" construction result in either the Kleene-Kreisel continuous functionals C (or its recursive substructure RC) or the hereditarily effective operations HEO. His approach is quite general and utilizes typed partial combinatory algebras (TPCAs) in an essential way. In particular (in an earlier version of the paper appearing as [Lon04], he notes that in the case of van Oosten's algebra $\mathcal{B}$ [vO99], a direct proof "would probably be feasible but rather tiresome." In this paper, we verify Longley's observation by giving such a direct proof via simulations between realizers for the two classes of functionals. Why do we bother with such a proof? Our ultimate goal is to understand the *efficiency* of such simulations. At type-level two, [BK02] considers a version of Ershov's presentation [Ers72, Ers74] of C as the extensional collapse of the Scott-continous (partial) functionals P. With this presentation, there is a natural complexity measure — *certificate size* — associated with realizers for continuous functionals, while for sequential functionals there is also measure, corresponding to the standard notion of decision-tree depth. Extending a technique of Blum [BI87, HH90, Tar89], it it shown that sequential realizers can simulate continuous realizers with quadratic overhead. The results of this paper, which give an inefficient simulation, may be viewed as a first step towards generalizing the result of [BK02] to all finite types.

Preliminaries. We assume the existence of surjective tuple coding functions $\langle \cdot, \ldots, \cdot \rangle_k : \mathbb{N}^k \to \mathbb{N}$ and a surjective sequence coding function $\langle \cdot \rangle : \mathbb{N}^* \to \mathbb{N}$. We will typcially drop the subscript on tupling functions when it is clear from the contexts. Right now we are not worried about quantitative aspects (e.g. bounds or efficient computability) of coding functions, but may need to revisit this later on. It does seem that surjectivity is important for some of our proofs. Also, we typically use length and projection functions implicitly in a pattern-matching fashion. We do sometimes use the sequence-length function lh which has the property that $\mathsf{lh}(\langle\rangle) = 0$ and $\mathsf{lh}(\langle u_0, \ldots, u_\ell \rangle) = \ell + 1$, as well as a sequence selection function $(\cdot)_i$ which has the property that $(\langle u_0, \ldots, u_\ell \rangle) = u_i$ for $0 \leq i \leq \ell$ (and is some default value for i not in the range.)

2 Continuous and sequential functionals of finite type

We first recall Kleene's [Kle59] definition of *countable functionals* via the method of *associates*. For sets X, Y, let $X \to Y$ denote the set of all functions from X to Y. For $f : \mathbb{N} \to \mathbb{N}$, and $n \geq 0$, let $\overline{f(n)}$ denote $\langle f(0), \ldots, f(n) \rangle$.

Definition 2.1. For each *type level* $k \geq 0$, $\mathsf{Ct}(k)$ is the set of *countable functionals of type level* k and $\mathsf{Ass}(k)$ is the set of *associates of type level* k. For each $F \in \mathsf{Ct}(k)$, there is a set $\mathsf{Ass}(F) \subseteq \mathsf{Ass}(k+1)$, the *associates of* F. For all k, $\mathsf{Ass}(k) = \bigcup_{F \in \mathsf{Ct}(k)} \mathsf{Ass}(F)$. We have $\mathsf{Ass}(0) = \mathsf{Ct}(0) = \mathbb{N}$, and $\mathsf{Ass}(1) = \mathsf{Ct}(1) = \mathbb{N} \to \mathbb{N}$. For $k = 0, 1$ and $F \in \mathsf{Ct}(k)$, $\mathsf{Ass}(F) = F$. For $k > 1$, we have $F : \mathsf{Ct}(k) \to \mathbb{N}$ is in $\mathsf{Ct}(k+1)$ if $\mathsf{Ass}(G) \neq \emptyset$, where $\alpha \in \mathsf{Ass}(G)$ iff for all $G \in \mathsf{Ct}(k)$ and all $\beta \in \mathsf{Ass}(G)$, there is an $n \geq 0$ such that

1. For all $m < n$, $\alpha(\overline{\beta(m)}) = 0$ (note that is always the case that $\alpha(\langle\rangle) = 0$.)
2. For all $m \geq n$, $\alpha(\overline{\beta(m)}) = F(G) + 1$.

In case there is an n for which (1) and (2) hold for α, β, we write $\alpha \bullet \beta$ to denote $F(G)$.

The countable functionals identify a subclass of higher-order functionals whose properties may be characterized in terms of ordinary (type-one) functions. This in turn allows for a very simple model of computation on higher-order inputs.

In [vO99], van Oosten provides a model of *hereditarily sequential functionals*, which is also based on type-one function application. As the name implies, in this model, associates may be viewed as sequential strategies, corresponding more closely to familiar notions of oracle computation. To begin, we review van Oosten's definition of a *dialogue* between partial functions.

Definition 2.2. An encoding $u = \langle u_0, \ldots, u_{n-1} \rangle$ of a sequence from $\mathbb{N}^*$ is a *dialogue* between $g, f : \mathbb{N} \rightharpoonup \mathbb{N}$ if for all i, $0 \leq i \leq n-1$, there is a j such that $g(u^{<i}) = 2j + 1$ and $f(j) = u_i$, where $u^{<i} = \langle u_0, \ldots, u_{i-1} \rangle$. The *application* $g|f$ is defined with value y (written $g|f = y$,) if there is a dialogue u between g and f such that $g(u) = 2y$. If $g(u^{<i}) = 2j + 1$ we will say that g *queries* f *at* j. If $g(u) = 2y$, we will say that g *answers* y. For a given g and f, we do not rule out the possibility that there is no dialogue between g and f (e.g., g may never answer, or may reach a point where it does not have a query.)

While this definition allows the definition of higher-order partial functionals, in this paper we are restricting attention to hereditarily total functionals. We now have the following

Definition 2.3. For each *type level* $k \geq 0$, $\mathsf{Seq}(k)$ is the set of *(hereditarily total) functionals of type level* k and $\mathsf{Sass}(k)$ is the set of *sequential associates of type level* k. For each $F \in \mathsf{Seq}(k)$, there is a set $\mathsf{Sass}(F) \subseteq \mathsf{Sass}(k+1)$, the *sequential associates of* F. For all k, $\mathsf{Sass}(k) = \bigcup_{F \in \mathsf{Seq}(k)} \mathsf{Sass}(F)$. We have $\mathsf{Sass}(0) = \mathsf{Seq}(0) = \mathbb{N}$, and $\mathsf{Sass}(1) = \mathsf{Seq}(1) = \mathbb{N} \to \mathbb{N}$. For $k = 0, 1$ and $F \in \mathsf{Seq}(k)$, $\mathsf{Sass}(F) = F$. For $k > 1$, we have $F : \mathsf{Seq}(k) \to \mathbb{N}$ is in $\mathsf{Seq}(k+1)$ if $\mathsf{Sass}(F) \neq \emptyset$, where $\sigma \in \mathsf{Sass}(F)$ iff for all $G \in \mathsf{Seq}(k)$ and all $\tau \in \mathsf{Sass}(G)$, it is the case that $\sigma | \tau = F(G)$.

3 The Equivalence

Our goal is to prove that for all $k \geq 0$, $\mathsf{Ct}(k) = \mathsf{Seq}(k)$. To do so, we work directly with associates, as follows:

Theorem 3.1. *For $k \geq 0$ and all $\alpha \in \mathsf{Ass}(k+1)$, $\beta \in \mathsf{Ass}(k)$, $\sigma \in \mathsf{Sass}(k+1)$ and $\tau \in \mathsf{Sass}(k)$, there are $\sigma^{\mathsf{c}} \in \mathsf{Ass}(k+1)$, $\tau^{\mathsf{c}} \in \mathsf{Ass}(k)$, $\alpha^{\mathsf{s}} \in \mathsf{Sass}(k+1)$ and $\beta^{\mathsf{s}} \in \mathsf{Sass}(k)$ such that*

1. $\alpha^{\mathsf{s}}|\tau = \alpha \bullet \tau^{\mathsf{c}}$
2. $\sigma^{\mathsf{c}} \bullet \beta = \sigma|\beta^{\mathsf{s}}$

References

[BI87] M. Blum and R. Imagliazzo. Generic oracles and oracle classes. In *Proc. 28th Ann. IEEE Symposium on Foundations of Computer Science*, pages 118–126, 1987.

[BK02] Samuel R. Buss and Bruce M. Kapron. Resource-bounded continuity and sequentiality for type-two functionals. *ACM Trans. Comput. Log.*, 3(3):402–417, 2002.

[Ers72] Ju. L. Ershov. Everywhere defined continuous functionals. *Algebra i Logika*, 11:656–665, 736, 1972.

[Ers74] Ju. L. Ershov. Maximal and everywhere defined functionals. *Algebra i Logika*, 13(4):374–397, 487, 1974.

[HH90] J. Hartmanis and L.A. Hemachandra. Robust machines accept easy sets. *Theoretical Computer Science*, (74):217–225, 1990.

[Kle59] S.C. Kleene. Countable functionals. In A. Heyting, editor, *Constructivity in Mathematics*, pages 81–100. North-Holland, Amsterdam, 1959.

[Lon04] John Longley. On the ubiquity of certain total type structures: (extended abstract). *Electron. Notes Theor. Comput. Sci.*, 73:87–109, 2004.

[Lon07] John Longley. On the ubiquity of certain total type structures. *Math. Struct. Comput. Sci.*, 17(5):841–953, 2007.

[Tar89] G. Tardos. Query complexity, or why is it difficult to separate $NP^a \cap co\text{-}NP^a$ from P^a by a random oracle? *Combinatorica*, (9):385–392, 1989.

[vO99] Jaap van Oosten. A combinatory algebra for sequential functionals of finite type. In *Models and computability (Leeds, 1997)*, volume 259 of *London Math. Soc. Lecture Note Ser.*, pages 389–405. Cambridge Univ. Press, Cambridge, 1999.