On the possibility of a Rice-like Theorem for primite recursive functions

A FEW SAMPLE PAGES...

Armando B. Matos, March 23, 2015

Contents

Also, for total recursive functions one can show that every decidable set is clopen as a subset of Baire space, which I guess is in a similar format to (the contra-positive of) Rice's theorem. But by the example this is not true for PR functions. By `aws`

B As posted in mathoverflow `computability/188137`

From `http://mathoverflow.net/questions/188137` with small modifications.

B.1 Original question

Consider decision problems in which the instance is a PR index i (or equivalently a "recursive" definition, or a LOOP program of) of a primitive recursive function. Denote the PR function (with PR index i) by φ_i. Examples of PR problems (input i):

> Problem P1 (decidable): let $n = \varphi_i(0)$. Are all integers $\varphi_i(1),\ldots,\ \varphi_i(n)$ prime?
>
> Problem P2: (undecidable, see `http://www.dcc.fc.up.pt/~acm/pr3.pdf`): $\exists n : \varphi_i(n) = 0$?

Conjecture. Looking to the program (index) is not more powerful than evaluating the function. In more detail:

<u>Conjecture.</u> The computational model ⟨Turing machine $M(i)$ with the PR index i of a PR function φ_i as input⟩ can not decide more properties than the (more restricted) model ⟨Turing machine M^f whose "input" is an oracle for computing $f(n)$ given n⟩.

<u>Notes.</u>

– We stress that the instance is a PR index (φ_i is always PR), not a TM index - it represents a LOOP program (say), not a set of quadruples.

– I use the following definition of "decidable" (transcribed from user aws): "P is decidable if there is a recursive function (or TM, using the CT thesis), that given the code for a primitive recursive function returns 1 if the function it codes is in P, and returns 0 otherwise".

– In "Rice (like) Theorem for primitive recursive functions?" I posted a similar but more vague question (no conjectures).

– We may obtain some information by looking to the definition of f. The maximum loop nesting of the LOOP program that defines f allows us to positive answers to to properties like

$$\exists n_0 \forall n \geq n_0 : f(n) \leq g(n)? \tag{4}$$

(the answer is positive for certain combinations of g and the Loop program corresponding to the index of f, see [15]). But this is not sufficient to decide that property. Note: apparently no oracle machine M^f implements the partial decision problem (4).

asked by Armando Matos

+1. This is a great question – very subtle. You want to know essentially whether having the PR definition of a primitive recursive function gives you any extra information beyond the course of values of that function.
Joel David Hamkins

B.2 Hoyrup example

> In an answer to "Rice (like) Theorem" for primitive recursive functions?"[10] Hoyrup gives an example of a property P that is decidable if a Loop program for the PR function is given, but undecidable by a Turing machine that only has access to an oracle for obtaining values of f. The results obtained also apply to other classes of total recursive functions.

It happens that there do exist non-trivial universal properties that are decidable. This property is expressed in terms of what we could call "primitive recursive Kolmogorov complexity".

Definition. If $v = (v_0, \ldots, v_{n-1})$ is a sequence of natural numbers then let $K_{\mathrm{pr}}(v)$ be the size of a shortest Loop program computing a function f extending v, i.e. satisfying $f(0) = v_0, \ldots, f(n-1) = v_{n-1}$.

Unlike the usual notions of Kolmogorov complexity, $K_{\mathrm{pr}}(v)$ is computable. However it is not primitive recursive.

For a function f, let $f|n$ be the finite sequence $(f(0), \ldots, f(n-1))$.

Claim. The property $\forall n, K_{\mathrm{pr}}(f|n) \leq n$ is decidable, given a Loop program for f.

Proof. Given a loop program p for f, one has $K_{\mathrm{pr}}(f|n) \leq |p|$ for all n. In order to check the property, one can only look at $n < |p|$. As K_{pr} is computable, the property is decidable.

Observe that the property is not decidable if one is only given f as oracle, as no finite prefix of f is sufficient to ensure the property: for each finite sequence $v = (v_0, \ldots, v_{n-1})$ there is v_n such that $K_{\mathrm{pr}}(v_0, \ldots, v_n) > n + 1$ hence no extension of $(v_0, \ldots, v_n)$ satisfies the property (the property is a closed subset of the Baire space that has empty interior).

More generally and for the same reasons, if $h : \mathbb{N} \to \mathbb{N}$ is a computable non-decreasing unbounded function then the property P_h defined by

$$f \in P_h \iff [\forall n : K_{\mathrm{pr}}(f|n) \leq h(n)]$$

is decidable given a Loop program for f but not given f as oracle. If $h(1)$ is sufficiently large then P_h is non-empty as it contains all the functions computed by Loop programs of size $\leq h(1)$.

[10] `http://mathoverflow.net/questions/155413` Question by: Armando B. Matos, answer by: Mathieu Hoyrup.

An analog of Rice and Rice-Shapiro theorem

So far, we know some basic properties that are decidable: extending a finite sequence v (decidable given f), the property P_h of having h-compressible prefixes (decidable given a Loop program). Now, there is an analog of Rice and Rice-Shapiro theorems, stating that they form a "subbasis" (as in topology) of the semi-decidable properties: every semi-decidable property can be obtained as a union of finite intersections of these simple properties.

Theorem. Let P be a property of primitive recursive function. The following are equivalent:

$f \in P$ is semi-decidable given a Loop program for f,

P is a computable disjunction of properties of the form

$$f \text{ extends } v \text{ and } \forall n : K_{\mathrm{pr}}(f|n) \leq h(n).$$

The result is more general as it applies to any class of total computable functions that can be computably enumerated (for instance the polynomial-time computable functions, the provably total computable functions, etc.). All this can be found in the paper [8].

B.3 Turetsky construction

- In the beginning you say "there is no Turing functional Γ with $i \in A \Leftrightarrow \Gamma^{\varphi_i}(0)\downarrow$. This is a counterexample to your conjecture".
  So we may suppose that φ_i is an enumeration of the PR functions (not of all partial recursive functions), right?

- An important detail in the construction of the recursive set A not recognized by any Turing functional is, I think, the selection in step 2 of a "very large" n". Here, "very large" is related to a function that grows faster than any PR function, so that "no φ_i extends…". Using this very large functions, can't we define a simpler construction?
  Extremly spaced 1's in a sea of'0s seem not to disturb the conjecture.

- Is A recursive? More specifically, isn't the following problem undecidable for many (all nontrivial?) sets A?
  Parameter: a nontrivial set (whatsoever) A:
  Input: two PR function indices (say PR definitions or Loop programs) i and j.
  Question: Is $\varphi_i = \varphi_j$? (function equality)

Your conjecture is false. We can construct a recursive set A such that if $\varphi_i = \varphi_j$, then $i \in A \Leftrightarrow j \in A$, but there is no Turing functional[11] Γ with $i \in A \Leftrightarrow \Gamma^{\varphi_i}(0)\downarrow$. This is a counterexample to your conjecture.

[11] See for instance [22], page 23.

References

[1] Cristian Calude, Helmut Jürgensen, and Ludwig Staiger. Computable functionals of finite type. *Theoretical Computer Science*, 410(24–25):2323–2335, 2009.

[2] John Case. Effectivizing inseparability. Technical Report 89-04, State University of New York (Buffalo, NY US), 1989.

[3] John Case, Kemal Ebcioglu, and Mark Fluk. R.e. inseparable general and subrecursive index sets (review). *The Journal of Symbolic Logic*, 48(4):1233–1236, 1983.

[4] G. S. Ceitin. Algorithmic operators in constructive complete separable metric spaces. *Doklady Akad. Nauk*, 128:49–52, 1959. (in Russian).

[5] N. Cutland. *Computability: An Introduction to Recursive Function Theory*. Cambridge University Press, 1980.

[6] von Ju. L. Eršov. On a hierarchy of sets I. *Algebra Logika*, 7:47–74, 1968. (in Russian); English translation, Algebra and Logic **7** (1968) pages 25-43.

[7] von Ju. L. Eršov. Computable functionals of finite type. *Algebra Logika*, 11:367–437, 1972. (in Russian); English translation, Algebra and Logic **11** (1968) pages 203-242.

[8] Mathieu Hoyrup. A Rice-like theorem for primitive recursive functions. `http://arxiv.org/abs/1503.05025`, `cs.LO`, 2015.

[9] Neil D. Jones. *Computability and complexity - from a programming perspective*. Foundations of computing series. MIT Press, 1997.

[10] Alexander Kechris. *Classical Descriptive Set Theory*. Springer-Verlag, 1995.

[11] Stephan Cole Kleene. *Introduction to Metamathematics*. North-Holland, 1952. Reprinted by Ishi press, 2009.

[12] G. Kreisel, D. Lacombe, and J. R. Shoenfield. *Partial recursive functionals and effective operations*, pages 290–297. Constructivity in Mathematics (editor: A. Heyting), Studies in Logic and the Foundations of Mathematics. North-Holland, 1959.

[13] Masahiro Kumabe and H. Reiju Mihara. Computability of simple games: A characterization and application to the core. Technical Report 6803, MPRA, Munich Personal RePEc Archive, 2007.

[14] Armando B. Matos. Primitive recursive functions: decidability problems, 2014.

[15] A. R. Meyer and D. M. Ritchie. The complexity of loop programs. In *Proceedings of 22nd National Conference of the ACM*, pages 465–469. Association for Computing Machinery, 1967.

[16] Bernard Moret. *The Theory of Computation*. Addison-Wesley, 1998.

[17] Yiannis N. Moschovakis. *Descriptive Set Theory*. American Mathematical Society, second edition, 2009.

[18] Piergiorgio Odifreddi. *Classical Recursion Theory – The Theory of Functions and Sets of Natural Numbers.* Studies in Logic and the Foundations of Mathematics. Elsevier North Holland, first edition, 1989. Second impression.

[19] Iain Phillips. Recursion theory. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 1, pages 79–187. Oxford University Press, 1992.

[20] G. H. Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the Americal Mathematical Society*, 74:358–366, 1953.

[21] Hartley Rogers. *Theory of Recursive Functions and Effective Computability.* MIT Press Cambridge, MA, 1987. Original edition 1967. Third printing (1992).

[22] Robert I. Soare. Turing oracle machines, online computing, and three displacements in computability theory, 2007. Parts of this paper were delivered in an address to the conference Computation and Logic in the Real World, at Siena, Italy, June 2007.

[23] Dieter Spreen. Effective inseparability in a topological setting. *Anals of Pure and Applied Logic*, 80(3):257–275, 1996.

[24] Dieter Spreen. On effective topological spaces. *The Journal of Symbolic Logic*, 63(1):185–221, 1998.

[25] Stanley Tennenbaum. Inseparable sets and reducibility. Technical report, The University of Michigan, College of Literature, Science, and the Arts, New York, 1961. ORA Project 03105.

[26] Alan Turing. Systems of Logic Based on Ordinals, a dissertation presented to the faculty of Princeton University in candidacy for the degree of Doctor of Philosophy, 1938.