Primitive recursive functions

Decidability problems

Only pages: 1, 2, 3, 5, 23, 30, 43, 49

Armando Matos

LIACC, UP

2014

Abstract

Although every primitive recursive (PR) function is total, many problems related to PR functions are undecidable. In this work we study several questions associated with PR functions and characterize their degree of undecidability. For instance, we show that the injectivity problem belongs to the class $\Pi_1^0 \setminus \Delta_0^0$ of the arithmetic hierarchy, while the surjectivity and the bijectivity problems belong to $\Pi_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$. It follows that these subclasses of PR functions can not be enumerated – and thus cannot be characterized by an effective property. Several other PR decision problems are studied like, for instance, problems related to the existence and number of zeros and to the size of the codomain.

We consider a specific but important decision problem, "does a given PR function has at least one zero?", and establish the necessary and sufficient input and output restrictions for decidability – the "frontiers of decidability" of this problem. We also study undecidability results for a more general class of problems, namely existential first order logic with equality and composition of PR functions. A PR function as a part of an "acyclic PR function graph" is also studied. This setting may be useful to evaluate the relevance of a given PR function as a part of a larger PR structure.

Contents

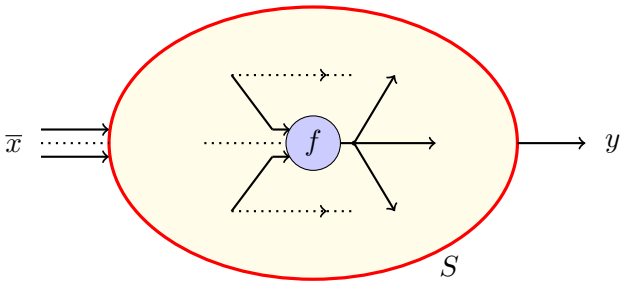

Figure 1: A PR function f is considered to be an *important* part of a complex structure S if the following problem is undecidable: instance: f, question: "$\exists \overline{x} : S(\overline{x}) = 0$?". We will show that an arbitrary acyclic system S containing one occurrence of f can always be reduced to the normal form illustrated in Figure 9 (page 43).

Part I, specific problems

In Section 3 (page 13) several decision problems associated with PR functions are studied. These problems include the existence of zeros, the equivalence of functions, "domain problems", and problems related with the injectivity, surjectivity, and bijectivity of PR functions. For each of these problems the degree of undecidability is established in terms of their exact location in the in the arithmetic hierarchy; see Figures 2 (page 23) and 3 (page 23).

Part II, classes of problems

Suppose that the PR function f, the instance of a decision problem, is "placed" somewhere inside a larger PR system S, see Figure 1. In algebraic terms this means that S is obtained by *composing* fixed PR functions with the instance f. A more general form of composition, based on *acyclic graphs* is also studied, see Figure 8 (page 40).

Then we ask an important question about $y = S(\overline{x})$, namely, "$\exists \overline{x} : S(\overline{x}) = 0$"? This is in fact a fundamental question:

- In terms of the Kleene normal form, the existence of a zero is the basic undecidable question, and it is intimately related – in fact, equivalent – to the halting problem.
- Many other decision problems can be reduced to this problem.

The *class of problems*, parametrized by S, is:
Question: A PR function f.
Instance: Is there some $\overline{x}$ such that $S(\overline{x}) = 0$?
In some sense we can say that f *can be considered an important component of* S *if (and only if) this problem is undecidable.*

Num	PR problem	Class	Proof
1	HAS-ZEROS	$\Sigma_1^0 \setminus \Delta$	page 15
2	EXACTLY-ONE-ZERO	$\Sigma_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$	page 15
3	AT-LEAST-k-ZEROS	$\Sigma_1^0 \setminus \Delta$	page 16
4	EXACTLY-k-ZEROS	$\Sigma_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$	page 16
5	ZERO-FUNCTION	$\Pi_1^0 \setminus \Delta$	page 16
6	∞-ZEROS	$\Pi_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$	page 16
7	CODOMAIN-k, $k = 1$	$\Pi_1^0 \setminus \Delta$	page 18
8	CODOMAIN-k, $k \geq 2$	$\Sigma_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$	page 18
9	FINITE-CODOMAIN	$\Sigma_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$	page 19
10	INJECTIVE	$\Pi_1^0 \setminus \Delta$	page 20
11	ONTO	$\Pi_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$	Section 3.3.2
12	BIJECTIVE	$\Pi_2^0 \setminus (\Sigma_1^0 \cup \Pi_1^0)$	Section 3.3.2

Figure 2: Some decision problems studied in this paper. In every problem the instance is a PR function. Among these problems, only HAS-ZEROS and AT-LEAST-k-ZEROS correspond to effectively "enumerable" languages or, equivalently, can be characterized by a "model of computation". The problems are numbered (first column, in this color) for reference in Figure 3, page 23.

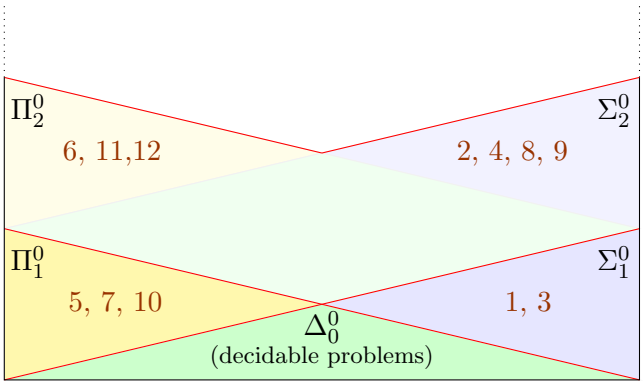

Figure 3: Location in the arithmetic hierarchy of the problems mentioned in Figure 2, page 23. The problem numbers (1 to 12 in this color) refer to the numbers in the first column of Figure 2.

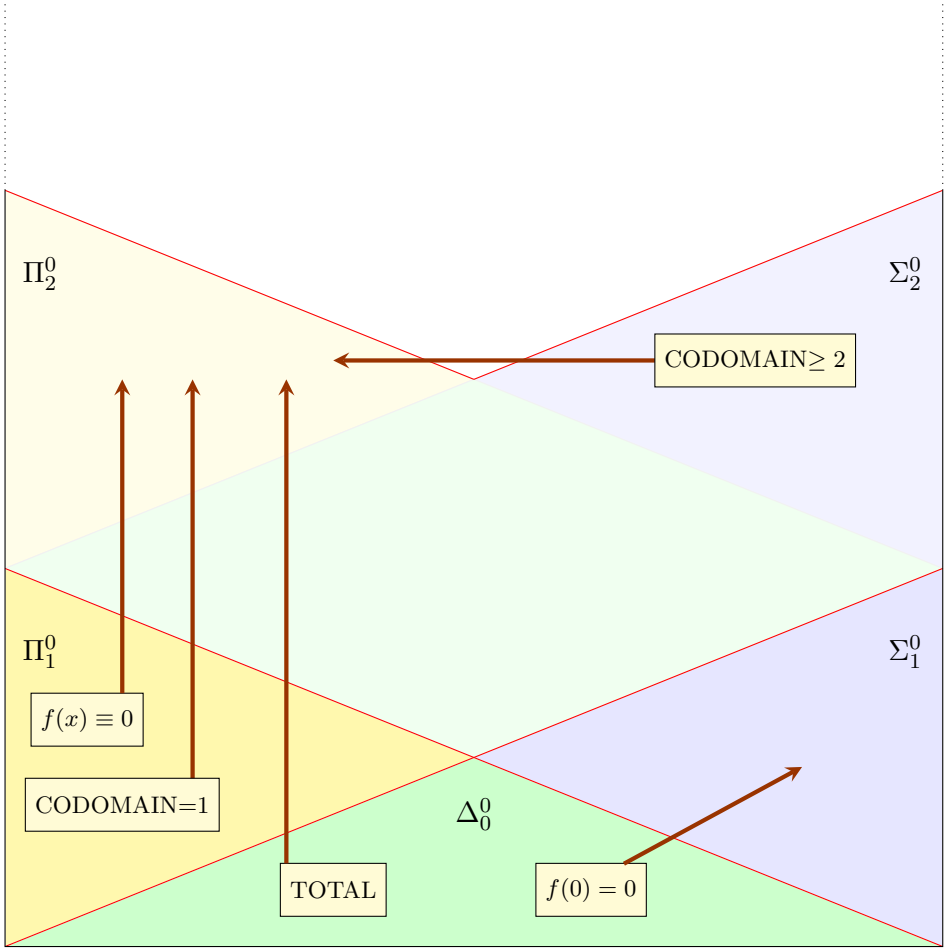

Figure 6: The classes of five particular problems when the instance is a primitive recursive function (origin of the arrow) and when it is a partial recursive function (tip of the arrow). CODOMAIN=1 denotes the problem CODOMAIN-k with $k = 1$, CODOMAIN≥ 2 denotes the problem CODOMAIN-k for some fixed $k \geq 2$, and $f(x) \equiv 0$ denotes the ZERO-FUNCTION problem. See also Figure 5, page 25.

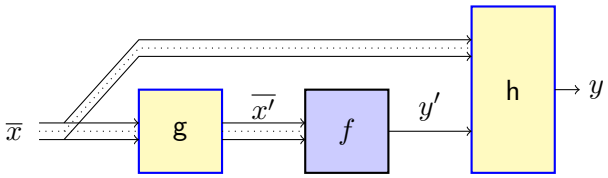

Figure 9: An arbitrary composition of one occurrence of f with PR functions can be reduced to this "normal form". The corresponding expression has the form $y = \mathsf{h}(\overline{x}, f(\mathsf{g}(\overline{x})))$ where g has multiple outputs.

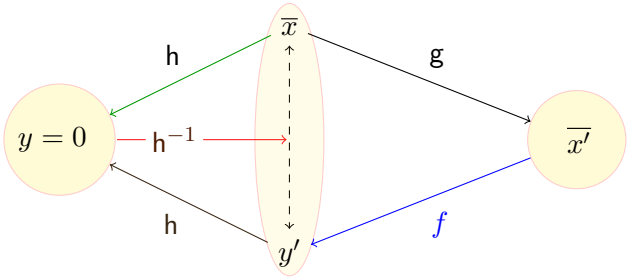

Figure 10: Relationship between the values in Figure 9, page 43. The global system S, $y = S(\overline{x})$, has a zero iff, for some $(\overline{x}, y') \in h^{-1}(0)$, we have $f(g(\overline{x})) = y'$. For the particular case $S(\overline{x}) \equiv \mathsf{h}(f(\mathsf{g}(\overline{x})))$, that is, when there is no green arrow in the diagram, there is an explicit ("acyclic") condition for the existence of a zero of S, see Theorem 22 (page 34). But no such condition is known for the general case.

It is interesting in particular to study systems in which the compositions of f with itself are allowed. Recall the proof of Theorem 20 (page 33), where u takes the role of f. Part of the proof of this result involves the statement that, for any fixed function g with an infinite domain (see Equation (6)), we have

$$\mu_y(f(\overline{x}, y))) = \mu_y(f(\overline{x}, \mathsf{g}(y))) \tag{10}$$

If we consider an expression like $f(\overline{x}, f(y))$ (instead of $f(\overline{x}, \mathsf{g}(y))$), the equality (10) may not hold. If this case, that is, when $\mathsf{g} = f$, the statement of Theorem 20 (page 33) is not useful. For instance, that criterion applied to the problem

$$\text{"Given } f\text{, does the function } f(f(x)) \text{ have a zero?"} \tag{11}$$

says that this problem is undecidable iff the codomain of f is infinite. But we can not characterize the decidability of a problem with instance f by an *undecidable constraint on f itself*, namely "the domain of f is infinite".

[12] Daniel Fredholm. Computing minimum with primitive recursion over lists. *Theoretical Computer Science*, 163(3):269–276, 1996.

[13] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[14] Bernhard Goetze and Werner Nehrlich. Loop programs and classes of primitive recursive functions. In *MFCS*, pages 232–238, 1978.

[15] Oded Goldreich and Bernd Meyer. Computational indistinguishability: algorithms vs. circuits. *Theoretical Computer Science*, 19(1):215–218, 1998.

[16] Oded Goldreich and Madhu Sudan. Computational indistinguishability: A sample hierarchy. *IEEE Conference on Computational Complexity 1998*, pages 24–33, 1998.

[17] R. L. Goodstein. Logic-free formalisations of recursive arithmetic. *Mathematica Scandinavica*, 2:247–261, 1954.

[18] A. Grzegorczyk. Some classes of recursive functions. *Rozprawy Matematyczne*, 4:1–45, 1953.

[19] Hans Hermes. *Enumerability, Decidability, Computability*. Springer-Verlag, 1969.

[20] Peter G. Hinman. *Recursion-theoretic Hierarchies*. Perspectives in mathematical logic. Springer-Verlag, 1978.

[21] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[22] Stefan Kahrs. The primitive recursive functions are recursively enumerable. University of Kent at Canterbury, Department of Computer Science, 200X.

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

[24] Stephen C. Kleene and Emil L. Post. The upper semi-lattice of degrees of recursive unsolvability. *Ann. of Math. (2)*, 59:379–407, 1954.

[25] Stephen Cole Kleene. Recursive predicates and quantifiers. *Transactions of the American Mathematical Society*, 53(1):41–73, 1943.

[26] Stephen Cole Kleene. Arithmetical predicates and function quantifiers. *Transactions of the American Mathematical Society*, 79(1):312–340, 1955.

[27] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, third edition, 2008.

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