A Complexity-Theoretic Proof of a Recursion-Theoretic Theorem

Amir M. Ben-Amram
Department of Computer Science
The Academic College of Tel-Aviv Yaffo
Tel Aviv, Israel
amirben@mta.ac.il

October 20, 2004

Abstract

It is well known that Complexity Theory inherited many tools from Recursion Theory. This note is an example where tools from Complexity Theory simplify the proof of a Recursion Theoretical result, illustrating the value of mingling the two subjects.

Background. We write $A \leq_{rec} B$ if there is a many-one reduction (recursive reduction) from A to B, namely a computable function f such that $d \in A \iff f(d) \in B$. It is well known that the halting problem $\text{HALT} = \{(p.d) \mid [\![\text{p}]\!]d \downarrow\}$ is RE-complete under such reductions. Thus, for a recursively enumerable set A to be complete it is necessary and sufficient to have $\text{HALT} \leq_{rec} A$.

Set $S \in \text{RE}$ is *simple* if its complement is infinite but has no infinite recursively enumerable subset. An example is the set of Kolmogorov-compressible strings (Li and Vitányi, 1997). In this note we prove a classic Recursion-Theoretical result:

> A simple set cannot be RE-complete under many-one reductions.

To prove this result, we show that $\text{HALT} \leq_{rec} A$ implies that $\overline{A}$ has an infinite recursively enumerable subset. The typical route from the basic definitions up to this result (as found in Recursion Theory books, e.g., Soare 1987) involves constructions which are quite intricate.

We next present a proof which avoids the advanced[1] recursion-theoretic machinery by using instead a well-known Complexity-Theoretic tool: the Time Hierarchy Theorem (e.g., Jones 1997)[2].

The new proof. Assume that $A \in \mathrm{RE}$ and that HALT $\leq_{rec} A$ via function f. We will prove that $\overline{A}$ has an infinite recursively enumerable subset.

Let $T_f(n)$ be an increasing, time constructible function that bounds the running time of computing f for input of length n, and let $T'(n) = T(2n)$. Function T' is increasing and time-constructible and thus the Hierarchy Theorem provides a decidable set B that cannot be decided in $O(T'(n))$ time.

Let $g(x) = f(p_B.x)$, where p_B is a fixed program that halts exactly on the elements of B. Function g can be computed in time

$$T_g(n) \leq T(|p_B| + 1 + n) \leq T(2n) = T'(n)$$

for n large enough. Observe that g reduces B to A. Thus, $C = g(\overline{B})$ is a recursively enumerable subset of $\overline{A}$. Assume C to be finite. We obtain an algorithm to decide B in $O(T'(n))$ time, namely: compute $g(x)$ and look the result up in a table for C. This contradicts the Hierarchy Theorem, so we conclude that C is infinite.

References

Jones, Neil D. (1997) *Computability and Complexity.* MIT Press, Cambridge, Mass., 1997.

Li, Ming and Vitányi, Paul (1997) *An Introduction to Kolmogorov Complexity and Its Applications.* 2nd ed., Springer-Verlag, New York, 1997.

Soare, Robert I. (1987) *Recursively Enumerable Sets and Degrees.* Springer-Verlag, Berlin 1987.

[1]relative to an undergraduate curriculum.

[2]There are slight variations in the time hierarchy theorem for different computational models. However, for our use of it, these differences can be ignored.