Proof regarding Numbers of Computer Programs vs. Numbers of Human Functions

J. H. DeBlois

October 31, 2024

1 Introduction: Tech Cone Tracker:

What needs proof is: Computer programs can be modeled as words in a language and listed in order with a proof that the size of the set is countably infinite. $\aleph_0$ stands for a countable infinity.

Human functions can be modeled, but attempts to list them fail, and we can prove that the number of such functions is uncountably infinite. Thus, forever bigger than $\aleph_0$.

Our Tech Cone Tracker shows the ever-expanding number of computer programs written so far. Possibly, we depict it in black and white. It also shows the much larger outer edge of human activities or functions. Possibly, we depict these in colors.

It is of interest when a human activity is partially replaced by a computer program. Some new human functions may become necessary. Our proof shows that the total of human functions in use must grow too, and by more. Our reference is Theory of Formal Languages with Applications, Dan A. Simovici and Richard L. Tenney, World Scientific, 1999.

2 Count the Number of Computer Programs

To count the computer programs, we use the concepts of sets, cardinality, bijection, alphabet, words and language. As Prof. Simovici mentioned (p. 57), "Programs and the data they manipulate may be regarded as words over appropriate alphabets."

A set is a collection of objects. The elements in a set may themselves be sets. We use $\mathbb{N}$ for the set of natural numbers. The power set is the set of all subsets of a set. Basic definitions are found on pp. 3-5 in Simovici and Tenney.

A relation is a set of ordered pairs. If A and B are sets, then the Cartesian product of A and B, written $A \times B$ is the set of all pairs (a, b) such that a is in A and b is in B. An equivalence relation is reflexive, symmetric and transitive. These definitions are found on pp. 6-8 in Simovici and Tenney.

A relation is a function if it satisfies a special condition. Definition 1.2.38, p. 12, A relation ρ is called a *function* if $(a, b_1) \in \rho$ and $(a, b_2) \in \rho$ implies $b_1 = b_2$ for all a, b_1, and b_2. Thus, a value in A maps to at most one value in B. See also definitions of one-to-one, total and onto. The set of all functions from A to B is denoted $A \longrightarrow B$.

Theorem 1.2.46, p. 13, Let A and B be two finite sets. We can calculate the number of functions from A to B: There exist $|B|^{|A|}$ such functions from A to B. Also, p. 15, there is a bijection from the set of subsets of M and the set of characteristic functions defined on M.

On p. 57, an alphabet is a finite non-empty set. The elements are symbols. Definition 2.2.1 A word of length n on an alphabet A is a sequence of length n of symbols of this alphabet.

Also, "Example 2.2.3, p. 58, Any C program is a word over the basic alphabet of this language that includes small and capital letters, as well as special symbols, such as parentheses, brackets, braces, spaces, new line characters, quotation marks, etc. Not all these characters are visible; in other words, some characters (such as spaces) appear on paper only as white spaces. For example, the famous C program:

```
#include <stdio.h> main(){ printf("hello, world\n"); }
```

can be looked at as a word."

Now, from cs622 class notes, "Numbering Words", we have: "Let $A = \{a_0, \ldots, a_{n-1}\}$ be an alphabet containing n symbols. Words over A can be encoded as natural numbers; in other words, we can define a bijection $\phi_A : A^* \longrightarrow N$ by:

$$\phi_A(x) = \begin{cases} 0 & \text{if } x = \lambda \\ n\phi_A(y) + i + 1 & \text{if } x = ya_i \end{cases}$$

for every x in A^*."

See Example, same page, if $A = \{a_0, a_1, a_2\}$, $x = a_0a_1a_0a_2$ and $y = a_2a_2a_2$, then:

$$\phi_A(x) = 3^3 \cdot 1 + 3^2 \cdot 2 + 3^1 \cdot 1 + 3 = 51$$
$$\phi_A(y) = 3^2 \cdot 3 + 3^1 \cdot 3 + 3 = 39$$

Note that any word of a certain length will have its interval, no two distinct words in A^k can be mapped into the same number in $l(k)$, so ϕ_A defines an injection of A^k into $l(k)$; and for every number m in $l(k)$ there is a word x in A^k such that $\phi_A(x) = m$. Therefore, ϕ_A is a bijection between A^* and $\mathbb{N}$.

Therefore, the number of possible computer programs is a countable infinity.

3 Count the Number of Human Functions

Now, we count the number of human functions. Suppose we start with a list L of the basic things humans can think of. Then, because ideas take form in different ways, we let each human function be a subset of that list. $P(L)$ can be shown to be not countable in a proof by contradiction. Suppose $P(L)$ is countable. Then we have the mapping $f : \mathbb{N} \longrightarrow P(L)$.

Per p. 44, we could have this list:

$$\begin{array}{lllllllll}
0: & a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & \cdots & \text{(a particular subset)} \\
1: & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & \cdots & \\
2: & a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & \cdots & \\
3: & a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & \cdots & \\
4: & a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & \cdots & \\
5: & a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & \cdots & \\
 & & \vdots & & & & & \\
\text{k}: & a_{k0} & a_{k1} & a_{k2} & a_{k3} & a_{k4} & \cdots & a_{kk}
\end{array}$$

where:

$$a_{ij} = \begin{cases} 0 & \text{if } j \notin f(i) \text{ where } f(i) \text{ is a particular subset} \\ 1 & \text{if } j \in f(i) \end{cases}$$

Then, quoting further from Simovici and Tenney, p. 44, "In other words, the a_{ij}s correspond to the characteristic function of the set $f(i)$. The set D is formed by 'going down the diagonal' and spoiling the possibility that $D = f(k)$, for each k. At row k, we look at a_{kk} in column k. If this is 1, i.e., if $k \in f(k)$ then we make sure that the corresponding position for the set D has a 0 in it by saying that $k \notin D$. On the other hand, if a_{kk} is a 0, i.e., $k \notin f(k)$, then we force the corresponding position for the set D to be a 1 by putting k into D. This guarantees that $D \neq f(k)$, because its characteristic functions differ from that of $f(k)$ in column k."

Thus no bijection.

This proof technique, diagonalizaton, first appeared in the 1891 paper of Georg Cantor.

4 References

Simovici, Dan A. and Richard L. Tenney, Theory of Formal Languages with Applications, World Scientific Publishing Co., 1999.