Computation, Information, and Intelligence (ENGRI/CS/INFO/COGST 172), Spring 2007
4/25/07: **Lecture 38 aid — The halting function**

Topics: Discovering some limits on the powers of computers (and by extension, people, or at least computer and information scientists) by:

- considering the machine/program duality from the beginning of the semester more formally, thus seeing that TMs are actually quite powerful, and
- proving that the halting function is not computable, thus showing that there are fundamental limits to our understanding of computational processes.

I. Enumeration of "B-input machines" We denote by $M_1, M_2, \ldots$ an infinite list (with no repeats) of *all* TMs that take only sequences of B's as input.[1] The list is such that *given a (suitably encoded) non-negative integer i, it is possible to recover the program for M_i* (i.e., there exists a Turing machine that does the job).

The sequences of B's are intended to be encodings of non-negative integers.[2] Thus, all computable functions whose domain is the non-negative integers[3] are represented in the list, but there are also many TMs on the list that don't compute such functions.

II. The halting function

$$h(M_i, j) = \begin{cases} 1\ (\text{yes}), & \text{if } M_i \text{ would halt given } j \text{ B's as input} \\ 0\ (\text{no}) & \text{if } M_i \text{ would not halt given } j \text{ B's as input} \end{cases}$$

Note that by "halting", we mean "would halt in a finite number of steps".

III. The evil machine X Suppose a termination detector D exists. Then we could construct a Turing machine X that, when given ℓ B's as input,

(a) recovers the program for M_ℓ;
(b) runs D to determine the value (0 or 1) of $h(M_\ell, \ell)$; and
(c1) if that value is "1", enters an artificial infinite loop
(c2) if that value is "0", sets its output to 1 and halts

Common point of confusion: we can show that X, and therefore D, does not exist, meaning that the halting function is not computable. But the function does *exist* (as much as any mathematical function, like $f(x) = x^2$, can be said to exist), and is well-defined, and so on. So for every M_i and j, there is a value, either 0 or 1, to $h(M_i, j)$; it's just that *we have no general way to determine what that value is for arbitrary M_i and j.*

---

[1] Technically speaking, we should specify some reasonably-sized fixed finite symbol alphabet that all the TMs on the list are restricted to using when it comes to writing things down on their tape (actually, it suffices simply to specify a maximum size for the "output alphabet" that TMs on the list can use). But these technical details can be ignored for the purposes of our discussion.

[2] This encoding is non-traditional; we have chosen it in order to remove a layer of self-reference from our proof.

[3] We technically should put some restrictions on the range, too — essentially, that the elements of the range be representable as finite sequences over a finite symbol set — but we'll ignore this point.