

COS 116

The Computational Universe

More about self-reproducing programs

This handout gives more details on the construction of self-reproducing programs, which was sketched in Lecture 10.

In this construction we assume that the machine has a separate “output” tape on which it can write English symbols. (Alternatively, we can assume that instead it represents English letters using some binary code.)

The self-reproducing program is designed in two steps. First we design part B. It is some kind of a translator program, which is capable of two types of translations. Whenever it is started with some sequence of bits on the tape, say V , it first produces the sequence of instructions that would print this sequence. This is the first act of “translation.”

(For instance, if the sequence of bits is 101, the sequence of instructions that produce it is
PRINT 1
PRINT 0
PRINT 1.)

Then part B will print the program whose binary code is exactly V , the sequence of bits that it was given. This is the second act of “translation.”

Note that part B is easy to write since the two “translations” are so mechanical.

Having written part B, we turn to part A. It is exactly the sequence of PRINT instructions that would print the binary code of B.

The self-reproducing program is the composite program where part A is followed by part B.

How does the self-reproduction happen? When we run the program with a blank tape, part A executes first, and prints a bit sequence on the tape which is the binary code of part B. Then part B takes over. It finds a sequence of bits on the tape (which happen to be its own binary code, but it doesn’t know this). After it does its two translations, it produces both part A and part B (of course, still not realizing that part B is itself).

This construction of a self-reproducing program can be easily modified to give the self-reproducing program any desired functionality ---such as deleting your files, or mailing itself to all email addresses in your addressbook. This is how computer viruses work.