The Virus Problem
CSE 5211 Analysis of Algorithms
Spring 2017 (April 26, 2017)

Dave Evans presents a nice description of the virus problem (VP), see On the Impossibility of Virus Detection

Definition 1: Virus
A virus is a computer program that when executed will copy its own code into another program.

Problem 1: Virus Problem

Input: A description of a program P and its input x.
Output: If P(x) behaves like a virus (running it can infect other files) output True. Otherwise, output False.

Assume there a program, virusDetect, that decides the Virus Problem (VP).

Listing 1: Virus Detection

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(Virus Detection) ≡</td>
</tr>
<tr>
<td></td>
<td>virusDetect(program P, input x) {</td>
</tr>
<tr>
<td></td>
<td>if (P(x) acts like a virus) then true;</td>
</tr>
<tr>
<td></td>
<td>else false;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>

A Diagonalization Argument

Assume there is a virus V. (Imagine how such a program could be written). Mimic the argument given in class and the notes to show there cannot be a program that decides the virus problem (VP).

That is, write a program, call it D(program P) if you like, that uses virusDetect, to create a contradiction.

Reduction of HP to VP

Spend a few minutes going over Evans’ note On the Impossibility of Virus Detection.

Another way to show the virus problem is undecidable is to show HP reduces to VP. That is, if VP were decidable, then HP would be decidable. And, since HP is undecidable, VP must be also.

The main idea is to construct a program makeVirus that first executes any input program P, and then serially, if P halts, executes a virus V.
Argue that program \texttt{halt} below decides the halting problem. Use this to conclude there cannot be a virus detection decider.

**Listing 3: Halting Decider**

\begin{verbatim}
2b (Halt Decider 2b) ≡
    halt(program P) {
        if (virusDetect(makeVirus, P)) then true;
        else false;
    }
\end{verbatim}

**References**