Section 3.1
Algorithms

Example:

The (infamous) Halting Problem

We wish to establish the nonexistence of a universal debugging program.

Theorem: *There does not exist a program which will always determine if an arbitrary program P halts.*

We say the Halting Problem is *undecidable.*

Sidenote: this is not the same as determining if a specific program or finite set of programs halts which is decidable.

There is always exists a program to determine if a specific program P halts:

• Construct program P1 which always prints 'yes' and halts.

• Construct program P2 which always prints 'no' and halts.

One of the two programs, P1 or P2, is the correct (deciding) program (we may not know which one!).

Hence this problem is decidable.

To simplify the argument: consider input-free programs only (which may call other procedures)

Proof:

Suppose there is such a program called HALT which will determine if any input-free program P halts.

HALT(P) prints 'yes' and halts if P halts,

otherwise,

HALT(P) prints 'no' and halts.

We now construct another procedure as follows:

procedure ABSURD;
if HALT(ABSURD) = 'yes' **then**
  while true **do** print 'ha'

Note that ABSURD is input-free.

• If ABSURD halts then we execute the loop which prints unending gales of laughter and thus the procedure does not halt.

• If ABSURD does not halt then we will exit the program and halt.

Hence,

• ABSURD

- halts if it doesn't

and

- doesn't halt if it does

which is an obvious contradiction. (You are free to loose sleep over this).

Hence such a program does not exist.

Q. E. D.

Note: This is not the same as asserting a program exists and we don't know how to write it or that it is very difficult to write such a program!
