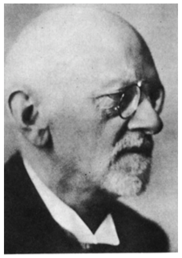
7.7: Computability



> Take any definite unsolved problem, such as the question as to the irrationality of the Euler-Mascheroni constant γ, or the existence of an infinite number of prime numbers of the form 2^n+1. However, unapproachable these problems may seem to us and however helpless we stand before them, we have, nevertheless, the firm conviction that their solution must follow by a finite number of purely logical processes.
>
> *-David Hilbert, in his 1900 address to the International Congress of Mathematics*

Halting Problem

Halting problem. Write a Java function that reads in a Java function f and its input x, and decides whether f(x) results in an infinite loop.

integer that equals the sum of its proper divisors

Ex: is there a *perfect* number of the form: 1, 1+x, 1+2x, 1+3x, ...
- x = 1: halts when n = 28 = 1 + 2 + 4 + 7 + 14.
- x = 2: finding odd perfect number is famous open math problem.

```
public void f(int x) {
   for (long n = 1; true; n = n + x) {
      long sum = 0;
      for (long i = 1; i < n; i++)
         if (n % i == 0) sum = sum + i;
      if (sum == n) return;
   }
}
```

halt if n is perfect

Undecidable Problem

A yes-no problem is undecidable if no Turing machine exists to solve it.

Theorem (Turing, 1937). The halting problem is undecidable.
- No Turing machine can solve the halting problem.
- By universality, not possible to write a Java function either.

Proof intuition: lying paradox.
- Divide all statements into two categories: truths and lies.
- How do we classify the statement: *I am lying.*

Key element of paradox: self-reference.

Halting Problem Proof

Assume the existence of `halt(f,x)`:
- Input: a function `f` and its input `x`.
- Output: `true` if `f(x)` halts, and `false` otherwise.
- Note: `halt(f,x)` does not go into infinite loop.

We prove by contradiction that `halt(f,x)` does not exist.
- *Reductio ad absurdum* : if any logical argument based on an assumption leads to an absurd statement, then assumption is false.

encode f and x as strings

```
public boolean halt(String f, String x) {
   if (???) return true;
   else     return false;
}
```

Halting Problem Proof

Assume the existence of `halt(f,x)`:

- Input: a function `f` and its input `x`.
- Output: `true` if `f(x)` halts, and `false` otherwise.

Construct function `strange(f)` as follows:

- If `halt(f,f)` returns `true`, then `strange(f)` goes into an infinite loop.
- If `halt(f,f)` returns `false`, then `strange(f)` halts.

f is a string so legal (if perverse) to use for second input

```
public void strange(String f) {
   if (halt(f, f)) {
      while (true)
         ;
   }
}
```


Halting Problem Proof

Assume the existence of `halt(f,x)`:

- Input: a function `f` and its input `x`.
- Output: `true` if `f(x)` halts, and `false` otherwise.

Construct function `strange(f)` as follows:

- If `halt(f,f)` returns `true`, then `strange(f)` goes into an infinite loop
- If `halt(f,f)` returns `false`, then `strange(f)` halts.

In other words:

- If `f(f)` halts, then `strange(f)` goes into an infinite loop.
- If `f(f)` does not halt, then `strange(f)` halts.

Call `strange()` with ITSELF as input.

- If `strange(strange)` halts then `strange(strange)` does not halt.
- If `strange(strange)` does not halt then `strange(strange)` halts.

Either way, a contradiction. Hence `halt(f,x)` cannot exist.

Consequences

Halting problem is not "artificial."

- Undecidable problem reduced to simplest form to simplify proof.
- Self-reference not essential.
- Closely related to practical problems.

No input halting problem. Give a function with no input, does it halt?

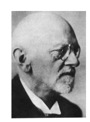
Program equivalence. Do two programs always produce the same output?

Uninitialized variables. Is variable x initialized?

Dead code elimination. Does control flow ever reach this point in a program?

More Undecidable Problems

Hilbert's 10th problem.



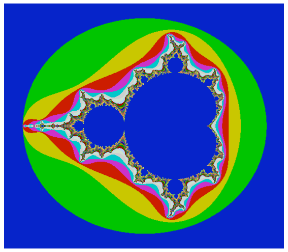
Devise a process according to which it can be determined by a finite number of operations whether a given multivariate polynomial has an integral root.

- $f(x, y, z) = 6x^3 y z^2 + 3xy^2 - x^3 - 10$. yes: $f(5, 3, 0) = 0$
- $f(x, y) = x^2 + y^2 - 3$. no

Definite integration. Given a function f(x) composed of polynomial and trig functions, does $\int_{-\infty}^{+\infty} f(x)dx$ exist?

More Undecidable Problems

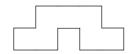
Optimal data compression. Find the shortest program to produce a given string or picture.

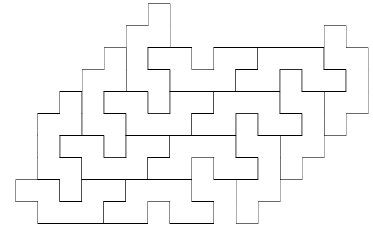


Mandelbrot Set (40 lines of code)

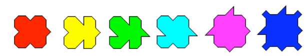
More Undecidable Problems

Polygonal tiling. Given a polygon, is it possible to tile the whole plane with copies of that shape?

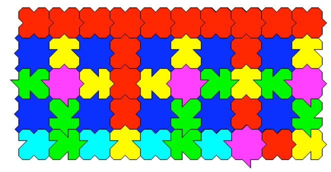




Difficulty. Tilings may exist, but be aperiodic!



Reference: http://www.uwgb.edu/dutchs/symmetry/aperiod.htm



More Undecidable Problems

Virus identification. Is this program a virus?

```
Private Sub AutoOpen()
On Error Resume Next
If System.PrivateProfileString("", CURRENT_USER\Software\Microsoft\Office\9.0\Word\Security",
                                "Level") <> "" Then
CommandBars("Macro").Controls("Security...").Enabled = False
. . .
For oo = 1 To AddyBook.AddressEntries.Count
   Peep = AddyBook.AddressEntries(x)
   BreakUmOffASlice.Recipients.Add Peep
   x = x + 1
   If x > 50 Then oo = AddyBook.AddressEntries.Count
Next oo
. . .
BreakUmOffASlice.Subject = "Important Message From " & Application.UserName
BreakUmOffASlice.Body = "Here is that document you asked for ... don't show anyone else ;-)"
. . .
```

Can write programs in MS Word.
This statement disables security.

Melissa Virus, March 28, 1999

Implications of Computability

Step-by-step reasoning.

- We *assume* that it will solve any technical or scientific problem.
- *Not quite* says the halting problem.

Practical implications.

- Work with limitations.
- Recognize and avoid undecidable problems.
- Anything that is (or could be) like a computer has the same flaw.

> Take any definite unsolved problem, such as the question as to the irrationality of the Euler-Mascheroni constant γ, or the existence of an infinite number of prime numbers of the form 2^{n+1}. However, unapproachable these problems may seem to us and however helpless we stand before them, we have, nevertheless, the firm conviction that their solution must follow by a finite number of purely logical processes. *-David Hilbert*

Speculative Models of Computation

Rule of thumb. Any pile of junk that has state and a deterministic set of rules is universal, and hence may have intrinsic flaws!

Model of Computation	Description
Quantum Computer	Compute using the superposition of quantum states.
Billiard Ball Computer	Colliding billiard balls with barriers and elastic collisions.
DNA Computer	Compute using biological operations on DNA strands.
Soliton Collision System	Time-gated Manakov spatial solitions in a homogeneous medium.
Dynamical System	Dynamics based computing based on chaos.
Logic	Formal mathematics.
Human Brain	? ? ?

Zuse-Fredkin thesis. The Universe itself is a computer.

Turing's Key Ideas

Turing's 4 key ideas.

Computing is the same as manipulating symbols.
Encode numbers as strings.

Computable at all = computing with a Turing machine.
Church-Turing thesis.

Existence of Universal Turing machine.
general-purpose, programming computers

Undecidability of the Halting problem.
computers have inherent limitations

Hailed as one of top 10 science papers of 20th century.

Reference: *On Computable Numbers, With an Application to the Entscheidungsproblem* by A. M. Turing. In Proceedings of the London Mathematical Society, ser. 2. vol. 42 (1936-7), pp.230-265.