7.6 Universality

Fundamental Questions

Universality. What is a general purpose computer?

Computability. Are there problems that no machine can solve?

Church-Turing thesis. Are there limits on the power of machines that we can build?

Pioneering work in the 1930's.

- (Princeton == center of universe).
- Hilbert, Gödel, Turing, Church, von Neumann.
- Automata, languages, computability, universality, complexity, logic.

Universality

Q. Which one of the following does not belong?


Cray


Dell PC


iMac


Espresso maker


Palm Pilot

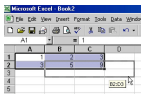

Xbox


Tivo


Turing machine


TOY

Java language


MS Excel


Java cell phone

Quantum computer


DNA computer

Python language

Java: As Powerful As Turing Machine

Turing machines are equivalent in power to TOY and Java.

- Can use Java to solve any problem that can be solved with a TM.
- Can use TM to solve any problem that can be solved with a TOY.
- Can use TOY to solve any problem that can be solved with Java.

Java simulator for Turing machines.

```
State state = start;
while (true) {
   char c = tape.readSymbol();
   tape.write(state.symbolToWrite(c));
   state = state.next(c);
   if      (state.isLeft())  tape.moveLeft();
   else if (state.isRight()) tape.moveRight();
   else if (state.isHalt())  break;
}
```

Turing Machine: As Powerful As TOY Machine

Turing machines are equivalent in power to TOY and Java.

- Can use Java to solve any problem that can be solved with a TM.
- Can use TM to solve any problem that can be solved with a TOY.
- Can use TOY to solve any problem that can be solved with Java.

Turing machine simulator for TOY programs.

- Encode state of memory, registers, pc, onto Turing tape.
- Design TM states for each instruction.
- Can do because all instructions:
  - examine current state
  - make well-defined changes depending on current state

TOY: As Powerful As Java

Turing machines are equivalent in power to TOY and Java.

- Can use Java to solve any problem that can be solved with a TM.
- Can use TM to solve any problem that can be solved with a TOY.
- Can use TOY to solve any problem that can be solved with Java.

TOY simulator for Java programs.

- Variables, loops, arrays, functions, linked lists, ...
- In principle, can write a Java-to-TOY compiler!

Java, Turing Machines, and TOY

Turing machines are equivalent in power to TOY and Java.

- Can use Java to solve any problem that can be solved with a TM.
- Can use TM to solve any problem that can be solved with a TOY.
- Can use TOY to solve any problem that can be solved with Java.

Also works for:

- C, C++, Python, Perl, Excel, Outlook, ...
- Mac, PC, Cray, Palm pilot, ...
- TiVo, Xbox, Java cell phone, ...

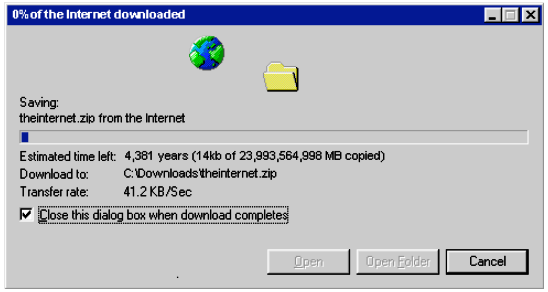
Does not work:

- DFA or regular expressions.
- Gaggia espresso maker.

Not Enough Storage?

Implicit assumption.

- TOY machine and Java program have unbounded amount of memory.
- Otherwise Turing machine is strictly more powerful.
- Is this assumption reasonable?



Universal Turing Machine

Java program: solves one specific problem.
TOY program: solves one specific problem.
TM: solves one specific problem.

Java simulator in Java: Java program to simulate any Java program.
TOY simulator in TOY: TOY program to simulate any TOY program.
UTM: Turing machine that can simulate any Turing machine.

General purpose machine.

- UTM can implement any algorithm.
- Your laptop can do any computational task: word-processing, pictures, music, movies, games, finance, science, email, Web, …

Church-Turing Thesis

Church Turing thesis (1936). Turing machines can do anything that can be described by a physically harnessable process of the universe.

Implications:

- No need to seek more powerful machines.
- If a yes-no problem can't be solved with a Turing machine, then it can't be solved on any physical computing device.

Remarks.

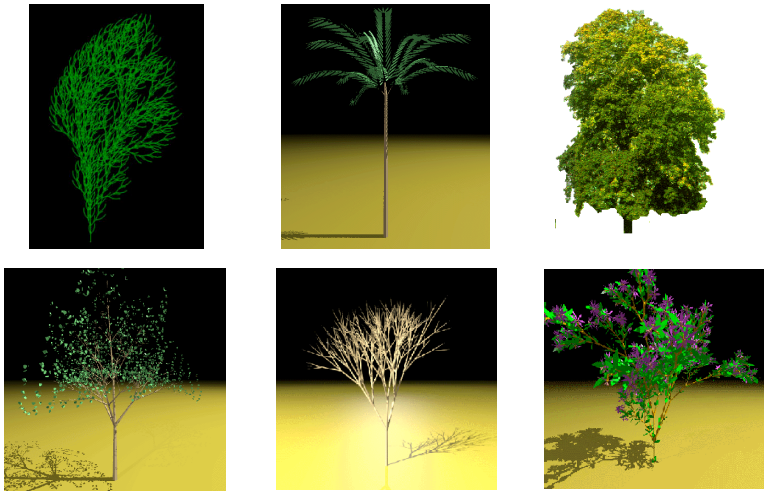
- "Thesis" and not a mathematical theorem because it's a statement about the physical world and not subject to proof.

Turing machine: a simple and universal model of computation.

Other Universal Models of Computation

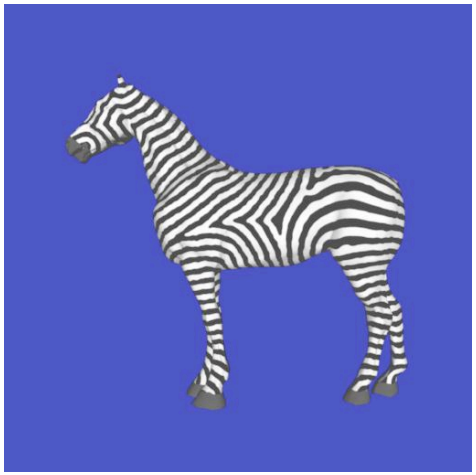
Model of Computation	Description
Enhanced Turing Machines	Multiple heads, multiple tapes, 2D tape, nondeterminism.
Untyped Lambda Calculus	A method to define and manipulate functions. Basis of functional programming language like Lisp and ML.
Recursive Functions	Functions dealing with computation on natural numbers.
Unrestricted Grammars	Iterative string replacement rules used by linguists to describe natural languages.
Extended L-Systems	Parallel string replacement rules that model the growth of plants.
Cellular Automata	Boolean array of cells whose values change according only to the state of the adjacent cells, e.g., Game of Life.
Random Access Machines	Finitely many registers plus memory that can be accessed with an integer address. TOY, G5, Pentium IV.
Programming Languages	Java, C, C++, Perl, Python, PHP, Lisp, PostScript, Excel

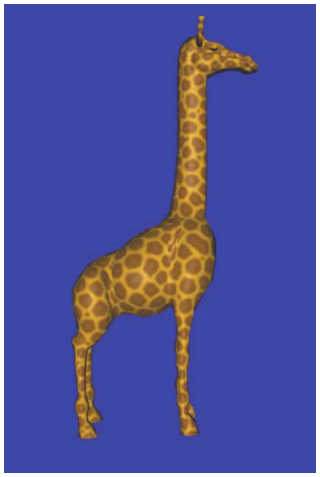
Lindenmayer Systems: Synthetic Plants



Reference: http://astronomy.swin.edu.au/~pbourke/modelling/plants/

Cellular Automata: Synthetic Zoo





Reference: *Generating Textures on Arbitrary Surfaces Using Reaction-Diffusion*" by Greg Turk, SIGGRAPH, 1991.

Implicit Physical Principles

Turing machine: embodies physical constraints to which all concrete computational processes are subjected.

Fredkin-Toffoli axioms governing computational processes (partial list).

- *The speed of propagation of information is bounded.*
  - TM head only move to adjacent cells
- *The amount of information which can be encoded in the state of a finite system is bounded.*
  - TM stores finitely many symbols per tape cell
- *It is possible to construct … physical devices which perform in a recognizable and reliable way the logical functions AND, OR, and FAN-OUT.*
  - can fabricate a TM out of physical parts, and run it reliably

Reference: E. F. Fredkin and T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3/4):219--253, 1982.