A New Computation Model for Reals

Yang Yue

Department of Mathematics
National University of Singapore

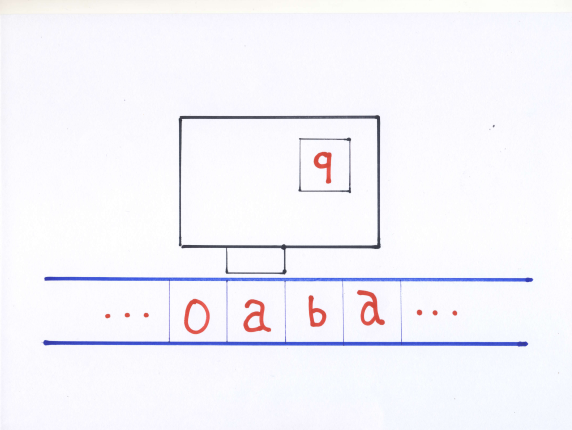
April 12, 2014

Outline

Main Question

- Turing machine is THE model for effective computations.
- ...but works only on (objects coded by) natural numbers.
- What about algorithms on real or complex numbers?
- Q: Does effectiveness make sense beyond natural numbers?

Standard Turing Machines (I): Physical Device



Standard Turing Machines (II): Program

3 types of commands:

- $q\ a\ b\ q'$: When in state q and seeing a, change a to b and change state from q to q'.
- $q\ a\ a\ L$: When in state q and seeing a, move left.
- $q\ a\ a\ R$: When in state q and seeing a, move right.

The Success of Turing Machine

- It fits the intuition of working mathematicians.
- It has natural “computation steps”, which leads to complexity theory.
- It has relativized versions, that is, oracle machines, which leads to degree theory.
- Its computability corresponds to definability in arithmetic.
- The success of Turing machine contrasts with models computing Reals.

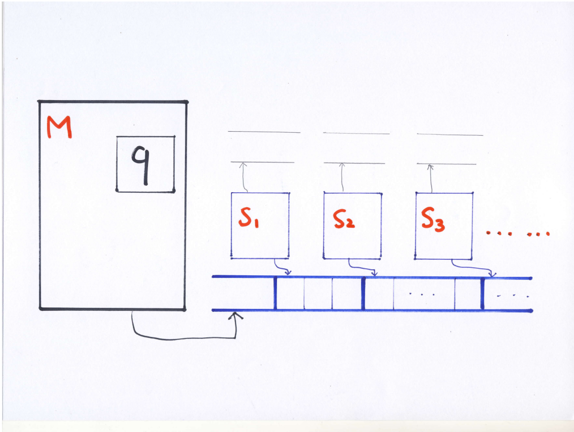
The Challenge (I)

- Algorithms must have an “finite character”.
- Real numbers are truly infinite objects.
- How to balance between finite and infinite?

The Challenge (II)

- Computations are “discrete”.
- Computations on Real numbers often use approximation, which based on continuity.
- How to balance between discreteness and continuity?

Master-Slave Machines (I): Physical Device



Master-Slave Machines (II): Program

Besides standard commands, add the following two new types:

- $q\ a\ P\ q'$, where P is a master command: P is a recursive function and the i-th slave will execute the program coded by $P(i)$.
- Note that P can be coded by a natural number e.
- E-command: $q_0 E q_1 q_2$, also called zero-test command: if all cells on tape are 0, then change state to q_1, else q_2.

Conventions

- We view each cell as an unlimited register and holding a rational number.
- We further assume that we have countably extra working tapes.
- Since our main target is computation on $\mathbb{R}$, we fully employ the Church thesis about computation on $\mathbb{N}$.

Computation on Reals

- A real number x is represented by any Cauchy sequence $\langle r_i : i \in \omega \rangle$ of rational numbers such that $\lim_i r_i = x$ with a fixed rate, e.g., for all $m, n > i$, $|r_m - r_n| < 2^{-i}$.
- Note we do not require that r_i is a recursive sequence of rationals.
- (Other typical representations are convertible.)

The Definition

Definition

We say that a partial function $f : \mathbb{R} \to \mathbb{R}$ is $\mathbb{R}$-Turing computable (or $\mathbb{R}$-computable) if there is a Master-Slave-machine M such that

$$f(x) = \begin{cases} y, & \text{if } M \text{ on input on any rep. of } x \text{ halts and the output is a rep. of } y; \\ \text{undefined}, & \text{if } M \text{ on input on any rep. of } x \text{ never halt.} \end{cases}$$

Definition

$S \subseteq \mathbb{R}$ is called $\mathbb{R}$-recursive or (strongly) $\mathbb{R}$-computable if its characteristic function is $\mathbb{R}$-computable.

Examples

Theorem

Most functions occurring in scientific computing are $\mathbb{R}$-computable, for example, 2^x, $\ln x$, $\sin x$, $\tan x$, $[x]$ are all $\mathbb{R}$-computable, so is equality.

Theorem

The set of natural numbers $\mathbb{N}$ is $\mathbb{R}$-computable.

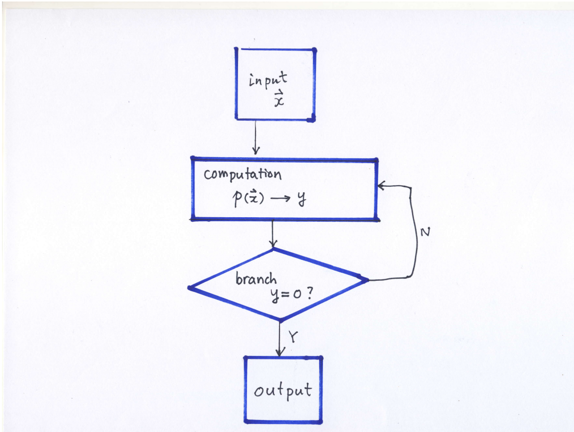
Theorem

The standard halting set $K = \{e \in \mathbb{N} : \varphi_e(e) \downarrow\}$ is $\mathbb{R}$-computable.

Remarks on “Super Features”

- The number of slaves is indeed infinite, this is one place where we step into infinity.
- Zero-test command E is another.
- Justification: All driven by the input, which is the only source of infinity given to us by our opponent.
- If the input x is a finite string s, the tasks become standard, thus we need no slaves.

BSS machine in a nutshell



Simulating BSS by Master-Slave

- Need to consider functions $\mathbb{N} \times R \to R$.
- Two slaves for the ring operations $(+, \times)$ and the E-command.
- We have the notion of r-computable, r for “ring".

General r-Recursive Functions

Define the class of general r-recursive functions from $\mathbb{N} \times R^n$ to R to be the smallest one containing

- zero, successor (for $\mathbb{N}$), projections (for R)
- ring operations in R
- zero-test in R

and is closed under

- composition
- primitive recursion w.r.t. $\mathbb{N}$
- μ-operation w.r.t. $\mathbb{N}$.

A Normal Form Theorem for Ring Computation

Theorem (Ng, Tavana and Y.)

f is r-computable iff f is general r-recursive.

The proof actually gives us more information on r-computation and offers us a normal form.

Note the role played by $\mathbb{N}$ and Turing control.

TTE in a nutshell

- Idea: Use approximations
- $$\begin{array}{ccc} 2^{<\omega} & \xrightarrow{f} & 2^{<\omega} \\ \downarrow & & \downarrow \\ 2^{\omega} & \xrightarrow{F} & 2^{\omega} \end{array}$$
- All TTE computable functions are continuous.
- Note: Each of them was induced by a single Master command.

General $\mathbb{Q}^\omega$-Recursive Functions

Define the class of general $\mathbb{Q}^\omega$-recursive functions from $\mathbb{N} \times R^n$ to R (where $R = \mathbb{Q}^\omega$) to be the smallest one containing

- zero, successor, projections
- TTE computable functions R
- zero-test in R

and is closed under

- composition
- primitive recursion w.r.t. $\mathbb{N}$
- μ-operation w.r.t. $\mathbb{N}$.

A Normal Form Theorem for $\mathbb{Q}^\omega$-computation

Theorem (Ng, Tavana and Y.)

f is general $\mathbb{Q}^\omega$-recursive iff it can be computed by some Master-Slave machine.

(Remark: To replace $\mathbb{Q}^\omega$ by $\mathbb{R}$ is still under study.)

A Church Thesis?

- Classical Church Thesis (on $\mathbb{N}$): Intuitively computable is Turing computable.
- Can we have something similar for $\mathbb{R}$, or $\mathbb{Q}^\omega$?
- (My guess is not yet, e.g., we need to ask working mathematicians.)

Background for Philosophical Discussions

- Hilbert Program: Is there a finitary way to show the consistency of mathematics?
- Gödel's incompleteness theorem: No recursive way.
- Turing: No mechanically computable way.

Limit for Machines; and for Minds?

- It is generally agreed that the limit of Turing machines is the limit of any finite mechanical devices.
- Q: Does this limit also apply to human mind?
- Turing (in words from *Gödel 1972a*): Mental procedures cannot go beyond mechanical procedures.

Quote from *Turing 1937*

The behaviour of the computer at any moment is determined by the symbols which he is observing, and his “state of mind” at that moment.

We may suppose that there is a bound B to the number of symbols or squares which the computer can observe at one moment...

We will also suppose that the number of states of mind which need be taken into account is finite.

The reasons for this are...If we admitted an infinity of states of mind, some of them will be “arbitrarily close” and will be confused.

Quote from *Gödel 1972a*

Gödel called it "A philosophical error in Turing's work".

What Turing disregards completely is the fact that *mind, in its use, is not static, but constantly developing*, i.e., that we understand abstract terms more and more precisely as we go on using them,...

Therefore, although at each stage the number and precision of the abstract terms at our disposal may be *finite*, both (and, therefore, also Turing's number of *distinguishable states of mind*) may *converge toward infinity* in the course of the application of the procedure.

Gödel's Abstract Terms

Gödel's *1972a* was a footnote of his *1972* "On an extension of finitary mathematics which has not yet been used", which is a revised version of his *1958*.

Abstract: "P. Bernays has pointed out that, even in order to prove only the consistency of classical number theory, it is necessary to extend Hilbert's finitary standpoint.

He suggested admitting certain abstract concepts in addition to the combinatorial concepts referring to symbols. The abstract concepts that so far have been used for this purpose are those of the constructive theory of ordinals and those of intuitionistic logic."

1972 Abstract (continued)

"It is shown that a certain concept of computable function of finite simple type over the natural numbers can be used instead,

where no other procedures of constructing such functions are necessary except primitive recursion by a number variable and definition of a function by an equality with a term containing only variables and/or previously introduced functions beginning with the function +1."

Final Remarks

- Master-Slave machine can compute standard halting problem.
- Internally, it is *finite*.
- The extra power was due to raising the type and allowing external extensions.
- Would we think of it as "effective"?
- Would it be closer to human mind?