Turing - Computability

Recall: "URM-computable" is closed under many useful operations.

The class of primitive recursive functions is URM-computable

zero, successor, projection, composition, recursion applied some finite number of times

Today: Is this all? Are there other computable functions outside this class?

What sort of functions are computable?

What sort of predicates are decidable?

If $R(x)$ and $S(x)$ are decidable predicates, so are

(i) not $R(x)$

(ii) $R(x)$ and $S(x)$

(iii) $R(x)$ or $S(x)$

$$f_R(x) = \begin{cases} 1, & \text{if } R(x) \text{ holds} \\ 0, & \text{if } R(x) \text{ does not hold} \end{cases}$$

Exercise: What are the corresponding computable functions?

$$g(x) = \begin{cases} 1, & x = 0 \\ 0, & x = 1 \end{cases}$$

$$g(f_R(x))$$

What sort of functions are computable?

Is exponentiation $\exp(x, y) = x^y$ computable?

$\exp(x, 0) = 1$

$\exp(x, y+1) = \mathrm{mult}(x, \exp(x, y))$

$\mathrm{mult}(x, 0) = 0$

$\mathrm{mult}(x, y+1) = x + \mathrm{mult}(x, y) \quad \mathrm{add}(x, \mathrm{mult}(x, y))$

What sort of functions are computable?

Is exponentiation $\exp(x,y) = x^y$ computable?

$$\exp(x,0) = 1$$

$$\exp(x,y+1) = \mathrm{mult}(x, \exp(x,y))$$

$$\mathrm{mult}(x,0) = 0$$

we already wrote a URM program for +

$$\mathrm{mult}(x,y+1) = x + \mathrm{mult}(x,y)$$

What about a really fast-growing function?

$$\mathrm{tow}(n) = 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \Big\} \text{height } n$$

Is this computable?

$$\mathrm{tow}(0) = 1$$

$$\mathrm{tow}(n+1) = \exp(2, \mathrm{tow}(n))$$

What about a really fast-growing function?

$$\mathrm{tow}(n) = 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\Big\}\text{ height } n$$

Is this computable? $\mathrm{tow}(0) = 2$

$$\mathrm{tow}(n+1) = \exp(\mathrm{tow}(n), 2)$$

So URMs can compute some very powerful primitive recursive functions

Are there other computable functions?

There are! (Can be shown via Gödel numbering and diagonalization)

Computable = partial recursive functions. — Dovetails neatly into lambda calculus, logic, definability, computability hierarchy...

primitive recursive + "the least y s.t. $f(\bar{x}, y) = 0$"

Is there a non-computable function?

This is easier to do in a different (but equivalent!) model of computation.

A Turing Machine is a finite-state automaton with an infinite tape.

DFA/NFA : Finitely many states, "one state memory"

PDA : Finitely many states, stack (LIFO)

TM : Finitely many states, tape (bidirectional)

Machines as language acceptors

* Think of the tape like serially-linked registers without indexing

How does one access a register without indexing?

With a tape head that can move left or right.

https://samwho.dev/turing-machines/