COMPUTABILITY (10/10/2023)

* Models of computation ?

→ Turing machines

→ λ-calculus (Church)

→ partial recursive functions (Gödel - Kleene)

→ canonical deduction systems (Post)

→ URM (Unlimited Register Machine)

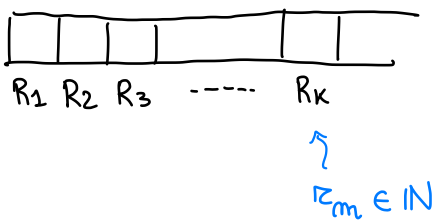
⋮

Church Turing Thesis

A function is computable by an effective procedure if and only if it is computable by a Turing machine

* Unlimited Register Machines

→ memory (unbounded)



→ executes a program : finite list of instructions

I_1
I_2
⋮
I_s

instruction set

→ Arithmetic instructions

- zero $\quad Z(n) \quad r_n \leftarrow 0$
- successor $\quad S(n) \quad r_n \leftarrow r_n + 1$
- transfer $\quad T(m,n) \quad r_n \leftarrow r_m$

→ Jump

$J(m, n, t)$ $\quad r_m = r_n$?
- yes: Jump to I_t
- no: continue with next instruction

* Computation

starts from
- initial configuration of registers
- executes I_1

terminates if the instruction to be executed next does not exist
→ last instruction
→ jump out of the program

Example :

		R_1	R_2	R_3
I_1	$J(2,3,5)$	1	2	0
I_2	$S(1)$	2	2	0
I_3	$S(3)$	2	2	1
I_4	$J(1,1,1)$	3	2	2

. .

* A computation can "diverge" (not terminate)

$I_1 \quad J(1,1,1)$

* Notation : Given $a_1, a_2, \dots \in \mathbb{N}$ and a program P

$P(a_1\, a_2 \dots)$ indicates the computation of P from $a_1, a_2, \dots$

- $P(a_1\, a_2 \dots)\downarrow$ eventually terminates
- $P(a_1\, a_2 \dots)\uparrow$ diverges

Given $a_1 \dots a_k \in \mathbb{N}$

$P(a_1 \dots a_k)$ denotes $P(a_1 \dots a_k\ 0\ 0 \dots)$

$P(a_1 \dots a_k)\downarrow a$ for $P(a_1 \dots a_k)\downarrow$
and in final configuration $r_1 = a$

URM - computable function

Given $f : \mathbb{N}^k \to \mathbb{N}$ (possibly partial) is URM - computable if there is a URM program P such that $\forall (a_1, \dots, a_k) \in \mathbb{N}^k$ $\forall a \in \mathbb{N}$

$$P(a_1 \dots a_k) \downarrow a \quad \text{iff} \quad (a_1, \dots, a_k) \in dom(f)$$
$$\text{and} \quad f(a_1 \dots a_k) = a$$

$$\mathcal{C}^{(k)} = \{ f \mid f : \mathbb{N}^k \to \mathbb{N} \text{ computable (URM)} \}$$
$$\mathcal{C} = \bigcup_{k \geq 1} \mathcal{C}^{(k)}$$

Example

* $f : \mathbb{N}^2 \to \mathbb{N}$

$f(x, y) = x + y$

R_1	R_2	R_3	
x	y	0	...
$x+k$		k	

R_1	
$x+y$	...

```
LOOP: J(2,3, STOP)        // k=y ?
      S(1)
      S(3)
      J(1,1, LOOP)
STOP:
```

* $g : \mathbb{N} \to \mathbb{N}$

$$g(x) = x \dot{-} 1 = \begin{cases} 0 & \text{if } x = 0 \\ x-1 & \text{otherwise} \end{cases}$$

R_1	R_2	R_3	
x	0	0	
	$k+1$	k	

```
      J(1,2, END)         x=0 ?
      S(2)
LOOP: J(1,2, RES)         x=k+1 ?
      S(2)
      S(3)
      J(1,1,LOOP)
RES:  T(3,1)
END:
```

$*\quad h : \mathbb{N} \to \mathbb{N}$

$$h(x) = \begin{cases} x/2 & \text{if } x \text{ even} \\ \uparrow & \text{otherwise} \end{cases}$$

R_1	R_2	R_3
x	0	0
	k	$2k$

```
LOOP:  J(1, 3, RES)
       S(2)
       S(3)
       S(3)
       J(1,1, LOOP)
RES:   T(2, 1)
```

$*$ <u>Function computed by a program</u>

given a program P and $k \geq 1$

$$f_P^{(k)} : \mathbb{N}^k \to \mathbb{N}$$

$$f_P^{(k)}(a_1, \dots, a_k) = \begin{cases} a & \text{if } P(a_1, \dots, a_k) \downarrow a \\ \uparrow & \text{if } P(a_1, \dots, a_k) \uparrow \end{cases}$$

$*$ <u>Question</u>: given $f : \mathbb{N}^k \to \mathbb{N}$ how many programs computing f?

0 or infinitely many

<u>EXERCISE</u> Consider URM^- machine without $T(m,n)$

$\mathcal{C}^-$ = class of URM^- computable functions

$$\mathcal{C} \overset{?}{=} \mathcal{C}^-$$

$\supseteq$

$\subseteq$

$T(m,n)$

```
       Z(n)
LOOP:  J(m,n, DONE)
       S(n)
       J(1,1, LOOP)
```

<u>proof</u>

$(\mathcal{C}^- \subseteq \mathcal{C})$ Let $f : \mathbb{N}^k \to \mathbb{N}$ computable in URM^- $f \in \mathcal{C}^-$

i.e. there is P in URM^- s.t. $f = f_P^{(k)}$

just observe that P is also a URM program $\Rightarrow f \in \mathcal{C}$

$(\mathcal{C} \subseteq \mathcal{C}^-)$ Let $f \in \mathcal{C}$ $\quad f : \mathbb{N}^k \to \mathbb{N}$

hence there is P URM program such that $f = f_P^{(k)}$

assume P to be well-formed : if it terminates it does at last instruction $+1$

without loss of generality (see below)

We show that there exists P' URM$^-$ machine such that

$$f_{P'}^{(k)} = f_P^{(k)} = f$$

by induction on h = number of $T(m,n)$ instructions in P

$(h=0)$ P with no transfer instructions is already a URM$^-$ prg.

hence $P' = P$

$(h \to h+1)$ Let P be URM program with $h+1$ transfer instr.

$$P \left\{ \begin{array}{ll} I_1 & \\ I_2 & \\ \vdots & \\ I_t & T(m,n) \\ \vdots & \\ I_s & \end{array} \right. \quad \rightsquigarrow \quad P'' \left\{ \begin{array}{ll} I_1 & \\ \vdots & \\ I_t & J(1,1,s+2) \\ & \\ I_s & \\ I_{s+1} & J(1,1,\mathrm{END}) \\ I_{s+2} & Z(n) \\ \mathrm{LOOP}: & J(m,n,t+1) \\ & S(n) \\ & J(1,1,\mathrm{LOOP}) \end{array} \right.$$

Now P'' has h transfer instructions

and $\quad f_P^{(k)} = f_{P''}^{(k)} \quad (*)$

By inductive hypothesis there P' URM$^-$ program such that $f_{P'}^{(k)} = f_{P''}^{(k)}$ $(*)$

Putting things together

$$f_P^{(k)} \underset{\uparrow (*)}{=} f_{P''}^{(k)} \underset{\uparrow (*)}{=} f_{P'}^{(k)} \quad \text{with } P' \text{ URM}^- \text{ program} \qquad \square$$

Note: for any URM-program P there is a well formed program P' computing the same function

in fact I_1

$\vdots$ $J(m,n,t)$ if $t > s$ replace it with $J(m,n,s+1)$

I_s

* Exercise: Variant URM^s machine

$T(\cancel{m,n})$ $T^s(m,n)$ $r_m \leftrightarrow r_n$

exchange content of registers

$$\mathcal{C}^s \overset{?}{=} \mathcal{C}$$

* EXERCISE: Consider $URM^=$ without jump

$$\mathcal{C}^= \overset{?}{=} \mathcal{C}$$

$\subsetneq$

try to characterise $\mathcal{C}^=$

(shape of functions in there)