COMPUTABILITY (06/10/2026)

* Model of computing

→ 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 iff it is computable by a Turing machine by an effective procedure

* Unlimited Register Machines

→ memory (unbounded)



$r_m \in \mathbb{N}$

→ executes a program : finite list of instructions

I_1
I_2
⋮
I_s

taken from an instruction set

→ Arithmetic instructions

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

→ conditional jump

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

* Computation

starts from { initial configuration of registers; executes I_1 }

terminates when trying to execute non-existing instruction

→ 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	...

↑ unconditional jump

* A computations can "diverge" (do not terminate)

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

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

$P(a_1, a_2, \ldots)$ indicates computation of P starting on $a_1, a_2, \ldots$

$P(a_1, a_2, \ldots) \downarrow$ eventually terminates

$P(a_1, a_2, \ldots) \uparrow$ diverges

Given $a_1, \ldots, a_k \in \mathbb{N}$

$P(a_1, \ldots, a_k)$ denotes $P(a_1, \ldots, a_k, 0, \ldots 0, \ldots)$

$P(a_1, \ldots, a_k) \downarrow a$ if $P(a_1, \ldots a_k) \downarrow$ and at the end $r_1 = a$

URM - computable function

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

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

$\mathcal{C}^{(k)} = \{ f \mid f: \mathbb{N}^k \to \mathbb{N}$ URM-computable $\}$

$\mathcal{C} = \bigcup_{k \geq 1} \mathcal{C}^{(k)}$

Examples

(1) $f: \mathbb{N}^2 \to \mathbb{N}$

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

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

≈

R_1	
$x+y$	...

```
LOOP : J(2,3,STOP)      // k=y ?   yes  return R1
       S(3)                         no   continue
       S(1)
       J(1,1,LOOP)
STOP :
```

(2) $g: \mathbb{N} \to \mathbb{N}$

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

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

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

* always undefined function

$\emptyset : \mathbb{N} \to \mathbb{N}$ computed by $J(1,1,1)$

$\emptyset(x) \uparrow$

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

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

R_1	R_2	R_3
x	0	0

$2K$ K

$\uparrow$

LOOP : $J(1,2,RES)$

$S(3)$
$S(2)$
$S(2)$
$J(1,1,LOOP)$

RES : $T(3,1)$

* Function computed by a program

given program P and $K \geq 1$

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

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

example :

P $T(2,1)$

$f_P^{(1)} : \mathbb{N} \to \mathbb{N}$

$f_P^{(1)}(x) = 0 \quad \forall x$

R_1	R_2
x	0

$f_P^{(2)}(x,y) = y$

R_1	R_2
x	y

Def : $f : \mathbb{N}^K \to \mathbb{N}$ is computable

if there is P URM - program s.t. $f = f_P^{(K)}$

(same def. as before :)

* Question : Given $f: \mathbb{N}^k \to \mathbb{N}$. how many programs computing f ?

0 or infinitely many

EXERCISE : Consider URM$^-$ machine where there is no $T(m,n)$ instruction

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

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

$\supseteq$

$\subseteq$



$T(m,n)$

$Z(m)$
LOOP : $J(m,m,\text{DONE})$
$S(m)$
$J(1,1,\text{LOOP})$

proof

($\mathcal{C}^- \subseteq \mathcal{C}$) Let $f: \mathbb{N}^k \to \mathbb{N}$ such that $f \in \mathcal{C}^-$. Then there is a URM$^-$ program P such that $f_P^{(k)} = f$. just observe that P is also a URM-program and conclude $f \in \mathcal{C}$

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

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

Assume that P well-formed : if it terminates, it ends into I_{s+1}.
(without loss of generality - see later observation)

We show by induction on h = number of $T(-,-)$ in P

that there exists P' URM$^-$-program such that $f_P^{(k)} = f_{P'}^{(k)}$

($h = 0$)

P has no transfer instructions

↳ P is a URM$^-$ program

we can take $P' = P$

$(h \to h+1)$ Let P be a URM program with $h+1$ transfer instructions

P { I_1 ⋮ I_t $T(m,n)$ ⋮ I_s } ⇝

P'' {
I_1
⋮
I_t $J(1,1,SUB)$
⋮
I_s
I_{s+1} : $J(1,1,END)$
$Z(n)$
LOOP : $J(m,n,t+1)$
$S(n)$
$J(1,1,LOOP)$
END :
}

By construction P'' has h transfers and

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

By inductive hyp. on P'' there is P' URM-program such that

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

Summing up

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

and thus P' is the desired program. □

OBSERVATION: Given a URM-program P there is a well-formed program P' such that

$$f_P^{(k)} = f_{P'}^{(k)} \qquad \forall k \geq 1$$

P { I_1 ⋮ $J(m,n,t)$ ⋮ I_s } with $t > s$, replace it with $J(m,n,s+1)$

EXERCISE Variant of the URM^s machine

$T(m,n)$ replaced by $T_s(m,n)$ $r_m \leftrightarrow r_n$ exchanges content of R_m and R_n

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

EXERCISE: Consider URM$^=$ without $J(-,-,-)$

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

$$\subseteq$$

$\uparrow$

characterise functions in $\mathcal{C}^=$