COMPUTABILITY (16/10/2023)

EXERCISE : URM^s machine · variant of URM

$T(\cancel{m},n)$ $\qquad T_s(m,n)$ $\qquad r_m \leftrightarrow r_n$

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

proof

$(\mathcal{C} \subseteq \mathcal{C}^s)$ Given $f \in \mathcal{C}$ $f: \mathbb{N}^k \to \mathbb{N}$ $\rightsquigarrow$ $f \in \mathcal{C}^s$

if $f \in \mathcal{C}$ then there is P URM program s.t. $f_P^{(k)} = f$

We know that there is P' URM - program without $T(\ )$ instructions s.t. $f_{P'}^{(k)} = f_P^{(k)}$. But P' is also a URM^S - machine program $\rightsquigarrow$ $f = f_{P'}^{(k)} \in \mathcal{C}^s$.

$(\mathcal{C}^s \subseteq \mathcal{C})$ Take $f: \mathbb{N}^k \to \mathbb{N}$ $f \in \mathcal{C}^s$ and let P a URM^S program such that $f = f_P^{(k)}$. We want to "transform" P into a URM program P' s.t. $f_{P'}^{(k)} = f_P^{(k)}$

$T_s(m,n)$ $\rightsquigarrow$ $T(m,i)$ $\quad R_i$ not used in P
$T(n,m)$
$T(i,n)$

$$\boxed{\text{A } URM^S\text{ - program } P \text{ can be transformed into a URM - program } P' \text{ such that } f_P^{(k)} = f_{P'}^{(k)}}$$

We proceed by induction on h = number of T_s instructions in P

$(h=0)$ P is already a URM program, take $P' = P$

$(h \to h+1)$ Let P has $h+1$ T_s instructions

$$P \begin{cases} I_1 \\ \vdots \\ I_t \quad T_s(m,n) \\ \vdots \\ I_s \end{cases} \rightsquigarrow \quad P'' \begin{cases} I_1 \\ \vdots \\ I_t \quad J(1,1,SUB) \\ \vdots \\ I_s \\ I_{s+1} \quad J(1,1,END) \\ SUB: \ T(m,i) \\ T(n,m) \\ T(i,n) \\ J(1,1,t+1) \\ END: \end{cases}$$

We need

→ P alway terminates (if it does) at time $s+1$

→ $i = \max(\{m \mid R_m \text{ is used in } P\} \cup \{k\}) + 1$

Then $f_{P''}^{(k)} = f_P^{(k)}$ and P'' has h T_s instructions.

Hence by inductive hyp. there is a URM program P' s.t. $f_{P'}^{(k)} = f_{P''}^{(k)}$

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

i.e. $f \in \mathcal{C}$. □

The proof is wrong : I am using the inductive hyp. on P'' which is not a URM^S - program (it contains both T and T_S)

You can make it work by proving a stronger assertion

" Every program P which uses all instructions, including T and T_S can be transformed in a URM - program P' s.t. $f_P^{(k)} = f_{P'}^{(k)}$ "

EXERCISE : Consider URM$^=$ without Jump instructions

$$\mathcal{C}^= \subsetneq \mathcal{C}$$

proof

An URM$^=$ - program

$$P \begin{cases} I_1 \\ \vdots \\ I_s \end{cases}$$

$\ell(P) = s$ length of program P

P terminates after $\ell(P)$ steps

All functions in $\mathcal{C}^=$ are total ⇝ $\mathcal{C}^= \subsetneq \mathcal{C}$

e.g. $f : \mathbb{N} \to \mathbb{N}$

$f(x) \uparrow \quad \forall x \in \mathbb{N}$ $\qquad f \in \mathcal{C}$ $\quad J(1,1,1)$

$f \notin \mathcal{C}^=$ because it is not total

(saying "it uses jump" is not sufficient
e.g. $J(1,1,2)$ computes $f(x) = x$
$\in \mathcal{C}^=$)

$|x|0|\dots$ (restrict to unary functions)

functions of the shape

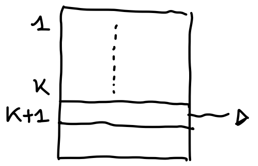
$$f(x) = c$$
or
$$f(x) = x + c$$
for c suitable constant

Denote $r_1(x, k)$ = content of R_1 after k-step of computation starting from $\underset{1}{|x|0|0|\dots}$

We prove by induction on k that $r_1(x,k) = \begin{cases} c \\ x+c \end{cases}$

$(k=0)$ $r_1(x,0) = x = x+0$ $c=0$ ok

$(k \to k+1)$ By inductive hyp. $r_1(x,k) = \begin{cases} c \\ x+c \end{cases}$ for $c \in \mathbb{N}$



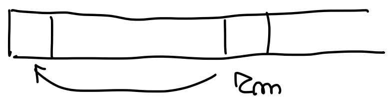
different cases according to the shape of I_{k+1}

3 cases

$I_{k+1} = Z(n)$ two subcases
- $n=1$ $r_1(x,k+1) = 0$
- $n>1$ $r_1(x, k+1) = r_1(x,k)$ ok, by ind. hyp.

$I_{k+1} = S(n)$
- $n=1$ $r_1(x,k+1) = r(x,k)+1$ ok, by ind. hyp
- $n>1$ $r_1(x,k+1) = r(x,k)$ " " " "

$I_{k+1} = T(m,n)$
- $m>1$ or $n=1$ $r_1(x, k+1) = r_1(x,k)$ ok by ind. hyp.
- $n=1$ and $m>1$



no hypotheses on r_m

I am lost....

<u>Idea</u> : $T(m,n)$ is "useless"

ok, but the argument uses jumps .... not working smoothely

The key observation is that the same property holds for all registers

$r_j(x, k)$ = content of R_j after k steps of computation starting from $\boxed{x \mid 0 \mid 0 \mid \ldots}$

Show by induction on k that for all k

$$r_j(x, k) = \begin{cases} c \\ x + c \end{cases} \quad \text{for } c \text{ suitable constant}$$

The proof goes smoothly.

(exercise)

for n-ary functions

$$f^{(n)}(x_1, \ldots, x_n) = \begin{cases} c \\ x_j + c \end{cases} \quad 1 \le j \le n, \quad c \in \mathbb{N} \text{ constant}$$

* <u>Decidable predicate</u>

$div(x, y)$ = " x divides y "

$div \subseteq \mathbb{N} \times \mathbb{N}$

$div = \{ (m, m * k) \mid m, k \in \mathbb{N} \}$

or

$div : \mathbb{N} \times \mathbb{N} \to \{ true, false \}$

<u>k-ary predicate</u> $\quad Q(x_1, \ldots, x_k) \subseteq \mathbb{N}^k$

$$Q : \mathbb{N}^k \to \{ \underset{\substack{\uparrow \\ 1}}{true}, \underset{\substack{\uparrow \\ 0}}{false} \}$$

<u>Def</u>. (decidable predicates):

Let $Q(x_1, \ldots, x_k) \subseteq \mathbb{N}^k$. We say that it is <u>decidable</u> if

$\chi_Q : \mathbb{N}^k \to \mathbb{N}$

$$\chi_Q(x_1, \ldots, x_k) = \begin{cases} 1 & \text{if } Q(x_1, \ldots, x_k) \\ 0 & \text{otherwise} \end{cases} \quad \text{is URM-computable}$$

Example: $Q(x_1, x_2) \subseteq \mathbb{N}^2$

$Q(x_1, x_2) = $ " $x_1 = x_2$ " decidable

$\chi_Q : \mathbb{N}^2 \to \mathbb{N}$

$$\chi_Q(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 = x_2 \\ 0 & \text{otherwise} \end{cases}$$

x_1	x_2	0	...

↑ output

$J(1, 2, \mathrm{TRUE})$

FALSE: $J(1, 1, \mathrm{RES})$

TRUE: $S(3)$

RES : $T(2, 1)$

Example: $Q(x) = $ " x is even " decidable

1	2	3	
x	0	0	

↑ k ↑ result

EVEN : $J(1, 2, \mathrm{YES})$
$S(2)$

ODD : $J(1, 2, \mathrm{NO})$
$S(2)$
$J(1, 1, \mathrm{EVEN})$

YES : $S(3)$

NO : $T(3, 1)$

* **Computability on other domains**

D countable

$\alpha : D \to \mathbb{N}$ bijective "effective"

(α^{-1} effective)

A^*, $\mathbb{Q}$, $\mathbb{Z}$, ...

$\cancel{\mathbb{R}}$

Given $f: D \to D$ function is <u>computable</u>

$$\begin{tikzcd} D \arrow[r, "f"] & D \arrow[d, "\alpha"] \\ \mathbb{N} \arrow[u, dotted, "\alpha^{-1}"] \arrow[r, "f^*"] & \mathbb{N} \end{tikzcd}$$

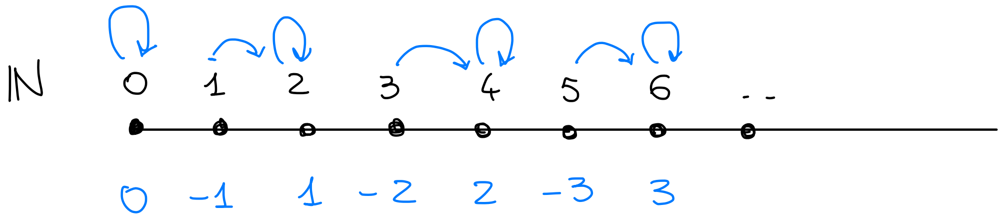
$f^* = \alpha \circ f \circ \alpha^{-1} : \mathbb{N} \to \mathbb{N}$

is URM-computable

<u>Example</u> : Computability in $\mathbb{Z}$

$\alpha : \mathbb{Z} \to \mathbb{N}$

$$\alpha(z) = \begin{cases} 2z & z \geq 0 \\ -2z-1 & z < 0 \end{cases}$$



$\alpha^{-1} : \mathbb{N} \to \mathbb{Z}$

$$\alpha^{-1}(n) = \begin{cases} \frac{n}{2} & n \text{ is even} \\ -\frac{n+1}{2} & n \text{ is odd} \end{cases}$$

$f : \mathbb{Z} \to \mathbb{Z}$

$f(z) = |z|$

computable

$f^* = \alpha \circ f \circ \alpha^{-1} : \mathbb{N} \to \mathbb{N}$

$$f^*(n) = \alpha \cdot f \cdot \underbrace{\alpha^{-1}(n)}_{} = \begin{cases} n \text{ even} & \alpha f\left(\frac{n}{2}\right) = \alpha\left(\frac{n}{2}\right) = 2\frac{n}{2} = n \\ n \text{ odd} & \alpha f\left(-\frac{n+1}{2}\right) = \alpha\left(\frac{n+1}{2}\right) = 2\frac{n+1}{2} = n+1 \end{cases}$$

$$= \begin{cases} n & \text{if } n \text{ even} \\ n+1 & \text{if } n \text{ is odd.} \end{cases}$$

$\uparrow$ URM-computable