COMPUTABILITY (16/11/2026)

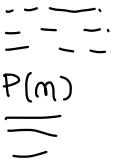
EXERCISE : URM^P with instructions

$Z(m)$
$T(m,m)$
$J(m,m,t)$
~~$S(m)$~~

$P(m)$ $\quad z_m \leftarrow z_m \dot{-} 1 = \begin{cases} 0 & \text{if } z_m = 0 \\ z_{m-1} & \text{otherwise} \end{cases}$

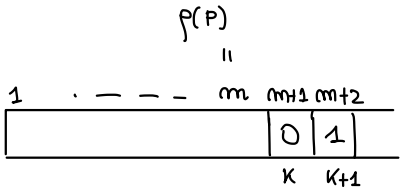
$\mathcal{C}^P \not\subseteq \mathcal{C}$

($\mathcal{C}^P \subseteq \mathcal{C}$) given a URM^P program



t: $P(m)$

$J(1,1,SUB)$



SUB : $Z(m+1)$
$Z(m+2)$
$J(m, m+1, END)$
$S(m+2)$

LOOP : $J(m, m+2, RES)$
$S(m+1)$
$S(m+2)$
$J(1,1,LOOP)$

RES : $T(m+1, m)$

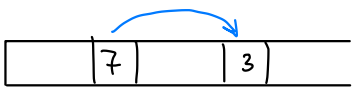
END : $J(1,1,t+1)$

FORMAL PROOF : using URM_p and URM instructions

Given any program P for URM_p machine, for every $K \in \mathbb{N}$, there is a URM program P' s.t. $f_P^{(K)} = f_{P'}^{(K)}$

(proof by induction on the number of predecessor instructions in P)

($\mathcal{C} \not\subseteq \mathcal{C}^P$) idea : you cannot increase registers



Given a program P of URM_P machine the maximum of registers after any number of steps t of computation of $P(\vec{x})$ is $\leq \max_{1 \leq i \leq k} x_i$

proof by induction on t

$(t=0)$ the memory is $\boxed{x_1 | \dots | x_k | 0 | 0 \dots}$ $\max_i r_i = \max_{1 \leq i \leq k} x_i$

$(t \to t+1)$ in fact $\boxed{x_1 | \dots | x_k | 0 | \dots}$

t steps $\}$

$\boxed{r_1' | r_2' | \dots}$ by ind. hyp

last step $\}$

$\boxed{r_1 | r_2 | \dots}$ $\max_i r_i' \leq \max_{1 \leq i \leq k} x_i$

case distinction depending on the instruction in last step

$Z(m)$
$T(m,n)$
$J(m,n,t)$
$P(m)$

no $\max_i r_i \leq \max_i r_i' \leq \max_{1 \leq i \leq k} x_i$

The successor function $S : \mathbb{N} \to \mathbb{N}$ $S(x) = x+1$,

$S \in \mathcal{C}$, $S \notin \mathcal{C}^P$ not URM^P computable

since $S(0) = 1$

$\boxed{0 | 0 | - - -}$

$\}$

$\boxed{1 |}$ $\max_i r_i \geq 1 > \max_{\| 0} x$ impossible

so $\mathcal{C}^P \not\subseteq \mathcal{C}$

EXERCISE : Termination is decidable for URM_P programs

hint: finite state

EXERCISE : Show that there is a total computable function $K : \mathbb{N} \to \mathbb{N}$ such that $E_{K(x)} = W_x \quad \forall x \in \mathbb{N}$

$P_x \overset{K}{\rightsquigarrow} P_{K(x)}$ set of outputs coincide with inputs where P_x halts

```
def P_{K(x)} (y):
    P_x(y)
    return y
```

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

$$f(x,y) = \begin{cases} y & \text{if } \varphi_x(y)\downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

$$= \underbrace{\mathbb{1}(\varphi_x(y))}_{\begin{array}{l}1 \text{ if } \varphi_x(y)\downarrow \\ \uparrow \text{ otherwise}\end{array}} * y$$

$$= \mathbb{1}(\psi_U(x,y)) * y$$

computable (composition of computable functions)

Hence by smn theorem there is $K : \mathbb{N} \to \mathbb{N}$ total computable s.t. $\forall x,y \in \mathbb{N}$

$$\varphi_{K(x)}(y) = f(x,y) = \begin{cases} y & \text{if } \varphi_x(y)\downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

Now K is the desired function : $W_x = E_{K(x)} \quad \forall x \in \mathbb{N}$

($W_x \subseteq E_{K(x)}$) let $y \in W_x$ i.e. $\varphi_x(y)\downarrow$ and this

$$\varphi_{K(x)}(y) = f(x,y) = y$$

i.e. $y \in E_{K(x)}$

($E_{K(x)} \subseteq W_x$) let $y \in E_{K(x)}$, i.e. there $z \in \mathbb{N}$ s.t. $\varphi_K(z) = y$

since $\varphi_{K(x)}$ is the identity when defined $z = y$ and $\varphi_x(y)\downarrow \rightsquigarrow y \in W_x$

□

EXERCISE: there is a total computable function $K : \mathbb{N} \to \mathbb{N}$

$W_{K(x)} = \mathbb{P}$ (natural numbers)

$E_{K(x)} = \{ n \in \mathbb{N} \mid n \geq x \}$

define $f(x,y) = \begin{cases} x + y/2 & \text{if } y \in \mathbb{P} \\ \uparrow & \text{if } y \notin \mathbb{P} \end{cases}$

$= x + q(t(2,y)) + \underbrace{\mu w.\ rm(2,y)}$

0 when $y \in \mathbb{P}$

$\uparrow$ when $y \notin \mathbb{P}$

computable, hence by smn theorem there is $K : \mathbb{N} \to \mathbb{N}$ s.t. $\forall x, y$

$$\varphi_{K(x)}(y) = f(x,y) = \begin{cases} x + y/2 & \text{if } y \in \mathbb{P} \\ \uparrow & \text{if } y \notin \mathbb{P} \end{cases}$$

K is the desired function

- $W_{K(x)} = \mathbb{P}$

- $E_{K(x)} = \{ n \mid n \geq x \}$ $\qquad \mathbb{P}$ $\qquad =$

$E_{K(x)} = \{ \varphi_{K(x)}(y) \mid y \in W_{K(x)} \}$

$= \{ \varphi_{K(x)}(y) \mid y \in \mathbb{P} \}$

$= \{ \varphi_{K(x)}(2z) \mid z \in \mathbb{N} \}$

$f(x,2z) = x + \frac{2z}{2} = x + z$

$= \{ x + z \mid z \in \mathbb{N} \}$

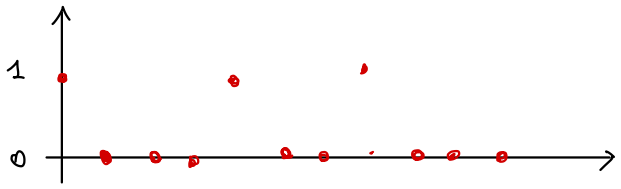
$= \{ n \mid n \geq x \}$

EXERCISE : Are there $f, g : \mathbb{N} \to \mathbb{N}$ s.t. f computable, g not computable

and $f \circ g$ computable ? YES

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

not computable



$f(x) = 0 \quad \forall x$

computable



then $f \circ g(x) = f(g(x)) = 0 \quad \forall x$ computable

* Are there $f, g : \mathbb{N} \to \mathbb{N}$ s.t. f not computable, g not computable

and $f \circ g$ computable ? YES

g as before $\quad g(x) = \begin{cases} 1 & \text{Halt}(x) \\ 0 & \text{otherwise} \end{cases}$

not computable

$$f(x) = \begin{cases} 0 & \text{if } x \leq 1 \\ \varphi_x(x)+1 & \text{if } x > 1 \text{ and } \varphi_x(x) \downarrow \\ 0 & \text{if } x > 1 \text{ and } \varphi_x(x) \uparrow \end{cases}$$

$$f(x) = \begin{cases} 0 & \text{if } x \leq 1 \\ \text{Halt}(x) & \text{otherwise} \end{cases}$$

non computable

$\forall x > 1 \quad \varphi_x \neq f$

$\forall y \in \mathbb{N} \quad \exists x > 1$ s.t. $\varphi_y = \varphi_x \neq f \quad \rightsquigarrow$ f not computable

EXERCISE : Can every function f which is computable can be obtained as $f = g \circ h$ with g, h not computable ?

EXERCISE: Prove that $pow_2 : \mathbb{N} \to \mathbb{N}$

$pow_2(x) = 2^x$ is in PR

by using only the definition of PR

(PR least class of functions
- including the basic functions ↙ zero, projection, successor
- closed under composition and primitive recursion)

$x+y$ $\begin{cases} x+0 = x \\ x+(y+1) = (x+y)+1 \end{cases}$

$x*y$ $\begin{cases} x*0 = 0 \\ x*(y+1) = (x*y)+x \end{cases}$

x^y $\begin{cases} x^0 = 1 \\ x^{y+1} = x^y * x \end{cases}$

finally

$pow_2(x) = 2^x = succ(succ(0))^x$

alternatively

$\begin{cases} pow_2(0) = 2^0 = 1 = succ(0) \\ pow_2(y+1) = 2^{y+1} = 2^y * 2 = 2^y + 2^y \end{cases}$

$x+y$ $\begin{cases} x+0 = x \\ x+(y+1) = (x+y)+1 \end{cases}$

alternatively

$\begin{cases} pow_2(0) = 2^0 = 1 = succ(0) \\ pow_2(y+1) = 2^{y+1} = 2^y * 2 = 2^y + 2^y = twice(2^y) \end{cases}$

$\begin{cases} twice(0) = 0 \\ twice(y+1) = 2(y+1) = 2y+2 = succ(succ(twice(y))) \end{cases}$

EXERCISE: Prove that fibonacci is primitive recursive

$$\begin{cases} f(0) = 1 \\ f(1) = 1 \\ f(m+2) = f(m) + f(m+1) \end{cases}$$

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

$g(m) = \pi(f(m), f(m+1))$ π encoding of pairs

$$\begin{cases} g(0) = \pi(f(0), f(1)) = \pi(1,1) \\ g(m+1) = \pi(f(m+1), f(m+2)) \end{cases}$$

$f(m+1) = \pi_2(g(m))$

$f(m) + f(m+1) = \pi_1(g(m)) + \pi_2(g(m))$

$$= \pi(\pi_2(g(m)), \pi_1(g(m)) + \pi_2(g(m)))$$

primitive recursive since $\pi, \pi_1, \pi_2 \in PR$

now

$$f(m) = \pi_1(g(m))$$

hence $f \in PR$