Let e be code for this TM.

Then: $f(k_1, .., k_n) = f'(\langle k_1, .., k_n \rangle)$
$= U(\mu c\, T(e, \langle k_1, .., k_n \rangle, c))$

where "=" means $\doteq$ when such c exists (which does if f terminates — M computes f!) and undef when $f \uparrow$ (which happens exactly when there is no such c). ✓

↳ Again big idea is: TMs + their computations can be encoded in natural way.

- checking whether a given c encodes a computation of a machine M (encoded by e) on an input n is a p.r. process
- and in fact a uniform p.r. process i.e. there is a single p.r. relation which performs this check, namely T
- also p.r. to extract last output of such a computation (this is U)
- so: given rec. f can find TM computing f w/ code e and compute $f(\vec{x})$ by searching for a halting computation of TM on input $\vec{x}$ (this only non-pr part)
- Fact that all TM computations coded into U, T leads to universal functions...

Universal Functions

- Sps $\mathcal{F}$ is a class of (possibly partial) functions of various arities, from $\mathbb{N} \to \mathbb{N}$.
  e.g. class of all p.r. functions
  class of all recursive functions
- We let $\mathcal{F}_n$ to denote the n-ary functions in $\mathcal{F}$.

- given $(n+1)$-ary $\varphi : \mathbb{N}^{n+1} \to \mathbb{N}$ and $e \in \mathbb{N}$ (index), we write $\varphi_e(\vec{x})$ for $\varphi(e, \vec{x})$ (so $\varphi_e : \mathbb{N}^n \to \mathbb{N}$)

Defn $\varphi : \mathbb{N}^{n+1} \to \mathbb{N}$ is **universal for** $\mathcal{F}_n$ if $\mathcal{F}_n = \{\varphi_e \mid e \in \mathbb{N}\}$

↖ functions can repeat in this list

- abstractly: $\mathcal{F}_n$ has a universal function iff $\mathcal{F}_n$ is cbl
  ↳ since if $\mathcal{F}_n = \{F_0, F_1, F_2, \ldots\}$
  can let $\varphi(e, \vec{x}) = F_e(\vec{x})$
  ↳ and if φ universal for $\mathcal{F}_n$ then $\mathcal{F}_n = \{\varphi_0, \varphi_1, \ldots\}$ is cbl.

- however: we'll often be interested in universal functions that belong to $\mathcal{F}$. (e.g. a universal function for the n-ary recursive functions that is also recursive)

Def'n $\mathcal{F}$ has **universal functions** or the **enumeration property** if $\forall n$, there is $\varphi^n \in \mathcal{F}_{n+1}$ which is universal for $\mathcal{F}_n$

- **note**: if $\mathcal{F}$ is closed under substitution by p.r. functions, then this is equiv to just a universal $\varphi^1 \in \mathcal{F}_2$ for $\mathcal{F}_1$:

  why every $f \in \mathcal{F}_n$ can be coded as a unary $f' \in \mathcal{F}_1$ by defining $f'(x) = f((x)_0, \ldots, (x)_{n-1})$

  then can get a universal φ^n by defining $\varphi^n(e, \vec{x}) = \varphi^1(e, \langle \vec{x} \rangle)$

Prop'n if $\mathcal{F}$ is all p.r. functions, or if $\mathcal{F}$ is all **total** recursive functions, then $\mathcal{F}$ does **not** have universal functions

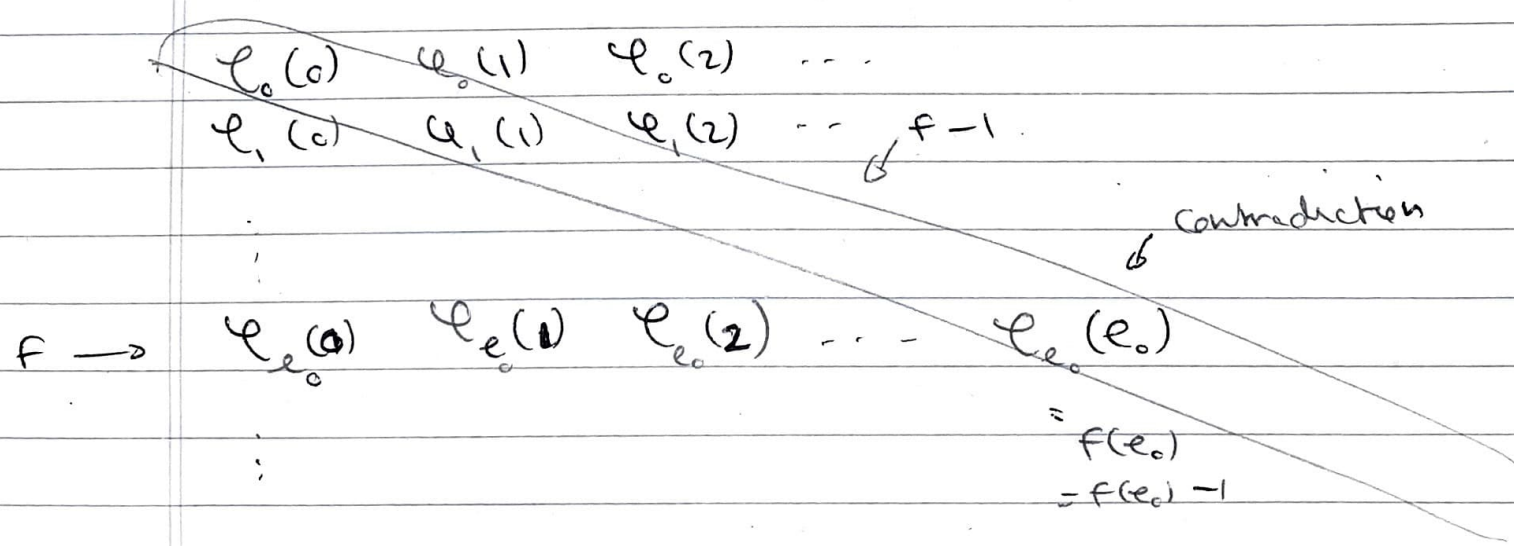
Pf: - by note above, suffices to show there is no universal $\varphi : \mathbb{N}^2 \to \mathbb{N}$ for $\mathcal{F}_1$

- if there were define $f : \mathbb{N} \to \mathbb{N}$ by $f(n) = \varphi(n,n) + 1$ ("diagonalize")
- f is pr if φ pr
  or total recursive if φ t.r., i.e. $f \in \mathcal{F}_1$
- then by universality of φ, there is e_0 s.t. $f = \varphi_{e_0}$, i.e. $f(n) = \varphi(e_0, n)$ $\forall n$

- but then $f(e_0) = \varphi(e_0, e_0) + 1$ (def'n of f)
  but also $= \varphi(e_0, e_0)$ (by above)
  contradiction ✓
  (using totality by assuming $f(e_0)$ is defined) ✓

- Summary: a 2-ary universal p.r. function φ says "we can list all unary p.r. functions in a p.r. way"
- but then diagonal function $g(n) = \varphi(n,n)$ is p.r. too, and so is $f(n) = \varphi(n,n)+1$
- hence f must appear in list... but it can't!

Pic:



Theorem (Kleene enumeration theorem) The class $\mathcal{F}$ of all partial recursive functions has universal functions.

Pf: - suffices to find universal $\varphi' = \varphi$.
- Use normal form theorem
  define $\varphi(e,x) = U(\mu c\, T(e,x,c))$
- as e varies over $\mathbb{N}$, hits code for every TM, hence every recursive unary f is φ_e for some e ✓

(think abt: why doesn't diagonalization work this time?)

Theorem there is a total recursive function which is universal for the p.r. functions.

Pf (Sketch): we first assign codes to p.r. functions as follows:

function	code
$x \mapsto c$	$\langle 0,1\rangle$
$x \mapsto x+1$	$\langle 1,1\rangle$
$(x_1,\dots,x_n) \mapsto x_i$	$\langle 2,n,i\rangle$
$f(\vec{x}) = g(h_1(\vec{x}),\dots,h_m(\vec{x}))$	$\langle 3,n,e_0,e_1,\dots,e_m\rangle$ where e_0 = code for g, e_i = code for h_i

$f(0) = w_0$

$f(i+1) = h(f(i), i)$ $\quad \langle 4, 1, e_0, w_0 \rangle$ w/ e_0 code for h (prim recursion w/ no parameters)

$f(0, \vec{y}) = g(\vec{y})$

$f(i+1, \vec{y}) = h(f(i, \vec{y}), i, \vec{y})$ $\quad \langle 5, n, e_1, e_2 \rangle$ e_1 code of g, e_2 code of h (parameters)

(so code for f looks like $\langle$ how f built, codes for functions needed to build $f \rangle$)

universal for p.r.

- we'll define $\varphi(e, x) = \varphi_e(x)$ which is total recursive s.t. if $f(x_1, .., x_n)$ is p.r. and e any code for f (could be more than one!) then
$$f(\vec{x}) = \varphi(e, \langle \vec{x} \rangle)$$
- conversely φ_e will be p.r. $\forall e \in \mathbb{N}$.

- explicitly, we define

$\varphi(e, x) = f((x)_0, .., (x)_{n-1})$ if e codes a p.r. function f of arity n, and x codes an n-tuple

$= 0$ o/w

- then φ is total by defn
- and φ_e is p.r. for every $e \in \mathbb{N}$:
  - ↳ if e not a code for a pr function $\varphi_e \equiv 0$
  - ↳ if e is a code, checking whether x codes tuple of correct arity is pr. hence can define φ_e in piecewise pr. way

- φ is intuitively recursive: to compute $\varphi(e,x)$
  - check if e codes a p.r. f
  - check if x codes a tuple $\vec{x}$ of correct arity
  - if both yes, compute $f(\vec{x})$!
    - ↳ all info needed contained in e, which functions needed to build f, how f built, which functions needed to build these, etc.. down to base functions
      - ↳ i.e. can extract "computation tree" for $f(\vec{x})$ from e

- to formalize, use arg similar to showing Ackermann recursive:

  $\varphi(e,x)$ = "search for least s coding a computation tree of $f(x)$ and return output of computation"

  need to code in some way, but

once this done, checking if a given s codes tree for $f(x)$ w p.r. (in e, s)
$= \varphi_e(x)$

only non-p.r. part w search for s ✓

Prop'n the graph of φ w not p.r.

Pf: Sps $R(e,x,y)$ iff $\varphi(e,x) = y$ w p.r.
- Define $S(e)$ iff $\varphi(e,e) = 1$
- also p.r. (obtained from R by pr sbn)
- i.e. χ_S is p.r.
- but then for some e_0, $\chi_S = \varphi_{e_0}$
  i.e. $S(e)$ iff $\chi_S(e) = 0$ iff $\varphi(e_0, e) = 0$
  (remember: 0 means true)
- but then $S(e_0)$ iff $\varphi(e_0, e_0) = 1$ (defn)
  iff $\varphi(e_0, e_0) = 0$ (by above)
  contradiction ✓

(again need φ w total)

Corollary there w a subset $A \subseteq \mathbb{N}$ which w recursive but not p.r.
(i.e. χ_A rec but not p.r.)

Pf: let $A = \{\langle e,x,y\rangle : \varphi(e,x) = y\}$
$=$ coded graph of φ

- then χ_A recursive (why?)
- but not p.r. (why?)

Recursively enumerable relations

- going back for a moment to general alphabet A.

Defn an n-ary relation $P \subseteq (A^*)^n$ is **recursively enumerable** (r.e.) if either $P = \emptyset$ or there is a total recursive $f: \mathbb{N} \to (A^*)^n$ that enumerates P (i.e. $P = \{f(0), f(1), \ldots\}$)

- **rek**: $f: \mathbb{N} \to (A^*)^n$ being recursive means each coordinate function f_i is recursive, where

$$f(k) = (f_1(k), \ldots, f_n(k))$$

- **rek** saying $f_i: \mathbb{N} \to A^*$ is recursive means f_i is restriction of a recursive $f_i': A^* \to A^*$ to $\mathbb{N}$.

... and right back to work on $\mathbb{N}$ anyway:

Theorem For $R \subseteq \mathbb{N}^n$, the following are equivalent:

(i) R is r.e.

(i)' There is a partial recursive $f: \mathbb{N} \to \mathbb{N}^n$ with $R = \text{ran}(f)$

(i)" there is a primitive recursive $f: \mathbb{N} \to \mathbb{N}^n$ with $R = \text{ran}(f)$; or $R = \emptyset$

(ii) there is a recursive $P \subseteq \mathbb{N}^{n+1}$ s.t. $R(\vec{x})$ iff $\exists y\, P(y, \vec{x})$

("R is projection of P")

(ii)' there is a p.r. P satisfying the above.

(iii) there is a partial recursive $f: \mathbb{N}^n \to \mathbb{N}$ with $R = \text{dom}(f)$.