(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.

$$R(\vec{x}) \text{ iff } \exists y\, P(y, \vec{x})$$

("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 $R = \text{dom}(f)$.

<u>Pf</u>: (i) ⟹ (ii): Sps R r.e.

- if $R = \emptyset$, (ii) immediate (why?)
- so sps $R = \text{ran}(f)$ for some tot. rec. $f: \mathbb{N} \to \mathbb{N}^n$ (write: $f = (f_1, f_2, \ldots, f_n)$)
- Define $P(y, \vec{x})$ iff

$$x_1 = f_1(y) \wedge x_2 = f_2(y) \wedge \cdots \wedge x_n = f_n(y) \quad \text{"}f(y) = \vec{x}\text{"}$$

then P is rec. (= is p.r., rec rel'ns closed under total rec. sub'n)

and $R(\vec{x})$ iff $\exists y\, P(y, \vec{x})$ ✓

(ii) ⟹ (iii): Given rec. P s.t. $R(\vec{x})$ iff $\exists y\, P(y, \vec{x})$, define

$$f(\vec{x}) = \mu y\, P(y, \vec{x})$$

- then f is rec. and $\vec{x} \in \text{dom}(f)$ iff $R(\vec{x})$ ✓

(iii) ⟹ (i''): Fix $f: \mathbb{N}^n \to \mathbb{N}$ rec. s.t. $R = \text{dom}(f)$; assume $R \neq \emptyset$, o/w done (why?)

- for some $e \in \mathbb{N}$ (code for f),
  $$f(\vec{x}) = U(\mu c\, T(e, \langle \vec{x} \rangle, c))$$
- notice: have $R(\vec{x})$ iff $\exists y\, T(e, \langle \vec{x} \rangle, y)$
  (iff "there is a code for a halting ~~comp~~ comp'n of $f(\vec{x})$")
- define $P(k,y)$ iff $T(e,k,y)$ (think $k = \langle \vec{x} \rangle$)
- since $R \neq \emptyset$, fix $\vec{z} \in R$ and define
  $$g(k) = \vec{z} \quad \text{if } \neg P(\langle (k)_1, \ldots, (k)_n \rangle, (k)_0)$$
  (if k not code for some $(\vec{x}, y) \in P$ return $\vec{z}$)
  $$= ((k)_1, \ldots, (k)_n) \quad \text{o/w}$$

really $g = (g_1, \ldots, g_n)$ and we define each g_i

- then g is p.r. ~~xxxxxxxxxxxx~~
  P is p.r., since T is.
- g says: "if k codes a halting comp'n for ~~xxx~~ $f(\vec{x})$, return $\vec{x}$, o/w return constant"
- so $\mathrm{ran}(g) = \mathrm{dom}(f) = R$ ✓

↳ we've shown (i) ⇒ (ii) ⇒ (iii) ⇒ ~~xxx~~ (i")
- since clearly (i") ⇒ (i), all three equiv
- also i" ⇒ i' and ii' ⇒ ii clear
- remains to show (i') ⇒ (ii')

(i') ⇒ (ii'): fix $f = (f_1, \ldots, f_n)$ partial rec. s.t. $R = \mathrm{ran}(f)$

– then $R(\vec{x})$ iff $\exists y\ f(y) = \vec{x}$

iff $\exists y [f_1(y) = x_1 \wedge \cdots \wedge f_n(y) = x_n]$

– each f_i (partial) rec., so there is code e_i s.t. $f_i(y) = U(\mu c T(e_i, \langle y \rangle, c))$

– so in above, replace each "$f_i(y) = x_i$" w/ "$U(\mu c T(e_i, \langle y \rangle, c)) = x_i$"

– rewrite as: $R(\vec{x})$ iff

$\exists y\ \forall i \leq n\ \exists c_i [T(e_i, \langle y \rangle, c_i) \wedge \forall d < c_i\ \neg T(e_i, \langle y \rangle, d) \wedge U(c_i) = x_i]$

really writing as a big conjunction

("for every i $\exists$ a code for a halting comp'n...")

equiv: $\exists y \exists s\ \forall i \leq n\ [T(e_i, \langle y \rangle, (s)_i) \wedge \forall d < (s)_i\ \neg T(e_i, \langle y \rangle, d) \wedge U((s)_i) = x_i]$

("$\exists$ a code $s = \langle c_1, \ldots, c_n \rangle$ for a tuple of halting comp'ns s.t. ...") (pulled out a quantifier using codes!)

equiv: $\exists z\ \forall i \leq n$ [... replace y w/ $(z)_0$, s w/ $(z)_1$]

which has form $\exists z\ \underbrace{P(z, \vec{x})}_{\text{p.r.}}$ ✓

<u>Corollary</u> every recursive relation is r.e.

<u>Pf</u> – if $R(\vec{x})$ recursive define $P(\vec{x}, y)$ iff $R(\vec{x}) \wedge y = y$ (also recursive)

– then $R(\vec{x})$ iff $\exists y\ P(\vec{x}, y)$ ✓

so R r.e. by (ii)

Comparing recursive and r.e.

- intuitively:
  R recursive if there is (total) algorithm (namely χ_R) that determines if $\vec{x} \in R$
  R r.e. if there is partial algorithm that determines (sometimes) if $\vec{x} \in R$
  namely: if $R = \mathrm{ran}(f)$, keep plugging in numbers to f, see if $\vec{x}$ ever outputs...

- intuitively r.e. strictly harder than rec.
- we prove! (takes some work)

Theorem $R \subseteq \mathbb{N}^n$ is recursive iff both R and $\neg R$ (complement) are r.e.

pf ($\Rightarrow$) clear, you try
($\Leftarrow$) - Sps P_1, P_2 rec. rel'ns s.t.
$R(\vec{x})$ iff $\exists y\, P_1(y, \vec{x})$ and $\neg R(\vec{x})$ iff $\exists y\, P_2(y, \vec{x})$
- define
$f(\vec{x}) = \mu y\,[P_1(y, \vec{x}) \vee P_2(y, \vec{x})]$
- then $f(\vec{x})$ recursive and total
- hence $R(\vec{x})$ iff $P_1(f(\vec{x}), \vec{x})$
$\Rightarrow R(\vec{x})$ is recursive (why? use χ_{P_1} to write χ_R) ✓

Theorem $f: \mathbb{N}^n \to \mathbb{N}$ is partial recursive iff $\text{graph}(f) \subseteq \mathbb{N}^{n+1}$ is r.e. (i.e. rel'n $f(\vec{x}) = y$ is r.e.)

Pf: ($\Rightarrow$) - for some e we have $f(\vec{x}) = U(\mu c\, T(e, \langle\vec{x}\rangle, c))$, so:

- $f(\vec{x}) = y$ iff $\exists c \big(T(e, \langle\vec{x}\rangle, c) \wedge \forall d < c\ \neg T(e, \langle\vec{x}\rangle, d) \wedge U(c) = y\big)$

"$\exists c$ (p.r rel'n)" which is r.e.

($\Leftarrow$) - Now spse $f(\vec{x}) = y$ iff $\exists t\, R(\vec{x}, y, t)$ for some recursive R

- then $f(\vec{x}) = (\mu s\, R(\vec{x}, (s)_0, (s)_1))_0$ which is recursive. ✓

Theorem (Closure rules for r.e. rel'ns)

1.) if R_1, R_2 n-ary r.e., so are $R_1 \wedge R_2$ and $R_1 \vee R_2$

2) if $R(m, \vec{x})$ r.e. so are:

$Q(n, \vec{x})$ iff $\forall m \leq n\ R(m, \vec{x})$ — bdd $\forall$

$S(\vec{x})$ iff $\exists m\, R(m, \vec{x})$ — unbdd $\exists$

3.) if $R(x_1, .., x_n)$ r.e. and $f_1, .., f_n$ (partial) recursive, then following rel'n r.e.:

$Q(\vec{y})$ iff $f_i(\vec{y})\downarrow$ for all $i \leq n$ $\wedge R(f_1(\vec{y}), ..., f_n(\vec{y}))$

Pf. (1.) Fix P_1 s.t. $R_1(\vec{x})$ iff $\exists y P_1(y, \vec{x})$ likewise P_2 for R_2

- then $R_1(\vec{x}) \wedge R_2(\vec{x})$ iff
  $\exists y P_1(y, \vec{x}) \wedge \exists z P_2(z, \vec{x})$ iff
  $\exists s \, (P_1((s)_0, \vec{x}) \wedge P_2((s)_1, \vec{x}))$
  "$\exists s$ (recursive)"
  which gives $R_1 \wedge R_2$ is r.e.
- for $\vee$: you try

(2.) Fix recursive P s.t. $R(m, \vec{x})$ iff $\exists t \, P(t, m, \vec{x})$

- then $Q(n, \vec{x})$ iff $\forall m \le n \, \exists t \, P(t, m, \vec{x})$
  iff $\exists s \, \forall m \le n \, P((s)_m, m, \vec{x}))$
  "$\exists s$ (recursive)" $\Rightarrow$ hence r.e.
- for $S(\vec{x})$: you try (code $\exists m \exists t$ as $\exists s = \langle m, t \rangle$)

(3) $Q(\vec{y})$ iff $\exists x \, (f_1(\vec{y}) = (x)_1 \wedge f_2(\vec{y}) = (x)_2 \wedge \dots$
$\wedge f_n(\vec{y}) = (x)_n \wedge R((x)_1, \dots, (x)_n)]$
"$\exists x$ (r.e.)" (by ①)
$\Rightarrow$ r.e (by ②) ✓

<u>Ex</u> Sps $g(\vec{x})$ is recursive and $R(\vec{x})$ is r.e.
define:

$$f(\vec{x}) = g(\vec{x}) \quad \text{if } R(\vec{x})$$
$$= \text{undef} \quad \text{if } \neg R(\vec{x})$$

- then f is recursive:

$$f(\vec{x}) = y \quad \text{iff} \quad \underbrace{\underbrace{R(\vec{x})}_{\text{r.e.}} \wedge \underbrace{g(\vec{x}) = y}_{\text{r.e.}}}_{\text{r.e.}}$$

$\Rightarrow$ "$f(\vec{x}) = y$" r.e $\Rightarrow$ f recursi