Pf: – suffices to prove every recursive function f definable: r.c. sets are domains of recursive f, so defined by $\exists y\,(f(\vec{x}) = y)$

– we will induct on the construction of f:

– show basic functions are definable

– then show definable functions are closed under composition, minimalization, and prim. recursion.

Basic f: projections π^n_i, constant 0-maps c_n, and successor all definable respectively by:

$\varphi(x_1, \ldots, x_i, \ldots, x_n, y) := y = x_i$

$\varphi(x_1, \ldots, x_n, y) := y = 0$

$\varphi(x, y) := y = S(x)$

Compositions: Sps $f(x_1, \ldots, x_k) = h(g_1(\vec{x}), \ldots, g_m(\vec{x}))$ and we have

$\chi(y_1, \ldots, y_m, z)$ defining (graph of) h

$\tau_i(x_1, \ldots, x_k, y)$ defining g_i $\forall i \leq m$

Then $\varphi(\vec{x}, z) := \exists y_1 \ldots \exists y_m (\tau_1(\vec{x}, y_1) \wedge \tau_2(\vec{x}, y_2) \wedge \ldots \wedge \tau_m(\vec{x}, y_m) \wedge \chi(y_1, \ldots, y_m, z))$

defines f

"$g_1(\vec{x}) = y_1 \wedge \ldots \wedge g_m(\vec{x}) = y_m \wedge h(y_1, \ldots, y_m) = z$"

Minimalization: Sps $f(\vec{x}) = \mu n(g(\vec{x},n)=0)$ and sps $\chi(\vec{x},n,y)$ defines "$g(\vec{x},n)=y$"
then f is defined by

$$\varphi(x,z) := \exists y\,(\chi(\vec{x},y,0) \wedge \forall y'(y'<y \Rightarrow \exists z'(z'\neq 0 \wedge \chi(\vec{x},y',z'))) \wedge z=y)$$

Primitive recursion: the tricky one!

- Sps f defined by the recursion:

$$f(0,\vec{x}) = g(\vec{x})$$
$$f(n+1,\vec{x}) = h(f(n,\vec{x}),n,\vec{x})$$

and we have fmlas
$\chi(\vec{x},y)$ defining "$g(\vec{x})=y$"
$\tau(a,b,\vec{x},c)$ defining "$h(a,b,\vec{x})=c$".

then f is defined by the "formula"

$$\varphi(n,\vec{x},z) := (n=0 \wedge \chi(\vec{x},z)) \vee (n\neq 0 \wedge \exists \ell_0 \exists \ell_1 \ldots \exists \ell_{n-1}[\chi(\vec{x},\ell_0) \wedge \tau(\ell_0,0,\vec{x},\ell_1) \wedge \tau(\ell_1,1,\vec{x},\ell_2) \wedge \ldots \wedge \tau(\ell_{n-1},n-1,\vec{x},z)])$$

↳ this is not a fmla since # of $\exists \ell_i$ quantifiers and $\tau(\ell_i,i,\vec{x},\ell_{i+1})$ clauses depend on n

↳ need to do some coding... Gödel saw how.

- Want: "definable coding function", i.e. a definable $\gamma(\ell, i)$ s.t. for any n and sequence $\ell_0, \dots, \ell_{n-1}$ there is ℓ s.t. $\gamma(\ell, i) = \ell_i$ $\forall i < n$
- could use the p.r. function $(\ell)_i$ if we knew definable in $\mathcal{N}$ - we don't (yet)
- but instead we use: $\beta(a, b, i)$ s.t. $\forall n$ and all $\ell_0, \dots, \ell_{n-1}$ $\exists a, b$ s.t. $\forall i < n$, $\beta(a, b, i) = \ell_i$
  ↪ β will store info in remainders instead of prime powers

Def'n Gödel's β-function is the map
$$\beta(a, b, i) := \text{rem}(a, 1 + (i+1)b)$$

where $\text{rem}(x, y) :=$ remainder of division $\frac{x}{y}$

Observe: β is definable in $\mathcal{N}$, e.g. by: $\varphi(a, b, i, z) :=$
$$z < (\underline{1} + (i+1) \cdot b) \wedge \exists k ((\underline{1} + (i+1)b)k + z = a)$$
where $\underline{1} = s(c)$, $i+1 = s(i)$ etc.

(ez'er to see: $\text{rem}(x, y) = z$ definable by: $z < y \wedge \exists k (y \cdot k + z = x)$)

Lemma If $n > 0$ then for every sequence $\ell_0, \dots, \ell_{n-1}$ there are a, b s.t.
$$\beta(a, b, i) = \ell_i \quad \forall i < n$$

Before proof, observe that we can now use β to get a def'n for f above: replace the clause $\exists l_0 \dots \exists l_{n-1}\,[\dots]$ by:

"$\exists a \exists b\ \forall i<n\ \exists l_i \exists l_{i+1}$
$[\beta(a,b,i) = l_i \wedge \beta(a,b,i+1) = l_{i+1}$
$\wedge g(\vec{x}) = l_0 \wedge h(l_i, i, \vec{x}) = l_{i+1}$
$\wedge l_n = y]$"

informally written — can you write formally? (exercise)

- So we need only prove Lemma to be done. Will need:

(CRT)

Chinese remainder theorem:
if $b_0, \dots, b_{n-1}$ are pairwise relatively prime and $l_0, \dots, l_{n-1}$ is a sequence s.t. $l_i < b_i$ $\forall i<n$ then there is $a \in \mathbb{N}$ s.t. $l_i = \text{rem}(a, b_i)$ $\forall i<n$
(i.e. $a \in \mathbb{N}$ solves the system
$a \equiv l_i \pmod{b_i}$)

Pf of Lemma from CRT:

- Let $k > \max\{n, l_0, \dots, l_{n-1}\}$
- Let $b = k!$
- For $i<n$ let $b_i = 1 + (i+1)\,b$
$b_0 = 1+b$, $b_1 = 1+2b$, ...

- Clearly $l_i < b_i$ $\forall i < n$ (by choice of k)
- we claim the b_i are pairwise coprime
  - if for $i < j$ b_i, b_j shared a prime factor p then $p \mid b_j - b_i$
    i.e. $p \mid b(j-i)$
    i.e. $p \mid k!(j-i)$ $\quad k > n$
  - then since $j - i < k$ we have $j - i \mid k!$
  - hence $p \mid k!$ (since $p \mid k!$ or $p \mid j-i$)
    i.e. $p \mid b$
  - but then since $p \mid b_i = 1 + (i+1)b$ follows $p \mid 1$, contr.

- hence b_i satisfy hypotheses of CRT
- let $a \in \mathbb{N}$ be as guaranteed by CRT
  so $\text{rem}(a, b_i) = l_i \quad \forall i < n$
  i.e. $\text{rem}(a, 1 + (i+1)b) = l_i \quad \forall i < n$
  i.e. $\beta(a, b, i) = l_i \quad \forall i < n$ ✓

<u>Pf of CRT</u>: pigeonhole argument!

- For a given $d \in \mathbb{N}$, define $r(d) = (d_0, \ldots, d_{n-1})$ where $d_i = \text{rem}(d, b_i)$
- <u>notice</u>: at most $b_0 \cdot b_1 \cdots b_{n-1} := t$ possible $r(d)$'s
- we claim r is 1-1 for $d \in \{0, 1, \ldots, t-1\}$
- if so: we are done, since then $r(d) = (l_0, \ldots, l_{n-1})$ for some $d \leq t-1$

- if not there are $e < d < t$ s.t. $r(e) = r(d)$
- so $e_i = d_i$ $\forall i < n$
- so $b_i \mid d - e$ $\forall i < n$
- since b_i coprime, this implies $b_0 b_1 \cdots b_{n-1} \mid d - e$
- but $b_0 \cdots b_{n-1} > d - e$, contr. ✓

Diophantine equations + Hilbert's 10th

- We consider two (related?) problems

Problem 1 We showed $A \subseteq \mathbb{N}^k$ is definable in $\mathcal{N}$ iff A arithmetical (i.e. $A \in \bigcup_n \Sigma^0_n$)

- But what is precise relation between definable (i.e. $\exists_n / \forall_n$) and arithmetical (i.e. Σ_n / Π_n) hierarchies?

- Definable: QF (quantifier free), $\exists_1, \forall_1, \exists_2, \forall_2, \ldots$
  Arithmetical: $\Delta^0_1, \Sigma^0_1, \Pi^0_1, \Delta^0_2, \Sigma^0_2, \Pi^0_2, \ldots$

- We observed: q.f. φ's define recursive A's, so $QF \subseteq \Delta^0_1$

- Can show: containment is strict

- Sketch of why: if φ q.f. $\exists$ a poly time algorithm to determine $\varphi(\vec{a})$ or $\neg\varphi(\vec{a})$: just need to check (essentially) if $\vec{a}$ is or is not the root of a polynomial some finite # of times.
- not all (recursive) characteristic functions run in poly time...

- in any case: since $QF \subseteq \Delta_1^0$, follows $\exists_1 \subseteq \Sigma_1^0$, $\forall_1 \subseteq \Pi_1^0$, ... etc.

Q: are these containments strict too? In particular, do we have $\exists_1 = \Sigma_1^0$?

Problem 2:

Def'n a Diophantine equation is an equation of the form

$$F(x_1, \ldots, x_k) = 0$$

where $F \in \mathbb{Z}[x_1, \ldots, x_k]$ is a (multivariate) polynomial w/ coeffs in $\mathbb{Z}$.

Note: can always rewrite such an eq'n as $G(\vec{x}) = H(\vec{x})$ where G, H have coeffs in $\mathbb{N}$.

↪ often interested in finding integer solutions to such eq'ns

Ex $x^2+y^2-z^2=0$ has inf. many integer solutions

$x^3+y^3-z^3=0$ has no (nonzero) integer sol'ns

(Pell's eq'n) $x^2-ky^2-1=0$ has ∞-many integer sol'ns iff k is not a perfect square.

Q: (Hilbert's 10th): is there an algorithm that decides whether a given Diophantine eq'n has integer solutions?

→ ie recursive

First we observe: there is such an algorithm iff there is an algorithm deciding whether a given eq'n has non-negative integer sol'ns.

Why: - Sps $\exists$ algorithm deciding if sol'ns in $\mathbb{Z}$ and $F(x_1,\dots,x_k)=0$ is Diophantine.

- Consider $F'(p_1,q_1,r_1,s_1,\dots,p_k,q_k,r_k,s_k)$
$:= F(p_1^2+q_1^2+r_1^2+s_1^2, \dots, p_k^2+q_k^2+r_k^2+s_k^2)$
- By Lagrange, every non-negative integer is the sum of four (integer) squares
- Follows: F has sol'ns in $\mathbb{N}$ iff F' has sol'ns in $\mathbb{Z}$ (which we can decide)