Robert M Solovay   M225 Metamathematics II   Tanan Aziz
23 Jan '92
Wednesday

Thy ($\langle \mathbb{R}; +, \cdot, 0, 1, -1, < \rangle$) : admits quantifier Elimination

Thy ($\langle \omega; +, \cdot, 0, 1, < \rangle$)

Results from last term:

T. Theory

T satisfies the isomorphism conditions

if whenever $\mathfrak{a}, \mathfrak{a}'$ are models of T

$\mathfrak{B}$ and $\mathfrak{B}'$ are substructures of $\mathfrak{a}, \mathfrak{a}'$

($\mathfrak{B} \subseteq \mathfrak{a}$, $\mathfrak{B}' \subseteq \mathfrak{a}'$) and $\mathfrak{B} \cong \mathfrak{B}'$ via φ

then φ can be prolonged to an isomorphism

$\psi : \mathfrak{C} \cong \mathfrak{C}'$, with $\mathfrak{B} \subseteq \mathfrak{C} \subseteq \mathfrak{a}$

$\mathfrak{B}' \subseteq \mathfrak{C}' \subseteq \mathfrak{a}'$

and $\mathfrak{c}, \mathfrak{c}'$ are models of T

T satisfies the submodel condition if whenever

$\mathfrak{a}, \mathfrak{B}$ are submodels of T,

$\mathfrak{a} \subseteq \mathfrak{B}$

and Φ is simply existential sentence of $L(\mathfrak{a})$,

then $\mathfrak{a}(\Phi) = \mathfrak{B}(\Phi)$

Thm A If T satisfies isomorphism and submodel conditions it admits elim of quantifiers

Thm B Assume T consistent

T has a constant

T admits elim of quantifiers

Any variable free sentence of T is decidable in T.

Then T is complete

$\mathbb{C} \quad \Leftrightarrow \quad \mathbb{R}$

Fields $\Leftrightarrow$ ordered fields

algebraically closed Field $\Leftrightarrow$ Real Closed Field

Recall FL is theory of fields

$L_{FL} = \{+, \cdot, 0, 1, -1\}$

The theory of OF:

$L_{OF} = \{+, \cdot, 0, 1, -1, <\}$

AX_{OF} : AX_{FL} +

<u>OF1</u> $\neg(x<x)$

<u>OF2</u> $(x<y \wedge y<z) \rightarrow x<z$

<u>OF3</u> $(x<y) \vee (x=y) \vee (y<x)$ /* trichotomy */

<u>OF4</u> $x<y \rightarrow (x+z<y+z)$

<u>OF5</u> $(0<z \wedge x<y) \rightarrow (x \cdot z < y \cdot z)$

Algebra: Van der Waerden Ch IX Real Fields

<u>Thm</u> (Artin-Schreier)

Let F be an ordered field

TFAE

$\boxed{1}$ If $a, b \in F$, $a<b$, $p(x)$ is a polynomial of one var with coefficients in F,

and $p(a)<0$, $p(b)>0$,

then $\exists c$: $a<c<b$ & $p(c)=0$

$\boxed{2}$ every positive element of F has a square root AND every poly of odd degree has a root

Dfn F is real closed if it satisfies [1] (or [2])

Clear (1) $\mathbb{R}$ is real closed

(2) RCF has recursive set of axioms

$\Leftrightarrow$ "There is a theory RCF whose models are precisely the real closed fields."

Goals

a) Show RCF admits elimination of Quantifiers

b) Show RCF is complete

c) Th($\mathbb{R}$) is equiv to RCF

RCF : OF + $\{IV_n \mid n \geq 1\}$

(IV_n is a first order sentence expressing : if p is a poly of degree n

$p(a) < 0 < p(b)$, $a<b \to (\exists c)(a<c<b \wedge p(c)=0)$)

$Ax_{Th(\mathbb{R})} = \{\varphi \mid \varphi$ is a sentence of L_{RCF} & $\langle \mathbb{R}; +, \cdot, 0, 1, -1, < \rangle \models \varphi\}$

(b) follow immediately from (a) and Theorem B.

So if we know (a), RCF complete

Exercise If T is complete & $a \models T$,

T is equiv to Th(a)

Apply : T = RCF

$a = \mathbb{R}$

RMS

Office Hours 717 Evans

MW 4^{10} - 5 o'clock

or by appointment

Goals (1) Th($\mathbb{R}$) is equivalent to RCF

(2) RCF admits elimination of quantifiers

Last time showed ETS

(1) RCF satisfies isomorphism condition.

(2) RCF " submodel condition.

Isomorphism Condition:

Suppose F_1 and F_2 are real closed fields

R_i is a substructure of F_i

Let $\varphi : R_1 \cong R_2$ as ordered rings

To show $\exists\, G_1^* \subseteq F_1$ real closed fields

$G_2^* \subseteq F_2$

$R_1 \subseteq G_1^*$

$R_2 \subseteq G_2^*$

and φ prolongs to an isomorphism $\varphi^{**} : G_1^* \cong G_2^*$

Let G_i be quotient field of R_i computed inside F_i

As usual φ prolongs to $\varphi^* : G_1 \cong G_2$

Easy to see $(a/b) > 0$ iff $(a/b)\, b^2 = ab > 0$ So φ^* is order preserving

Let $G_i^* = \{\alpha \in F_i \mid \alpha \text{ is algebraic over } G_i\}$

Easy to see: G_i^* is real-closed

van der Waerden: The isomorphism $\varphi^* : G_1 \cong G_2$ prolongs to $\varphi^{**} : G_1^* \cong G_2^*$ QED

Let $\alpha \in G_1$

What should $\varphi^{**}(\alpha)$ be?

we should have $P_1(\alpha) = 0$ for some P_1 with coefficients in G_1

Basic non-trivial fact:

(Sturm's algorithm) The number of roots of P_1 in any real closed extension can be computed working solely in G_1.

Let P_2 be obtained from $P_1(x)$ by applying φ^* to coefficients

P_2 has as many roots in G_2^* as P_1 has in G_1^*.

α is the j^{th} root of P_1.

send α to the j^{th} root of P_2.

$= \varphi^{**}(\alpha)$

End of Discussion

Submodel Condition

Let $G \subseteq F$

G, F real closed fields

A simply existential sentence in $LT[G]$

$\exists x \, \Phi(x)$

To see

$F \models A \Leftrightarrow$

$G \models A$

Well $G \models A \rightarrow F \models A$ is trivial

$a \in G$

$G \models \Phi(a)$

then $F \models \Phi(a)$ since Φ has no quantifiers

so $F \models \exists x \, \Phi(x)$

654-307

3+4=7

Reduction:

Suppose $P_1(x), \ldots, P_n(x)$ are polynomials,

$P_1(x) = x$ & no $P_i(x) \equiv 0$.

Let $b \in F$

To find $a \in G \ni$

$P_i(a) = 0$ iff $P_i(b) = 0$

$P_i(a) > 0 \Leftrightarrow P_i(b) > 0$

<u>Case 1</u> b is a root of some P_i

By Sturms algorithm

P_i has same number of roots in G & in F

So $b \in G$

Done: Take $a = b$

Let $R = \{\alpha \mid \alpha$ is a root of some $P_i\}$

Let $R = \{\xi_1, \ldots, \xi_n\}$ $\quad \xi_1 \leq \xi_2 \leq \ldots \leq \xi_n$

<u>Case 2</u> For some i, $\xi_i < b < \xi_{i+1}$

For any j,

P_j has constant sign in ~~$(\xi_i, \ldots)$~~ (ξ_i, ξ_{i+1})

since it has no roots in this interval.

So $a = \left(\frac{\xi_1 + \xi_2}{2}\right)$ this works

ie $P_j(a) > 0 \Leftrightarrow P_j(b) > 0$

$P_j(a) \neq 0$, $P_j(b) \neq 0$

<u>Case 3</u> $b > \xi_n$

Take $a = \xi_n + 1$

this works since

P_i's are of constant sign on (ξ_n, ∞)

K. Solovay M225(b) Recursion Theory Jan 27 '92 Monday

Advertisment for the theorems we are building up to.

Church's Theorem
Gödels 1st incompleteness theorem
Gödels 2nd incompleteness theorem

L - first order language with finitely many non-logical symbols

$\underline{P}$ - $0, S, +, \cdot$

ZFC - $\in$

A is a set of formulae of L
We will be defining precisely notion "A is recursive"
intuition: Can mechanically decide of a formula φ of L whether or not $\varphi \in A$.

$\underline{P}$ - Peano Arithmetic   Model: $\langle \omega; 0, S, +, \cdot \rangle$ $Sx = x+1$

Axioms of PE

(1) $0 \neq Sx$
(2) $Sx = Sy \to x = y$
(3) $x + 0 = x$
(4) $x + Sy = S(x+y)$
(5) $x \cdot 0 = 0$
(6) $x \cdot Sy = (x \cdot y) + x$

plus an infinite collection of induction axioms
φ is a formulae, x a variable
$[\varphi_x[0] \wedge [(\forall x)(\varphi \to \varphi_x[Sx]]] \to \forall x \varphi$
Can not get by with finitely many

Q

Axioms 1) - 6)
8) $x \neq 0 \to (\exists y)(x = Sy)$

$Q \nvdash x + y = y + x$

Sources:
(1) "Undecidable Theories" by Tarski, Mostowski, Robinson
(2) "Schoenfield"
(3) "Introduction to Mathematical Logic" by Schoenfield

Theorem: If T is a thy, $T \vdash 0=1$ and Q is relatively interpretable in T, then $\{\varphi \mid \varphi$ is a thm of $T\}$ is not recursive

Corollary (Church): The set of logically valid sentences of L_T is not recursive

Corollary (Church): The set of logically validities in a single binary predicate is not recursive. (From set theory embedding.)

GB set thy finitely axiomatizable; an example of a nontrivial logical validity is $\wedge GB \rightarrow 4$ Sq Thm

Theorem: (Gödel) 1st Incompleteness Theorem

Far from obvious but

Let T be a theory. Suppose

(0) L_T finite

(1) T is consistent

(2) Q is rel. interpretable in T

(3) T has a recursive set of Axioms

Then T is incomplete.

Theorem: (Gödel) 2nd Incompleteness Theorem

Babbage didn't have vaccuum tubes

If T is rec. axiomatizable, and Q is relatively interpretable in T, then we can express in T, "T is consistent"

Call this sentence of T, con T

If 0) L_T is finite

1) T is consistent

2) AX_T are recursive

3) Q is rel. interpretable in T, then $T \nvdash$ con T

Going to define notion of a recursive function

$\omega = \{0, 1, 2, \dots\}$

$f : \omega^n \to \omega$

"Logic seems to be a long series of tedious verifications whats the point"
Brato me

$V; V^{-\epsilon}; V^{-}; V^{-}; {}^{1/2}; -$

Intuition: A function $f : \omega^n \to \omega$ if there is recursive if there is an explicit algorithm which computes it (requiring no creativity) No limitations of a practical sort on how long algorithm will take.

Examples:

(1) $x+y$

(2) xy

(3) x^y

(4) $i \mapsto p_i$

(5) $f(i) = i^{th}$ digit in decimal expansion of π

(6) $J(n) = 2^{2^{\cdot^{\cdot^{2}}}} \updownarrow n$

Equiv $J(0) = 1$
$J(n+1) = 2^{J(n)}$

(7) $F(J(n))$ (in $\{0, \dots, 9\}$)
To get i^{th} digit of π need at least i steps (not clearly proved)

Recursive Functions:

Recursive Functions will be those computed by register machines

A register machine has an ∞ number of registers $R_\emptyset, R_1, R_2, \ldots, R_n, \ldots$

and a special register I (instruction register)

Machine runs under control of a program $\langle S_\emptyset, \ldots, S_n \rangle$ (S_i are instr)

If $c(I) = j$ then we will next execute instruction j.

If $j > n$ or S_j is HALT m/c will stop

Kinds of Instructions:

(1) ADD1 i: $c(R_i) := c(R_i) + 1$

$c(I) := c(I) + 1$

(2) CPY i j: $c(R_j) := c(R_i)$

$c(I) := c(I) + 1$

$c(R_i) := c(R_i)$

(3) STZ i: $c(R_i) := 0$

$c(I) := c(I) + 1$

(4) TRA i: $c(I) := i$

(5) HALT: Machine Halts

(6) SUB1 i j: If $c(R_i) > 0$ then $\{c(R_i) := c(R_i) - 1;\ c(I) := c(I) + 1;\}$

else if $c(R_i) = 0$ then $c(I) := j$;

A program is a finite sequence of instructions $\langle S_\emptyset, \ldots, S_n \rangle$

Let Π be a program, and let $n \geqslant 1$

We will define f_π^n a function

$\mathrm{Dom}(f_\pi^n) \subseteq \omega^n$

$\mathrm{Ran}(f_\pi^n) \subseteq \omega$

Let $\langle x_1, \dots, x_n \rangle \in \omega^n$

Here is defn of $f_\pi^n(\langle x_1, \dots, x_n \rangle)$

Start register machine with program π,

$c(I) := 0$

$c(R_0) = 0$

$c(R_i) = x_i$ for $1 \le i \le n$

$c(R_j) = 0$ for $j > n$

Let machine ~~halt~~ run until it halts

If machine never halts $f_\pi^n(\langle x_1, \dots, x_n \rangle)$ is not defined.

If it does halt, $c(R_0)$ in halted configuration is $f_\pi^n(\langle x_1, \dots, x_n \rangle)$

Definition: A function is partial recursive if for some n, π $f = f_\pi^n$

Definition: A function is $f: \omega^n \to \omega$ is recursive if

(1) $Dom(f) = \omega^n$

(2) f is partial recursive

Example: $+$ is recursive

X	R1
Y	R2

```
0  CPY 1 0
1  SUB 2 4
2  ADD 1 0
3  TRA 1
4  HALT
```

Some particular functions: (Bringing the cast of characters on to the stage)

$Z: \omega \to \omega$; the identically zero function

$Z(n) = 0$ for all n

$S: \omega \to \omega$; the successor function

$S(x) = x + 1$

If $1 \le i \le n$;

$\Pi_i^n : \omega^n \to \omega$; the projection function

$\Pi_i^n (x_1, \ldots, x_n) = x_i$

Composition:

If $f: \omega^n \to \omega$

and $h_1, \ldots h_n : \omega^m \to \omega$

then we can define a new fcn

$g: \omega^m \to \omega$

by composition from this data

$g(x_1, \ldots, x_m) = f(h_1(\vec{x}), \ldots, h_n(\vec{x}))$

Primitive Recursion:

Let $c \in \omega$

Let $g: \omega^2 \to \omega$

We can define a fcn f by $f(0) = c$

$f(n+1) = g(n, f(n))$

f is obtained from c, g by prim recursion

$0! = \cancel{0}\ 1$

$(n+1)! = (n+1) \cdot n! \quad = (Sn) \cdot n!$

Let $n \geq 2$

Let $f: \omega^{n-1} \to \omega$

$g: \omega^{n+1} \to \omega$

We can define a fcn $h: \omega^n \to \omega$ as follows

We can set $h(x_1, \dots, x_{n-1}, 0) = f(x_1, \dots, x_{n-1})$

$h(x_1, \dots, x_{n-1}, Sk) = g(x_1, \dots, x_{n-1}, k, h(\vec{x}, k))$

Defn: μ-recursion

$\mu z\ (\dots)$
least integer $z\ (\dots)$

Let $f: \omega^{n+1} \to \omega$ is a fcn

and suppose $\forall x_1 \dots \forall x_n \exists y\ f(x_1, \dots, x_n, y) = 0$

then the fcn $h: \omega^n \to \omega$ is obtained from f by μ-recursion

$h(x_1, \dots, x_n) = \mu y [f(x_1, \dots x_n, y) = 0]$

Defn: A collection of functions G is closed under composition if whenever $f: \omega^n \to \omega$ is in G and g is defined from $f, h_1, \dots, h_n$ by composition then $g \in G$

Analogous defns of "closed under prim recursion"
"closed under μ-recursion"

Definition: The μ-recursive functions are the smallest class of functions G such that

(1) $Z, S \in G$

(2) $\Pi^n_i \in G$ if $1 \le i \le n$ for all $n \in \omega$

(3) G is closed under composition, primitive recursion, and μ-recursion

Certainly, there does exist a class (all fns which map $\omega^n \to \omega$ some n) So take intersections of all classes which are closed $\Rightarrow$ get smallest!

Look at operations on to get 'plus' : finite sequence

Definition: The primitive recursive functions are the smallest class of functions G such that

(1) $Z, S \in G$

(2) $\Pi^n_i \in G$ if $1 \le i \le n$

(3) G is closed under composition and prim rec

Remarks: (1) Every primitive recursive fcn is μ recursive

(2) There are μ recursive functions which aren't primitive recursive.

(3) PR $\Rightarrow$ visibly recursive

(4) any practically computable fcn is primitive recursive

asi HW # 1

ms 1> Show any partial recursive fn can be computed by a program containing only ADD1 and SUB1.

ms 2> Suppose $h(n) = 2n$ Then h is recursive

ans 3> Show that if Π is a program that computes h, then it contains at least one ADD1 instr and at least one SUB1 instruction ($\Rightarrow$(1) opt)

ms 4)

eorem: Let $h : \omega^n \to \omega$

TFAE (1) h is recursive

(2) h is μ-recursive

We'll start with (2) $\Rightarrow$ (1)

It suffices to show

(1) Z, S, Π_i^n are recursive

(2) the class of recursive functions is closed under composition, p.r., and μ-recursion.

① STZ 0
HALT
$\Rightarrow Z$ is recursive

② ~~ADD1/1~~
CPY 1 $\emptyset$
ADD1 $\emptyset$
HALT
$\Rightarrow S$ is recursive

③ CPY i $\emptyset$
HALT
$\Rightarrow \Pi_i^n$ is recursive

efinition: A program Π is nice for computing a partial function $f : \omega^n \to \omega$ if

(i) The last instruction of π is a HALT

(ii) Any halt under control of Π is caused by executing this in

(iii) If Π halts on inputs $\langle x_1, \ldots, x_n \rangle$ then upon halting the contents of the registers

$c(R_i) = x_i ; 1 \le i \le n$

$c(R_i) = 0 ; i > n$

roof: (that every partial recursive fn has a nice program)

Let $\Pi_\emptyset$ be a program that computes f. $\Pi_\emptyset = I_\emptyset I_1 \ldots I_{r-1}$

Let Π_1 obtained from $\Pi_\emptyset$ as follows

(a) Add a HALT instruction at end

(b) If some I_j is a HALT; $j < r$ Replace it by a TRA r.

If some I_j is a HALT; $j<r$ Replace it by TRA r

If some I_j TRA s $(s \ge r)$ replace it by SUB1 i r

If some I_j is SUB1 i s $(s \ge r)$ replace

Otherwise o.w. leave I_j alone

$R_1, \ldots, R_{n+k}$ ...

| CPY 1 n+k+1 ; $I_\emptyset$
| CPY 2 n+k+2
| CPY 3 n+k+3
| . . . .
| CPY n n+k+n ; I_{n-1}

B1 is a copy of Π_1 n added to numbers of all instructions referred and final halt HALT instruction deleted.

2 CPY n+k+1 1 STZ n+1

⋮ ⋮

CPY n+n+k n STZ n+k+n

HALT

That does it

Lemma: Let $f : \omega^n \to \omega$ be partial recursive
Let ~~$\emptyset$~~ $i_\emptyset, \ldots, i_n$ be $(n+1)$ distinct integers
Let F be a finite set of integers disjoint from $i_\emptyset, \ldots, i_n$
and a finite set G (nullify regs)

Then there is a program Π s.t. if Π is started with $c(R_{i_k}) = x_i$ for $1 \le k \le n$, $c(R_{i_0}) = 0 \leftarrow c(R_{i_\emptyset})$ be 0
and $c(R_j) = \emptyset$, for $j \in G$,
then Π will halt iff $f(\langle x_1, \ldots, x_n \rangle)$ is defined. When it halts,

$c(R_{i_0}) = f(\vec{x})$

. . . . . .

No register R_j with $j \in F$ will be touched during a run of Π,
upon halting

$c(R_{i_k}) = x_k$ for $1 \le k \le n$

$c(R_j) = 0 \quad \forall j \in G$

Should be clear

——x——

Proposition: Recursive functions are closed under composition.
Let $f : \omega^m \to \omega$, $h_1, \ldots, h_m : \omega^n \to \omega$ be recursive
Let $g : \omega^n \to \omega$ be defined by $g(\vec{x}) = f(h_1(\vec{x}), \ldots, h_m(\vec{x}))$
Then g is recursive

Let $m = n = 2$.
Let $f : \omega^2 \to \omega$ $h_1 : \omega^2 \to \omega$ $h_2 : \omega^2 \to \omega$

(contd) overleaf

i.e. $g(x_1, x_2) = f(h_1(x_1, x_2), h_2(x_1, x_2))$

Program for g

Let Π_1: compute $h_1(c(R_1), c(R_2))$ stores it in $c(R_3)$
and leaves $c(R_1), c(R_2)$ unchanged

Π_1 Π_2: computes $h_2(c(R_1), c(R_2))$ and stores it in $c(R_4)$
and leaves $c(R_1), c(R_2)$ unchanged

Π_2 computes $f(c(R_3), c(R_4))$ and store it in R_0

Π_3 HALT

This works!

K. Solovay ~~~ Wednesday

Lemma:

Recursive functions are closed under primitive recursion operation.

Proof:

will do special case $n=2$ (kindly look at $n=1$)

$f: \omega \to \omega$
$g: \omega^3 \to \omega$ } recursive

$h(x,0) = f(x)$

$h(x,y+1) = g(x,y,h(x,y))$

R∅ Answer
R1 x
R2 y-s
R3 s
R4 h(x,s)
R5 copy of h(x,s)

∵ I'm a logician rather than a mathematician I start w/ ∅ rather than 1.

Program for h

$\boxed{1}$ Put $f(c(R_1))$ in R_4

or $\boxed{2}$ SUB1 R2 END

~~ADD1 R3~~

3 CPY R4 R5

$\boxed{4}$ $g(c(R_1), c(R_3), c(R_5))$ into R4

5 ADD1 R3

$\boxed{6}$ TRA LOOP

$\boxed{7}$ CPY 4 ∅

$\boxed{8}$ HALT ; may require minor modification

Remarks (1) $\boxed{1}$ and $\boxed{4}$ stand for blocks of code
(2) Its important that in $\boxed{1}$ & $\boxed{4}$ input variables are not clobbered and scratch registers are reset to 0.

Easy to prove that at the start of the $(s+1)^{st}$ iteration of LOOP $c(R_1) = x$; $c(R_2) = y-s$; $c(R_3) = s$; $c(R_4) = h(x,s)$

If we have a prim recursive function of 1 variable

$g(0) = c$

$g(Sn) = f(n, g(n))$

where f is a rec fn

"Put c into R4"

STZ 4

$\left.\begin{array}{l}\text{ADD 1 4}\\ \text{ADD 1 4}\\ \cdots\\ \text{ADD 1 4}\end{array}\right\}$ c times

μ-recursion:

Do for $n = 1$

$f : \omega^2 \to \omega$

Define a partial function $g : \omega \to \omega$

$g(x) = \mu y\,[f(x, y) = 0]$ undefined if for no y, $f(x, y) = 0$.

R1 x

R2 cycles thru possible y's : $y = 0, 1, 2, \ldots$

R3 $f(x, y)$

1. STZ 2

LOOP $\boxed{2}$ compute $f(c(R_1), c(R_2))$ and put it in R_3

3. SUB 1 3 EXIT
4. ADD 1 2
5. TRA LOOP

EXIT 6. CPY 2 0

7. ~~EXIT~~ HALT

QED: every μ recursive function is recursive

$\mathcal{C}$ is a collection of total functions containing Z, S, π^n_i's closed under composition and primitive recursion.

Examples:

1. $\mathcal{C}$ is μ recursive functions
2. $\mathcal{C}$ is primitive recursive functions

$$F(a,b,c) = G(H(b,c), K(G(b,c,a),a), c)$$

we will show that if
$G : \omega^3 \to \omega$
$K : \omega^2 \to \omega$
$H : \omega^2 \to \omega$

are all in $\mathcal{C}$ so is F.

Define a series of functions $F_i : \omega^3 \to \omega$ as follows:

$\vec{v}$ is (v_1, v_2, v_3)

$F_1(\vec{v}) = v_1$ $\qquad$ $F_4(\vec{v}) = H(v_2, v_3)$

$F_2(\vec{v}) = v_2$ $\qquad$ $F_5(\vec{v}) = G(v_2, v_3, v_1)$

$F_3(\vec{v}) = v_3$ $\qquad$ $F_6(\vec{v}) = K(G(v_2, v_3, v_1), v_1)$

F_1 is Π^3_1 so its in $\mathcal{C}$

sim F_2 is Π^3_2, F_3 is Π^3_3

$F_4 = H(F_2(\vec{v}), F_3(\vec{v}))$ ($F_4 \in \mathcal{C}$) (composition)

sim $F_5 \in \mathcal{C}$

$F_6 = K(F_5, F_1)$ as my laziness asymptotically goes to ∞.

$F(v_1, v_2, v_3) = G(F_4(\vec{v}), F_6(\vec{v}), F_3(\vec{v}))$

Prop (1) Every constant function lies in $\mathcal{C}$ (in any # of vars)

(2) '$+$' is in $\mathcal{C}$

$$x + 0 = x \quad = \pi_1^1(x)$$

$$x + Sy = S(x+y)$$

(3) '$\cdot$' is in $\mathcal{C}$

$$x \cdot 0 = 0$$

$$x \cdot Sy = x \cdot y + x$$

Consider $pd(0) = 0$ $\qquad pd(x) = x-1$ if $x > 0$

$pd(Sx) = x$ $\qquad pd(0) = 0$

Then 'pd' is in $\mathcal{C}$

Consider $x \dot{-} 0 = x$ $\qquad x \dot{-} y = 0$ if $y > x$

$x \dot{-} Sy = pd(x \dot{-} y)$ $\qquad = x - y$ o.w.

Then '$\dot{-}$' is in $\mathcal{C}$

Definition:

Let $R \subseteq \omega^n$. Then the characteristic function of R

$\chi_R : \omega^n \to \omega$ is defined as follows

if $\vec{x} \in R$ $\quad \chi_R(\vec{x}) = 1$

else if $\vec{x} \notin R$ $\quad \chi_R(\vec{x}) = 0$

Definition: R "is in" $\mathcal{C} \Leftrightarrow \chi_R \in \mathcal{C}$

$\mathcal{C}$ = prim recursive

μ recursive

recursive

Toolkit of primitive recursive functions

(1) x^y is p.r.

$$x^0 = 1$$
$$x^{y+1} = x^y \cdot x$$

(2) $sg(x) = 0$ if $x = 0$
$= 1$ if $x > 0$

$$sg(0) = 0$$
$$sg(Sx) = 1$$

3) $\overline{sg}(x) = 1$ if $x = 0$

$\overline{sg}(x) = 0$ if $x > 0$

$$\overline{sg}(x) = 1 \dot{-} sg(x)$$

Proposition: If $R(x_1, \dots, x_n)$, $S(x_1, \dots, x_n)$ are prim recursive predicates,

so are R or S, if R then S, $\neg R$, $R \Leftrightarrow S$, R and S

Proof: It suffices to handle and and not (complete)

$\chi_{\neg R}(\vec{x}) = \overline{sg}(\chi_R(\vec{x}))$: composition of prim rec

$\chi_{R \wedge S}(\vec{x}) = \chi_R(\vec{x}) \cdot \chi_S(\vec{x})$

Ch 6 §2 of Schoenfield

Proposition $<, \le, =$ are prim recursive

$\chi_<(x,y) = 1$ iff $x - y < 0$

$\therefore \chi_<(x,y) = sg(y \dot{-} x)$

$\chi_\le(x,y) \Leftrightarrow x < Sy$

$\chi_=(x,y) \Leftrightarrow (x \le y) \ \& \ (y \le x)$

Definition by cases:

You have $R_1(x_1, \ldots, x_n)$

$\cdots\cdots$

$R_m(x_1, \ldots, x_n)$

For any $\vec{x}$ exactly one of $R_1(\vec{x}), \ldots, R_m(\vec{x})$ holds

we has functions

$f_1, \ldots, f_m : \omega^n \to \omega$

Define a new function $f : \omega^n \to \omega$ by

$$\begin{aligned} f(x) &= f_1(\vec{x}) \text{ if } R_1(\vec{x}) \\ &= f_2(\vec{x}) \text{ if } R_2(\vec{x}) \\ &\cdots\cdots \\ &= f_m(\vec{x}) \text{ if } R_m(\vec{x}) \end{aligned}$$

Proposition: If $R_1, \ldots, R_m$ are prim recursive
$f_1, \ldots, f_m$ " "
and f is defined as above then
f is prim recursive

Proof: set $f(\vec{x}) = \sum_{i=1}^{m} f_i(\vec{x}) \cdot \chi_{R_i}(\vec{x})$

$\mu x < y\ R(z_1, \ldots, z_n, x)$ is defined as follows:

case (1) $(\exists x)\ (x < y \wedge R(z_1, \cdots, z_n, x))$

$$\mu x < y\ R(\vec{z}, x) = \mu x\ R(\vec{z}, x)$$

"If that was the worst stupidity of my life I'd be doing fine"

case (2) otherwise

$$\mu x < y\ R(\vec{z}, x) = y$$

Proposition:

If R is primitive recursive, then so is $g(\vec{z}, y) = \mu x < y\ R(\vec{z}, x)$

$g(\vec{z}, 0) = 0$ (default value)

$g(\vec{z}, Sy) =$ defn by cases ① $g(\vec{z}, y) < y$

$$g(\vec{z}, Sy) = g(\vec{z}, y)$$

case ② $g(\vec{z}, y) = y$ and $R(\vec{z}, y)$

$$g(\vec{z}\ Sy) = y$$

case ③ o.w.

$$g(\vec{z}, Sy) = Sy$$

details:

$P(\vec{z}, y, r)$:

if $r < y$,

$P(\vec{z}, y, r) = s$

if $r \geq y$ and $R(\vec{z}, y)$

$P(\vec{z}, y, r) = y$

o.w.

$P(-) = y + 1$

$g(\vec{z}, 0) = Z_n(\vec{z})$

$g(\vec{z}, Sy) = P(\vec{z}, y, g(\vec{z}, y))$

Limited Quantifiers:

$(\exists x < y)\ R(\vec{z}, x)$ means

$(\exists x)\ (x < y \wedge R(\vec{z}, x))$

$(\forall x < y)\ R(\vec{z}, x)$ $\quad (\forall x)(x \rightarrow y \rightarrow R(x, y))$

Easy fact:

$(\forall x < y)\ R(\vec{z}, x) \iff \neg (\exists x < y) \neg R(\vec{z}, x)$

oposition:

If $R(z_1, \cdots, z_n)$ is prim recursive, then so is

$(\exists x < y)\ R(\vec{z}, x) = S(\vec{z}, y)$

$S(\vec{z}, 0) = F$

$S(\vec{z}, sy) = S(\vec{z}, y) \vee R(\vec{z}, y)$

Corollary:

If R is prim recursive, so is $(\forall x<y) R(\vec{z}, x)$

use easy fact & result for $(\exists x<y)$

Caution:

Not true that if $R(\vec{z}, x)$ is prim recursive then $(\exists x) R(\vec{z}, x)$ is recursive

Example:

$n!$ is prim recursive

$0! = 1$

$(Sn)! = (Sn) \cdot n!$

x divides $y \iff (\exists z \le y)(x \cdot z = y)$

$\iff (\exists z < Sy)(x \cdot z = y)$

$\therefore$ "x divides y" is prim recursive

x is prime $\iff (x \geq 2) \wedge (\forall y \le x)(y | x \to ((y = x) \vee (y = 1)))$

$\therefore$ x is primitive recursive

Theorem: (Euclid) If $n \geq 2$ then there is a prime p : $n < p \le n!+1$

Proof: Let q be the least divisor of $n!+1$ (> 1)

Claim that q is prime

If $q \le n$, q divides $n!$

but q divides $n!+1$

$\Rightarrow q | 1$ which is absurd

Theorem: The function $i \to p_i$ is prim recursive

$$p_0 = 2$$

$$p_{i+1} = \mu z \leq (p_i! + 1)\ [\, z \text{ is prime} \wedge z > p_i \,]$$

This clearly works

Fundamental Thm of Arithmetic:

Obvious $\exists$ prime factorization

uniqueness tougher (not true in some extensions)

Theorem (Fundamental Theorem of Arithmetic):

Let $n \geq 2$, $n \in \omega$.

(Existence) Then $\exists\, r \geq 1$ and primes $p_1, \ldots, p_r$ with

$p_1 \leq p_2 \leq \ldots \leq p_r$

$n = p_1 \cdots p_r$; trivial proof (we) take that as known or ignored"

(Uniqueness) If also $n = q_1 \cdots q_s$
with $s \geq 1$ $q_1 \leq q_2 \leq q_3 \cdots \leq q_s$. q_i's prime

Then $r = s$ and $p_i = q_i$ for $1 \leq i \leq r$

Based on $p \mid s \cdot t \Rightarrow p \mid s$ ~~and~~ or $p \mid t$ (p prime)

Definition:

Let $\langle x_0, \ldots, x_n \rangle$ be a finite sequence of integers

To this we associate a a number $\langle\langle x_0, \ldots, x_{n-1} \rangle\rangle$

as follows

$$\prod_{i<n} p_i^{x_i+1}$$

$6 = 2 \cdot 3$

$12 = 2 \cdot 2 \cdot 3$

$65 = 5 \cdot 13$

$\langle\langle\rangle\rangle = 1$ (Universal convention that empty sequence is 1)

$\langle\langle 0 \rangle\rangle = 2$

$\langle\langle 1, 0 \rangle\rangle = 12$

$\langle\langle 1 \rangle\rangle = 4$

Remark:

65 is not a sequence #

Defn: $Seq = \{\langle\langle x_0, \ldots, x_{n-1} \rangle\rangle : n \in \omega \quad x_i\text{'s} \in \omega\}$

Lemma: Seq is prim recursive

$x \in Seq \iff$ (1) $x \geq 1$

(2) If p divides x, p is prime;

and $q < p$ q is prime

then q divides x

$(\forall p \leq x)(\forall q \leq x)$ [———]

clearly works

Lemma:

There is a primitive recursive fcn lh

such that if $x \in Seq$ & $x = \langle\langle a_\emptyset, \ldots, a_{r-1}\rangle\rangle$

then $lh(x) = r$

Proof:

Take $lh(x) = (\mu z \leq x)$ (p_z doesn't divide x)

Prop:

There is a prim rec. fn $(x)_i$ (a fn of x, i) such that

if $x = \langle\langle a_\emptyset, a_1, \ldots, a_{r-1}\rangle\rangle$

and $i < r$

then $(x)_i = a_i$

$(x)_i$ ~~is~~ $= (\mu z \leq x)$ (p_i^{z+2} doesn't divide x)

Remark $p_i \geq i+2$

$2^x \geq x+1$ for all $x \in \omega$

Proposition:

There is a p.r. function of two variables $a * b$ such that

if $a = \langle\langle x_0, \ldots, x_{m-1} \rangle\rangle$

$b = \langle\langle y_0, \ldots, y_{n-1} \rangle\rangle$

then $a * b = \langle\langle x_0, \ldots, x_{m-1}, y_0, \ldots, y_{n-1} \rangle\rangle$

Lemma:

Let $f : \omega^{n+1} \to \omega$ be primitive recursive

Define $g : \omega^{n+1} \to \omega$

and $h : \omega^{n+1} \to \omega$

as follows

$g(\vec{x}, z) = \sum_{i<z} f(\vec{x}, i)$ (Empty sum is 0)

$h(\vec{x}, z) = \prod_{i<z} f(\vec{x}, i)$ (Empty product is 1)

as in any assoc monoid the empty iterated product is the unit of the monoid

Then g and h are prim recursive

$a * b = a \cdot \prod_{i<lh(b)} p_{lh(a)+i}^{(b)_i + 1}$; visibly a prim recursive fn of a and b.

Let $f : \omega^{n+1} \to \omega$

we associate a new function $\bar{f} : \omega^{n+1} \to \omega$ as follows

$\bar{f}(\vec{x}, z) = \langle\langle f(\vec{x}, 0), \ldots, f(\vec{x}, z-1) \rangle\rangle$

Note: $\bar{f}(x, 0) = 1$ (empty)

Proposition: f is primitive recursive $\Leftrightarrow$ $\bar{f}$ is primitive recursive

$$f(\vec{x}, z) = \left[\bar{f}(\vec{x}, z+1)\right]_z$$

① $\bar{f}$ is prim rec
② defn is correct

$$\bar{f}(\vec{x}, 0) = 1$$
$$\bar{f}(\vec{x}, z+1) = \bar{f}(\vec{x}, z) \cdot p_z^{f(\vec{x}, z)+1}$$

so if f is prim recursive so is $\bar{f}$

Proposition: (Course of values recursion)

Let $g : \omega^n \to \omega$

define f by

$$(*) \quad f(\vec{x}, z) = g(\vec{x}, \bar{f}(\vec{x}, z))$$

Note that $\exists$ unique f satisfying this equation

Clear that there is a unique f satisfying $(*)$

Proposition: If g is prim rec so is f.

Proof: ETS $\bar{f} : \omega^n \to \omega$ is prim recursive

$$\bar{f}(\vec{x}, 0) = 1$$
$$\bar{f}(\vec{x}, z+1) = \bar{f}(\vec{x}, z) \cdot p_z^{g(\vec{x}, \bar{f}(\vec{x}, z))+1}$$

So clearly $\bar{f}$ is prim recursive

Code instructions by number thus

$$\begin{array}{lll} \text{ADD1 } i & \longrightarrow & \langle\langle 0, i\rangle\rangle \\ \text{SUB1 } i\ j & \longrightarrow & \langle\langle 1, i, j\rangle\rangle \\ \text{CPY } i\ j & \longrightarrow & \langle\langle 2, i, j\rangle\rangle \\ \text{TRA } i & \longrightarrow & \langle\langle 3, i\rangle\rangle \\ \text{STZ } i & \longrightarrow & \langle\langle 4, i\rangle\rangle \\ \text{HALT} & \longrightarrow & \langle\langle 5\rangle\rangle \end{array}$$

Distinct instructions get distinct numberings

If $P = \langle I_0, \ldots, I_{n-1}\rangle$ is a program

$$*P = \langle\langle *I_0, \ldots, *I_{n-1}\rangle\rangle$$

Suppose a register machine is in a state with program P

$$c(I) = n$$

$$c(R_j) = x_j \quad \text{(almost all zero)}$$

Code this state by the following number

$$2^{*P} \cdot 3^{c(I)} \cdot 5^{z}$$

where $z = \prod_{i \in \omega} p_i^{c(R_i)}$ (where we ignore the 1's in this infinite product thereby getting a finite product)

Wednesday

Lemma:

Let $n \geq 1$ Then the function $g : \omega^n \to \omega$ defined by
$$g(x_0, \dots, x_{n-1}) = \langle\langle x_0, \dots, x_{n-1} \rangle\rangle$$
is prim recursive

$$g(x_0, \dots, x_{n-1}) = p_0^{x_0+1} \, p_1^{x_1+1} \cdots p_{n-1}^{x_{n-1}+1}$$

Lemma:

"x is the Gödel # of an instruction" is prim rec

Proof:

This is $x = \langle\langle 0, i \rangle\rangle$ $\quad (\exists i \leq x)(x = \langle\langle 0, i \rangle\rangle)$

$\sim$ $\quad (x = \langle\langle 0, (x)_1 \rangle\rangle)$

$\vdots$

$x = \langle\langle s \rangle\rangle$

Lemma:

"x is the Gödel # of a program" is prim rec

Proof:

$x \in Seq \wedge (\forall i \leq lh(x))((x)_i$ codes an instruction)

code register machine state by

$\langle\langle s_0, s_1, s_2 \rangle\rangle$

where s_0 codes a program

s_1 is the contents of the Instruction Register

(*) $\displaystyle s_2 = \prod_{j \in \omega} p_j^{c(R_j)}$ $\quad c(R_j) = (s_2 \cdot p_j)_j$ ☺

Lemma:

Let $n \geq 1$

There is a primitive recursive function $I_n(p, x_1, \ldots, x_n)$

such that if p codes a program,

$I_n(p, x_1, \ldots, x_n)$ codes starting configuration

on inputs $x_1, \ldots, x_n$

$I_n(p, \vec{x}) = \langle\langle p, 0, \prod_{1 \leq j \leq n} p_j^{x_j} \rangle\rangle$

s codes an ~~instantaneous~~ instantaneous state of a register machine iff

(1) $s \in Seq$

(2) $lh(s) = 3$

(3) $(s)_0$ codes a program

(4) $(s)_2 \geq 1$

Proposition:

There is a primitive recursive predicate $H(s) \ni$

if s codes a register machine instantaneous state,

$H(s)$ iff s codes a halted state $\quad ((s)_0 = s_0, (s)_1 = s_1$, etc

$H(s) \quad (s)_1 \geq lh((s_0))$ or $(s_0)_{s_1} = \langle\langle 5 \rangle\rangle$

Halt

"This is the thing that drove savages to worship gods"

Lemma:

There is a primitive recursive fn,

$Next : \omega \to \omega \ni$ if s codes a non halted instantaneous register state,

$Next(s)$ codes the next such state

Proof:

Case 1 s is not the code of a register machine state or s is code of a halted state

Then $Next(s) = s$

As before we write $s_\emptyset$ for $(s)_\emptyset$ etc

$t = (s_\emptyset)_{s_1}$ (t is next instruction to execute) (t is not a HALT)

Case 2A $t = \langle\langle 0, i \rangle\rangle$ (ADD1 i)

Then $Next(s) = \langle\langle s_\emptyset, s_1 + 1, s_2 \cdot p_i \rangle\rangle$

case 2B $t = \langle\langle 1, i, j \rangle\rangle$ (SUB1 i j)

If p_i does divide s_2

then $Next(s) = \langle\langle s_\emptyset, s_1 + 1, s_2 \div p_i \rangle\rangle$

o.w. $Next(s) = \langle\langle s_0, j, s_2 \rangle\rangle$

here $(a \div b) = (\mu z \le a)(z \cdot b = a)$

case 2C $t = \langle\langle 2, i \rangle\rangle$ (TRA i)

$Next(s) = \langle\langle s_\emptyset, i, s_2 \rangle\rangle$

case 2D $t = \langle\langle 3, i, j \rangle\rangle$ (CPY i j)

$Next(s) = \langle\langle s_0, \cancel{s_1} s_1 + 1, w \rangle\rangle$ & $(\exists b \le s)(p_j \nmid b$ and $(\exists c \le s)\ s_2 = b \cdot p_j^c)$ and $u = b\, p_j^a$

where w is

$\mu u \le s\ p_s^s$

$(\exists a \le s)(p_i^a$ divides s_2 and p_i^{a+1} does not divide $s_2)$

very confusing

case 2 E $\quad t = \langle\langle 4, i \rangle\rangle \quad (>12\,i)$

$$Next(S) = \langle\langle s_0, s_1+1, w\rangle\rangle$$

where $w = (\mu b \le s)(\exists c \le s)(b \circ p_i^c = s_2)$ ($\nmid p_i \nmid b$)

Definition:

A computation of a register machine is a sequence of instantaneous states,

$s_0, \text{---}, s_n$

where s_{i+1} is the next state after s_i

s_n is halted

and for $i < n$, s_i is not halted

Code is a computation $\langle s_0, \ldots s_n\rangle$ by

$\langle\langle \# s_0, \ldots, \# s_n \rangle\rangle$

where $\# s_i$ codes s_i

$\# s_{i+1} = Next(\# s_i)$

Kleene T predicate:

$T(e, x, y)$

Definition:

$T(e,x,y)$ "iff" $\left.\right\}$ e is Gödel number of a program

$x \in \omega$

y codes a computation $\xrightarrow{\text{prim rec predicate}}$ &

$(y)_0 = I_1(e, x)$

Computation Sequence:

Sequence of instantaneous states

$s_0, \ldots, s_n \quad (n \geq 0)$

$\ni$ ① s_{i+1} is next state after s_i

② s_n is halted

③ For $i < n$, s_i isn't halted

Gödel # $\langle s_0, \ldots, s_n \rangle$

by $\langle\langle s_0, \ldots, s_n \rangle\rangle$

Proposition:

The set of Gödel numbers of computational sequences is prim rec

Proof:

Clear

$T(e, n, y)$: Kleene T predicate

(1) e is G# of a program

(2) y is G# of a computation sequence

(3) $(y)_0 = I(e, n)$ (starting state for program with #e on input n.)

Clear that T is prim recursive

Notation:

Write $(x)_{i,j}$ for $((x)_i)_j$

Proposition:

There is a prim recursive function U (for "upshot") such that if s is the Gödel number of a computation sequence, then $U(s)$ is contents of R_0 i.e. $c(R_0)$ in the final state of s.

$$U(s) = ((s)_{lh(s)-1,2^*}, 2)_0$$

[Pr]oposition:

Let f be a partial recursive function of 1 variable

$f: \omega \to \omega$

Let a program for f be having G^*

$f(x) \simeq U(\mu y\, T(e,x,y))$

$\uparrow$

one side defined iff other is
if both are defined, they are equal

[P]roof:

clear

[Co]rollary:

Let $f: \omega \to \omega$ be recursive. Then f is recursive.

Proof: Let e be a G^* for a program for f;
$f(x) = U(\mu y\, T(e,x,y))$ and $\forall x \exists y\, T(e,x,y)$

so if we set $g(x) = \mu y\, T(e,x,y)$

Then T is primitive recursive

so g is μ recursive

so $f(x) = U(g(x))$ is μrecursive

Kleene predicate for n variables

$T_n(e, x_1, \ldots, x_n, y)$: (1) e codes a program

(2) y codes a computation sequence

(3) $(y)_0 = I_n(e, x_1, \ldots, x_n)$

If e is G# of a particular rec fn $f : \omega^n \to \omega$ we have

$$f(x_1, \ldots, x_n) \simeq U(\mu y\, T_n(e, x_1, \ldots, x_n, y))$$

Definition: $\Phi_e(x)$

(1) If e is a G# of a program
$\Phi_e(x)$ is value computed on i/p x.
Might be undefined

(2) If e is not # of a program, $\Phi_e(x)$ is undefined.

Theorem: (Turing Enumeration Theorem)

There is a partial recursive function of two variables $\Phi(x,y)$
$\ni \Phi(x,y) \simeq \Phi_x(y)$

Proof:

$\Phi(e,y) \simeq U(\mu y\, T(e,x,y))$

Routine to write a program for Φ

⓪ If e not G# of a program, go into a loop
① Search for least $y \ni T(e,x,y)$
② If find it, compute $U(y)$ and output it.

Definition:

$K = \{e \mid \Phi_e(e) \text{ is defined}\}$

Theorem:

K is not recursive

Proof: Will assume K is recursive & reach a contradiction.

Define a partial function h thus

$h(j) = 0$ if $\Phi_j(j)$ is undefined

$h(j)$ is undefined if $\Phi_j(j)$ is defined

If χ_K is recursive

$h(e) = \mu y [\chi_K(e) = 0 \wedge y = 0]$

This clearly defines h (assuming K is partial recursive)

and h is partial recursive

So $h = \phi_j$

so $h(j)$ is defined $\Leftrightarrow$ $\Phi_j(j)$ is defined

But also $h(j)$ defined $\Leftrightarrow$ $\Phi_j(j)$ is not defined

So $0 = 1$

Upshot: K is not recursive

Quasi HW #2

1. Devise a G#ing of prim recursive functions. Might be that same fn has several G#s. Being a G# of a prim rec fn should be prim rec.
2. For your G#ing prove there is a prim rec fn P such that $P(n)$ is the Godel # of a program that computes the prim recursive function w/ index n
3. There is a recursive function that is not prim recursive.

Hint: "Enumeration Thm" fn of 2 var which enums any fn of 1

Diagonal Method

(a) $h_1 : \omega \longrightarrow \{0,1\}$

(b) h_2 · can be chosen st $\forall$ prim rec f, $h_2(n) > f(n)$ for all but finitely many n.

Wednesday

$Q \quad +, \cdot, 0, S$

Q1 $Sx = Sy \to x = y$

Q2 $0 \neq Sx$

Q3 $x \neq 0 \to (\exists y)(x = Sy)$

Q4 $x + Sy = S(x+y)$

Q5 $x + 0 = x$

Q6 $x \cdot 0 = 0$

Q7 $x \cdot Sy = (x \cdot y) + x$

$N \quad <$

$\neg(x < 0)$

$(x < Sy) \leftrightarrow (x < y \vee x = y)$

$(x < y) \vee (x = y) \vee (y < x)$

$\mathfrak{N} = \langle \omega; 0, S, +, \cdot \rangle$

Define by induction on $n \in \omega$ a term $\underline{n}$ (the numeral for n)

$\ni$ $\underline{0} = 0$

$\underline{x+1} = S \,\hat{}\, \underline{x}$

eg $\underline{3} = SSS0$

Q can't prove $+$ is commutative; $x \leq x$; etc

can prove things about standard numbers

Definition:

Let ϕ be a sentence of L_Q

Q decides $\phi \Leftrightarrow Q \vdash \phi$ or $Q \vdash \neg\phi$

Definition:

Let $\phi(v_0, \ldots, v_{n-1})$ be a formula. Then Q numeralwise decides ϕ (ϕ is numeralwise decidable) if whenever $x_0, \ldots, x_{n-1} \in \omega$ then Q decides $\phi(\underline{x_0}, \ldots, \underline{x_{n-1}})$

Definition:

Let $R \subseteq \omega^n$ be a relation. $\phi(v_0, \ldots, v_{n-1})$ is a formula of L_Q with indicated free variables.

then ϕ numeralwise represents R in Q if $\langle x_0, \ldots, x_{n-1} \rangle \in R$ then $Q \vdash \phi(\underline{x_0}, \ldots, \underline{x_{n-1}})$ and if $\langle x_0, \ldots, x_{n-1} \rangle \notin R$ then $Q \vdash \neg\phi(\underline{x_0}, \ldots, \underline{x_{n-1}})$

Remark.

TFAE

(1) $\phi(v_0, v_1, \ldots, v_{n-1})$ numeralwise R in Q

(2) (a) ϕ defines R in η

(ie $R = \{ \langle \vec{x} \rangle \mid \eta \models \phi(\vec{x}) \}$

(b) ϕ is num. decidable in Q

The proof is left to you

Let $f: \omega^n \to \omega$ be a function

$\text{Graph}(f) \subseteq \omega^{n+1} = \{ \langle \vec{x}, y \rangle \mid f(\vec{x}) = y \}$

We say f is numeral representable in Q $\overset{\Delta}{\Leftrightarrow}$

Graph(f) is numeral wise representable in Q

Let $f: \omega^n \to \omega$ ~~be recursive~~

Theorem:

If f is recursive, f is num rep in Q

Proposition:

Let $s, t \in \omega$

$Q \vdash \underline{s} + \underline{t} = \underline{s+t}$

(eg $Q \vdash SS0 + SS0 = SSSS0$)

Proof:

We prove (by "induction in the metatheory") on t,

that for all s

$Q \vdash \underline{s} + \underline{t} = \underline{s+t}$

$t = 0$

To see $Q \vdash \underline{s} + \underline{0} = \underline{s+0} = \underline{s}$ This is Q4

Suppose prop holds at t.

To see it holds for t+1.

work in Q

$\underline{s} + \underline{t+1} = \underline{s} + S\underline{t} \overset{Q5}{=} S(\underline{s} + \underline{t}) \overset{\text{ind hyp}}{=} S(\underline{s+t}) = \underline{s+t+1}$ QED

Proposition:

Let $n \in \omega$

$$Q \vdash (x \leq \underline{n} \leftrightarrow (x = \underline{0} \lor x = \underline{1} \text{ or } \ldots \quad x = \underline{n}))$$

Proof:

By induction on n (in Metatheory)

case (a) $n = 0$

$x \leq 0 \leftrightarrow (\exists y)(y + x = \underline{0})$

By $\underline{Q3}$ $x = 0 \lor (\exists z)(x = Sz)$

$\underline{0} + \underline{0} = \underline{0}$

so $\underline{0} \leq \underline{0}$

$y + Sz = \underline{0}$

$\hookrightarrow S(y + z) = 0$

contradicting Q2

So $Q \vdash x \leq \underline{0} \leftrightarrow x = \underline{0}$

K. Solovay

112 — Recursion theory

Friday

Q1 $Sx = Sy \to x = y$

Q2 $0 \neq Sx$

Q3 $x \neq 0 \to (\exists y)(x = Sy)$

Q4 $x + 0 = x$

Q5 $x + Sy = S(x+y)$

Q6 $x \cdot 0 = 0$

Q7 $x \cdot Sy = x \cdot y + x$

Proposition:

Let $n \in \omega$

$Q \vdash x \leq \underline{n} \leftrightarrow (x = \underline{0} \vee x = \underline{1} \vee \dots x = \underline{n})$

Proof:

By induction in the metatheory on n

$n = 0$ Done last time

| $x = \underline{0} \to x = \underline{n}$
| $\underline{0} \leq \underline{n}$ $(\exists y)(u + \underline{0} = \underline{n})$
| $Q \vdash \underline{n} + \underline{0} = \underline{n}$

Reason in Q

$\leftarrow$ is trivial: If $i \leq n$, $Q \vdash \underline{(n-i)} + \underline{i} = n$

so $Q \vdash \underline{i} \leq \underline{n}$ $(= (\exists u)(u + \underline{i} = \underline{n})$

$\to$ Work in Q

assume $x \leq \underline{n+1}$ $(\exists y)(y + x = \underline{n+1})$

Two cases

(i) $x = 0 : \to (x = \underline{0} \vee \dots x = \underline{n+1})$ correct

(ii) $x = Su$

$y + Su = \underline{n+1}$ ~~$= S\underline{n}$~~ $= S\underline{n}$

$\quad \parallel$

$S(y+u) = S\underline{n}$

$y + u = \underline{n}$

$u \leq \underline{n}$ using IH

$u = \underline{0} \vee u = \underline{1} \vee \dots u = \underline{n}$

$x = \underline{S0} \vee x = \underline{S1} \vee \dots x = S\underline{n}$

$\to x = \underline{1} \vee x = \underline{2} \vee \dots x = \underline{n+1}$

$x = \underline{0} \vee \dots \quad x = \underline{n+1}$

Proposition:

Let $s, t \in \omega$

$Q \vdash \underline{s} \cdot \underline{t} = \underline{s \cdot t}$

Proof:

Uses Q6 and Q7 and preceding proposition. By induction in metatheory on t. Details left to you.

Proposition:

Let τ be a closed term of L_Q (ie τ is variable free)

Let $\eta(\tau) = n$

Then $Q \vdash \tau = \underline{n}$

"Hoping you're yawning along with me rather than getting lost"

Proof:

Induction in metatheory ~~of~~ on length of τ. Left to you

Proposition:

Let $n, m \in \omega$

(a) If $n = m$, $Q \vdash \underline{n} = \underline{m}$

(b) If $n \neq m$, ~~Q/H~~ $Q \vdash \neg(n = m)$

(a) is utterly trivial

(b) by induction on minimum of n and m

Case 1 $\min(n, m) = 0$

Say $n = 0$ $m \neq 0$

$Q \vdash \underline{0} \neq S(\underline{m-1})$ (Q2)

Case 2 $\min(n, m) > 0$

$Q \vdash \underline{m} = \underline{n} \rightarrow \underline{m-1} = \underline{n}$ ~~*~~ (Q1)

By IH $Q \vdash \underline{m-1} \neq \underline{n-1}$

So by logic $Q \vdash \neg(\underline{m} = \underline{n})$

Definition: $(t \leq s)$

stands for $(\exists w)(w + t = s)$

where w is some variable not appearing in t or s.

Corollary:

$x \le y$ is numeralwise decidable in Q

Proof:

Let n, and $m \in \omega$

To see Q decides $n \le m$

Case 1 $\quad n \le m$

So for some r, $r + n = m$

$Q \vdash \underline{r} + \underline{n} = \underline{m}$

$Q \vdash (\exists u)(u + \underline{n} = \underline{m})$

$Q \vdash \underline{n} \le \underline{m}$

Case 2 $\quad n > m$

$Q \vdash x \le \underline{m} \leftrightarrow [x = \underline{0} \vee \ldots \vee x = \underline{m}]$

$Q \vdash \underline{n} \le \underline{m} \leftrightarrow [\underline{n} = \underline{0} \vee \ldots \vee \underline{n} = \underline{m}]$

But $Q \vdash \underline{n} \ne \underline{0}$

$\vdots$

$Q \vdash \underline{n} \ne \underline{m}$

so $Q \vdash \neg(\underline{n} \le \underline{m})$

Theorem:

Let $n \in \omega$

$Q \vdash (x \le \underline{n})$ or $(\underline{n} \le x)$ $\quad$ ($Q \nvdash (x \le y) \vee (y \le x)$ But will do has $Q \vdash (x \le \underline{n}) \vee (\underline{n} \le x)$)

Proof:

By induction on n:

$n = \underline{0}$

$Q \vdash x + \underline{0} = x \quad (Q4)$

$Q \vdash \underline{0} \le x$

$Q \vdash \underline{0} \le x \vee x \le \underline{0}$

Assume $Q \vdash x \le \underline{n}$ or $\underline{n} \le x$

To see $Q \vdash x \le \underline{n+1}$ or $\underline{n+1} \le x$

need the following lemma

Lemma:

$Q \vdash x + \underline{n} = S^{n} x$

Proof:

clear $x + (\underline{n+1}) = x + S\underline{n} =$

$S(x + \underline{n}) =$

$S(S^{n} x) =$

$S^{n+1} x$

QED

Case 1 $x \le \underline{n}$

$x \le \underline{n} \to (x = \underline{0} \vee \cdots x = \underline{n}) \to x \le \underline{n+1}$

Case 2

$\underline{n} \le x$

$w + \underline{n} = x$

$Q \vdash 0 + x = x$ (???)

case 2A $w = 0$

$x = \underline{0} + \underline{n} = \underline{n}$ so $x \le \underline{n+1}$

So $x \le \underline{n+1}$ or $\underline{n+1} \le x$

case 2B $w = St$

$x = St + \underline{n} = S^{n+1} t = t + \underline{n+1}$

so $\underline{n+1} \le x$ (since $x = t + \underline{n+1}$)

so $(x \le \underline{n+1})$ or $(\underline{n+1} \le x)$

so $Q \vdash x \le \underline{n+1}$ or $\underline{n+1} \le x$

We prefer this weak theory Q so that its clear what is being proved & what is just already there. Even this weak thy fails to be decidable

Definition:

$\Delta_\emptyset$ formulas

Smallest class of formulas $\emptyset$:

(1) $t_1 = t_2$ is $\Delta_\emptyset$ (t_1, t_2 terms)

(2) $t_1 \le t_2$ is $\Delta_\emptyset$

(3) ϕ, ψ are $\Delta_\emptyset$, so are $\phi \vee \psi$, $\neg \phi$

(4) If $\phi(x)$ is $\Delta_\emptyset$ so is $(\exists x \le t)\, \phi(x)$ $(\exists x)(x \le t \wedge \phi(x))$ (x doesn't appear in t)

Proposition:

If ϕ is Δ_0, ϕ is numeralwise decidable (in Q)

Proof:

ETS that a Δ_0 sentence w(no free variables is decidable (in Q)

Proof:

~~By induction on number of $\exists$ symbols & for this no. fixed by induction on length~~

We prove this by induction on

(a) the no. of $\exists$'s in ϕ

and (within this induction) by induction on length

case 1 ϕ is $t_1 = t_2$

Let $n_1 = \eta(t_1)$

$n_2 = \eta(t_2)$

$Q \vdash t_1 = \underline{n_1}$

$Q \vdash t_2 = \underline{n_2}$

and Q decides $\underline{n_1} = \underline{n_2}$

So Q decides $t_1 = t_2$

case 2 ϕ is $t_1 \leq t_2$

can do for numerals $\Rightarrow$ can have $Q \vdash \phi$ or $Q \vdash \neg\phi$

i.e. Q decides ϕ

case 3 $\phi = \psi \vee \chi$

clear

case 4 $\phi = \neg\psi$

clear

Case 5 ϕ is $(\exists x \le t)\,\psi(x)$

Let $n = \eta(t)$

$Q \vdash t = \underline{n}$

$Q \vdash x \le \underline{n} \Leftrightarrow x = \underline{0} \vee \ldots \vee x = \underline{n}$

$Q \vdash (\exists x)(x \le t \wedge \psi(x) \leftrightarrow (\exists x)(x \le n \wedge \psi(x))$

$\leftrightarrow \psi(\underline{0}) \vee \psi(\underline{1}) \ldots \psi(\underline{n})$

(each has fewer existential quantifiers and so can apply IH.

By IH

Q decides $\psi(\underline{i})$ for $i \le n$

So Q decides $\psi(\underline{0}) \vee \ldots \vee \psi(\underline{n})$

But $Q \vdash \psi \leftrightarrow (\psi(\underline{0}) \vee \ldots \vee \psi(\underline{n}))$

so Q decides ϕ

$\boxtimes$ QED

Theorem:

Let $f : \omega^n \to \omega$ be recursive

Then f is representable in Q. (Converse is true too, will do later)

Proposition:

S is representable in Q

$y = Sx$ works to express graph(S) Δ_0 formulas $\Rightarrow$ numerically decidable

Proposition:

Z is representable in Q

$y = 0$ works

Proposition:

$\pi^n_i(x_1, \ldots, x_n) = y$ is rep in Q $\quad$ $x_i = y$ works

Theorem:
Every recursive function is num. rep. in Q

Already checked Z, S, π^n_i are num rep in Q

Proposition:
The class of fns num rep in Q is closed under μ recursion

Lemma:
Class of numeralwise decidable formulas is closed under Boolean operations and limited quantification.
(Proved last time)

Proof:
For ease of ~~notation~~ consider $f : \omega^2 \to \omega$ such that f is numeralwise reprentable in Q
by $\theta(x, y, z)$ $(F(x,y) = z)$
and $\forall x \exists y\, F(x,y) = 0$

Let $G(x) = \mu y[F(x,y) = 0]$
To see G is num rep in Q
$\boxed{G(x) = y}$
$\psi(x,y) = \theta(x,y,0) \wedge (\forall z \leq y)\, \theta(x,z,0) \to z = y$
Take it as completely clear that ψ defines G in std model $\mathfrak{N}$
By lemma, since θ is num decidable so is ψ
So by an early remark, ψ num rep G

Proposition:

The class of num rep functions is closed under composition. Will give proof in special case, functions of one variable, proof is totally general.

Let $f: \omega \to \omega$ be num rep in Q by $\Theta(x,y)$ $(F(x)=y$

$G:\omega\to\omega$ be num rep by $\Psi(y,z)$ $(G(y)=z))$

To see $H:\omega\to\omega$ is defined by $H(x)=G(F(x))$

then H is num rep in Q

here's the obvious try which doesn't work.

$(\exists y)[\Theta(x,y) \wedge \Psi(y,z)] = \chi(x,z)$

claim:

1. χ defines H in std model
2. If $H(m)=n$, then $Q \vdash \chi(\underline{m},\underline{n})$

$n = H(m) = G(F(m)) = G(p)$

Let $p = F(n)$ so $n = G(p)$ since $n = H(m)$

$Q \vdash \Theta(\underline{m},\underline{p})$

$Q \vdash \Psi(\underline{p},\underline{n})$

$Q \vdash (\exists y)[\Theta(\underline{m}, \cancel{\underline{p}}\, y) \wedge \Psi(y,\underline{n})]$

$Q \vdash \chi(\underline{m},\underline{n})$

suppose $q \neq n$

can $Q \vdash \neg\chi(\underline{m},\underline{q})$

know $\Theta(\underline{m},\underline{x})$ happens only when we want it to in standard model. Don't know if holds in non std models!! could pull out a nonstd model (nrgn) 2

cultures in mathematics
- relating proof to stupidity

Definition:

Let $F:\omega^n\to\omega$ be a function

Let $\theta(x_1,\dots,x_n,y)$ be a formula of Q

Then θ numeralwise functionally represents F if: whenever $a_1,\dots,a_m\in\omega$ and $p=F(a_1,\dots,a_n)$

Then $Q\vdash\theta(\underline{a_1},\dots,\underline{a_n},y)\leftrightarrow y=\underline{p}$

Proposition:

If θ num. fun. rep. F,

then θ num rep F.

Proof:

Let $x_1,\dots,x_n\in\omega$. Let $p=F(x_1,\dots,x_n)$

Have to see

~~$Q\vdash\theta(\underline{x_1},\dots,\underline{x_n},\underline{p})$~~

$Q\vdash\theta(\underline{a_1},\dots\underline{a_n},\underline{p})$ (1)

(2) If $q\neq p$, $Q\vdash\neg\theta(\underline{a_1},\dots,\underline{a_n},\underline{q})$

Proof of (1): $Q\vdash\theta(\underline{a_1},\dots,\underline{a_n},\underline{p})\leftrightarrow\underline{p}=\underline{p}$ comes from (*)

But $Q\vdash\underline{p}=\underline{p}$

$\therefore Q\vdash(1)$

Proof of (2):

$Q\vdash\theta(\underline{a_1},\dots,\underline{a_n},\underline{q})\leftrightarrow\underline{q}=\underline{p}$

But $q\neq p\to Q\vdash\neg\underline{q}=\underline{p}$

so $Q\vdash\neg\theta(\underline{a_1},\dots,\underline{a_n},\underline{q})$

Proposition:

Let $F: \omega^n \to \omega$ be num rep in Q

Then F is also is num fcn rep in Q

Proof: (For $n = 1$)

Let $\theta(x, y)$ num rep $F : \omega \to \omega$

Define $\psi(x, y)$ thus:

$\theta(x, y) \wedge (\forall z \le y)\,[\theta(x, z) \to z = y]$

To see ψ works

Let $m \in \omega$. Let $n = F(m)$

To see

$Q \vdash \psi(\underline{m}, y) \leftrightarrow y = \underline{n}$

$(\leftarrow)$: routine since $Q \vdash y \le \underline{n} \leftrightarrow (y = \underline{0} \vee \ldots \vee y = \underline{n})$

Remains to see

$Q \vdash [\theta(\underline{m}, y) \wedge (\forall z \le y)(\theta(\underline{m}, z) \to z = y] \to y = \underline{n}$

Work in $Q + [\cdots]$

To show: $y = \underline{n}$

Recall lemma: $Q \vdash y \le \underline{n} \vee \underline{n} \le y$

Case 1: $y \le n$

But then $y = \underline{0} \vee y = \underline{1} \vee \cdots\ y = \underline{n}$

But for $i < n$, $Q \vdash \neg\theta(\underline{m}, \underline{i})$

we are assuming $\theta(\underline{m}, y)$

we get $y = \underline{n}$

Case 2:

$\underline{n} \le y$

But $\theta(\underline{m}, \underline{n})$

so since $\underline{n} \le y$ & $\theta(\underline{m}, \underline{n})$, have $\underline{n} = y$ QED

Back to our main proof.

Now let $\theta_1(x,y)$ num fcn rep F.

Let $\psi_1(x,y)$ num fcn rep G

(We know θ_1, ψ_1 exist by preceding prop)

Let $\chi_1(x,z)$ be $(\exists y)[\theta_1(x,y) \wedge \psi_1(y,z)]$

I'll show: χ_1 num fcn rep H

Let $m \in \omega$

Let $p = F(m)$ So $n = H(m)$

$n = G(p)$

$Q \vdash \chi_1(\underline{m}, z) \Leftrightarrow z = \underline{n}$

$(\Leftarrow)$: As before

$(\rightarrow)$: ~~If~~ $Q \vdash \chi_1(\underline{m}, z)$: $(\exists y)[\theta_1(\underline{m}, y) \wedge \psi_1(y, z)]$

But $\theta_1(\underline{m}, y) \rightarrow y = \underline{p}$ (θ_1 num fcn rep F)

Work in $Q[\chi_1(\underline{m}, z)]$

So for some y,

$\theta_1(\underline{m}, y) \wedge \psi_1(y, z)$

But θ_1 num fcn rep F,

& $F(m) = n$

so $y = \underline{p}$

so $\psi_1(\underline{p}, z)$

so $z = m$

QED

So closure of numeral wise rep. functions under composition.

Lemma:

There is a function $\beta(c, d, i)$

such that

(1) For every $a_\emptyset, \dots, a_n$ of integers

there are integers c, d:

$\beta(c,d,i) = a_i$ for $0 \leq i \leq n$

(2) β is num rep in Q

$\beta(c,d,i) = rem(c, 1 + (c+i)d)$

onday

Lemma: (Godels β function)

There is a function $\beta : \omega^3 \to \omega$ such that

(1) β is num rep in Q

(2) if $a_0, \ldots, a_n$ is a sequence (finite) of integers (≥ 0) there are c, d in ω such that

$$\beta(c, d, i) = a_i \quad (0 \leq i \leq n)$$

Will assume lemma and prove the following propositions:

Prop: The class of functions num rep in Q is closed under prim recursion

Cor: (To proposition)

(1) Every recursive fcn is numeralwise rep in Q

(2) Map exp: $(x, y) \to x^y$ is num rep in Q

(3) There is a formula $\varphi(x, y, z)$ that expresses $x^y = z$ in $\mathcal{N}$ $\quad \mathcal{N} = \langle \omega, 0, S, +, \cdot \rangle$

Will prove proposition for functions of 2 variables (general case is entirely similar)

Situation: $F : \omega \to \omega$ $\left(F(x) = y\right)$ num fcn rep by $\theta(x, y)$

$G : \omega^3 \to \omega$ num fcn rep by $\psi(x, y, z, w)$ $(w = G(x, y, z))$

$\beta : \omega^3 \to \omega$ num fcn rep by $B(x, y, z, w)$ $(w = \beta(x, y, z))$;

Define $H: \omega^2 \to \omega$ by prim rec

$H(x, 0) = F(x)$

$H(x, Sy) = G(x, y, H(x, y))$

To see H is num rep in Q

The following formula num rep H in Q

$\chi(x, y, z)$: $\exists c, \exists d \ni$

[(0) $(\forall i \leq y)(\exists! u)$ $B(c, d, i, u)$
unique

(1) $(\exists u) B(c, d, \underline{0}, u) \wedge \theta(x, u)$

(2) $(\forall i \leq y)$ [if $Si \leq y$ then $\exists u \exists v$ $B(c, d, i, u) \wedge B(c, d, Si, v)$ & $\psi(x, i, u, v)$]

(3) $B(c, d, y, z)$]

↳ can take conjunction of all 4 formulas

We have to check

If $x, y, z \in \omega$ & $H(x, y) = z$,

then $Q \vdash \chi(\underline{x}, \underline{y}, \underline{z})$

For $0 \leq i \leq y$ $a_i = H(x, i)$

By Lemma (still to prove)

choose ~~c, d ∈ ω~~ $c, d \in \omega$

$\beta(c, d, i) = a_i$

for $0 \leq i \leq y$
with this c, d easy to check

$Q \vdash \chi(\underline{x}, \underline{y}, \underline{z})$

Remains to see: If $x, y, z \in \omega$ & $H(x,y) \neq z$
$Q \vdash \neg \chi(\underline{x}, \underline{y}, \underline{z})$

Reason in $Q + \chi(\underline{x}, \underline{y}, \underline{z})$ and get a contr.

~~and step by step prove~~
~~... c, d ... $\to B(c, d, \underline{i}, \underline{a_i})$~~
~~But ... c, d ...~~

$\exists c, d \ldots$
Now define $a_0, \ldots, a_y$ as before

and step by step prove
$\ldots c, d \ldots \to B(c, d, \underline{i}, \underline{a_i})$
for $0 \leq i \leq y$

But $\ldots c, d \ldots\ B(c, d, \underline{y}, \underline{z})$
$\& \ B(c, d, \underline{y}, \underline{a_y})$
So $\underline{z} = \underline{a_y}$

Absurd since $a_y = H(x,y) \neq z$

QED prop modulo lemma

$\omega = \{0, 1, 2, \ldots\}$
$\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$

Proposition:
Let $x, y \geq 1$ let z be their gcd of x and y.
Then there are integers α and β in $\mathbb{Z}$
$z = \alpha x + \beta \cdot y$

Let $W = \{z > 0 : \exists \alpha, \beta \in \mathbb{Z} : w = \alpha x + \beta y\}$

$W \neq 0$ $\quad x = 1 \cdot x + 0 \cdot y$

Let u be least in W.
(By general nonsense)
Will see that gcd of x and y is u.
We show first if s divides x, s divides y
then s divides u $\quad (s \mid u) \Rightarrow s \leq u$ (since $u \geq 1$)

$x = sa$
$y = sb$
$u = \alpha x + \beta y = (\alpha a + \beta b)s$
so s divides u

Need Lemma:

Lemma:
let $x \in \mathbb{Z}$
let $y \in \omega$, $y \geq 1$

Then $x = qy + r$ with $q \in \mathbb{Z}$ $r \in \omega$ $0 \leq r < y$

also if $x = q_1 y + r_1$ with $q_1 \in \mathbb{Z}$ $r_1 \in \omega$ $0 \leq r_1 < y$,

Then $q = q_1$ and $r = r_1$.

$r = \text{rem}(x, y)$
$0 = \text{rem}(x, 0)$

we now show u divides x
(u divides y is entirely similar)

If not $x = q \cdot u + r$ w/ $0 < r < u$
$(q \in x)$

$r = x - qu = x - q(\alpha x + \beta y)$
$= (1 - q\alpha)x + \beta y$

(*) Proof: since x & y are rel prime $\exists \alpha, \beta$ $\alpha x + \beta y = 1$
$a = u\beta y + v\alpha x$ will work

so $r \in \omega$, $0 < r < u$
contradiction
so u divides x.

Dfn. Let $x, y \geq 1$
then x, y are rel prime
$\Leftrightarrow \gcd(x, y) = 1$

Theorem: (CRT) (*)
Let $x, y \geq 1$ be rel prime Let $0 \leq u < x$ Let $0 \leq v < y$ (*)
Then $\exists a$: $\text{rem}(a, x) = u$ ~~rem(a, y) = v~~ $\text{rem}(a, y) = v$

Chinese Remainder Thm $(n=2)$

Let $a_0, a_1 \geq 1$ Let $0 \leq x_i < a_i$ Assume a_0, a_1 rel prime

Then there is a c such that:

(1) $0 \leq c < a_0 \cdot a_1$

(2) $\text{rem}(c, a_i) = x_i \quad (i = 0, 1)$

Special case $x_0 = 1$
$x_1 = 0$

C_*:
$\text{rem}(C_*, a_0) = 1$
$\text{rem}(C_*, a_1) = 0$

$x_0 = 0$
$x_1 = 1$

C_{**}:
$\text{rem}(C_{**}, a_0) = 0$
$\text{rem}(C_{**}, a_1) = 1$

Then $x_0 \cdot C_* + x_1 \cdot C_{**}$ will work for c

Proof:

Since a_0, a_1 are rel prime there are $\alpha, \beta \in \mathbb{Z}$

~~$\alpha a_0 + \beta a_1$~~ $\alpha a_0 + \beta a_1 = 1$

$\text{rem}(\alpha a_0, a_0) = 0$

$\text{rem}(\alpha a_0, a_1) =$

$\text{rem}(1 - \beta a_1, a_1) = 1$

So αa_0 works for C_{**}

βa_1 works for C_*

Let $u = x_0 \cdot \beta a_1 + x_1 \cdot \alpha a_0$ (Above was motivation)

Claim: $\text{rem}(u, a_0) = x_0 \quad \text{rem}(u, a_1) = x_1$

$u = x_0(\beta a_1) + x_1(\alpha a_\emptyset)$
$= x_0(1 - \alpha a_\emptyset) + x_1(\alpha a_\emptyset)$
$= x_0 + [(x_1 - x_0) \cdot \alpha] a_\emptyset$
$\Rightarrow \text{rem}(u, a_\emptyset) = x_0$
an entirely similar calculation yields $\text{rem}(u, a_1) = x_1$
But $u = r a_\emptyset a_1 + c$ with $0 \le c < a_\emptyset a_1$ and $r \in \mathbb{Z}$
Claim:
$\text{rem}(c, a_\emptyset) = x_0$
$\text{rem}(c, a_1) = x_1$
we know $\text{rem}(u, a_\emptyset) = x_0$
$u = \sigma a_\emptyset + x_0 \quad \sigma \in \mathbb{Z}$
$\therefore \quad c = u - r a_\emptyset a_1$
$= \sigma a_\emptyset - r_\emptyset a_1 + x_0$
$= (\sigma - r a_1) a_\emptyset + x_\emptyset$

So $\text{rem}(c, a_\emptyset) = x_0$
The other calculation is similar.

Proposition:
Let p be prime
Suppose $p | a \cdot b$ with $\cancel{p, a, b} \in \mathbb{Z}$ $a, b \geq 1$
Then $p | a$ or $p | b$
Proof:
Suppose $p \nmid a$ to see $p | b$
$p \nmid a \Rightarrow GCD(p, a) = 1$
$\Rightarrow \exists \alpha, \beta \in \mathbb{Z}$
$\Rightarrow \exists\ \alpha p + \beta a = 1$
$\Rightarrow \alpha p b + \beta a b = b$
But look: $p | p \Rightarrow p | \alpha p b \quad p | ab \Rightarrow p | ab$

so $p \mid LHS$ $\quad (p \mid x \wedge p \mid y \Rightarrow p \mid (x+y))$

so $p \mid b$

QED

Corollary:

If $a_\emptyset, \ldots, a_n \geq 1$ and p is prime, $p \mid a_\emptyset \cdots a_n$

then for some i, $0 \leq i \leq n$ we have $p \mid a_i$

Chinese Remainder Theorem (General case)

Let $a_\emptyset, \ldots, a_n$ be positive integers

Assume if $0 \leq i < j \leq n$

then a_i is rel prime to a_j

Let $0 \leq x_i < a_i$

Then $\exists\, c$

(1) $0 \leq c < a_\emptyset \cdots a_n$

(2) $\text{rem}(c, a_i) = x_i$

Proof: Induction on n

case $n = 0$. Trivial

case $n = 1$ already done

case $n = k+1$

Notice $a_\emptyset, \ldots, a_k$ are rel prime

So by IH

$\exists\, c_1 : 0 \leq c_1 < a_\emptyset \cdots a_k$,

$\text{rem}(c_1, a_i) = x_i$ for $0 \leq i \leq k$

Claim:

~~$a_\emptyset \cdot \ldots$~~ $a_0 \cdots a_k$ and a_{k+1} are rel prime

If not let p be prime, $p \mid a_{k+1}$ & $p \mid a_0 \cdots a_k$

so for some j, $0 \leq j \leq k$, $p \mid a_j$

so for some j, $0 \leq j \leq k$ $p \mid a_j$

But $p \mid a_{k+1}$ Absurd since $\gcd(a_j, a_{j+1})$ are rel prime

By CRT for $n=2$,
Let c be such that

$$\operatorname{rem}(c, a_{k+1}) = x_{k+1}$$
$$\operatorname{rem}(c, a_0 \cdots a_k) = c_1$$
$$0 \le c < a_0 \cdots a_{k+1}$$

Remains to see $\operatorname{rem}(c, a_i) = x_i$ for $0 \le i \le k$

But $c = c_1 + a(a_0 \cdots a_k)$

$$\operatorname{rem}(c_1, a_i) = x_i \quad (0 \le i \le k)$$

so $\operatorname{rem}(c, a_i) = x_i$ (?!)

Lemma:

Let ~~$n \ge$~~ $n \ge 1$

The numbers

$1 + n!$

$1 + 2 \cdot n!$

$1 + 3 \cdot n!$

$\cdots$

$1 + i \cdot n!$

$\cdots$

$1 + (n+1) \cdot n! \qquad = \{1 + k \cdot n! \mid 1 \le k \le n+1\}$

are all relatively prime.

Proof:

Suppose not. Then for some prime p,

$$p \mid 1 + i \cdot (n!)$$
$$p \mid 1 + j \cdot (n!)$$
$$1 \le i < j \le n+1$$

So $p \mid (j-i) \cdot 1 \cdot 2 \cdot \ldots n$

$1 \leq (j-i) \leq n$

So for some $k \leq n$

$p \mid k$

so $p \leq k \leq n$

But $p \mid 1 + i \cdot n!$

Since $p \leq n$, $p \mid n!$ so $p \mid i \cdot n!$

so $p \mid (1 + i n!) - (i \cdot n!) = 1$

so $p \mid i$

contradiction

Dfn:

$\beta(c, d, i) = \text{rem}(c, 1 + (i+1) d)$

Lemma:

Let $a_\emptyset, \ldots, a_n$ be non negative integers

Then $\exists\, c, d$: $\beta(c, d, i) = a_i$ for $0 \leq i \leq n$

Proof:

Let m be very large

$m > n+1$

$m > a_\emptyset$

$\vdots$

$m > a_n$

Let $d = m!$

$1+d, \ldots, 1+md$ are rel prime

$1 + i \cdot d \geq d \geq m > a_{i-1}$ Let c be such that $\text{rem}(c, (i+1)d) = a$

This works!

$\beta(c, d, i) = y \leftrightarrow$

$0 \leq y < (1 + (i+1) \cdot d)$ &

$(\exists z \leq c)[c = z \cdot (1 + (i+1) \cdot d + y]$

This is Δ_0 so num rep in Q

So QED Every recursive fcn is num fcnly rep in Q

Symbols of our language:

$\exists$	0	v_0	8
$\vee$	1	v_1	9
$\neg$	2	v_2	10
~~&~~ +	3	$\vdots$	
$\cdot$	4	v_i	$8+i$
0	5	$\vdots$	
S	6		
=	7		

"Random Fluctuation in my Brain Circuits"

Godel Number formula's and terms

$\langle s_0, \ldots, s_n \rangle$ gets G.#

$\langle\langle \# s_0, \ldots, \# s_n \rangle\rangle$

$\langle 0 \rangle$ gets code $\langle\langle 5 \rangle\rangle = 2^{5+1} = 2^6 = 64$ "Dont blow it Robert"

If $\varphi_1, \ldots, \varphi_n$ is a sequence of formulas

it gets G.#

$\langle\langle \# \varphi_1, \ldots, \# \varphi_n \rangle\rangle$

Last time we assigned Gödel numbers to:
(a) symbols of L_e
(b) all finite sequences of symbols
(c) all finite sequences of finite sequences of symbols

$\#\exists$ is Gödel number of symbol "$\exists$"
x is the Gödel number of a term
x "is a term"
x is a term sin wholeheartedly

Proposition:
The predicate "x is the G# of a term" is prim rec

oof will use course of values recursion
Recall general set up $f : \omega^n \to \omega$
Introduce a fcn $\bar{f} : \omega^n \to \omega$
$\bar{f}(x_1, \dots, x_{n-1}, y)$
$= \bar{f}(\vec{x}, y) = \langle\langle f(\vec{x}, 0), \dots, f(\vec{x}, y-1)\rangle\rangle$
we showed:
if $f(\vec{x}, y) \underset{\#}{=} g(\bar{f}(\vec{x}, y))$ (#) with g primitive recursive,
then f is prim recursive
Recall proof:
Given g, exactly one f satisfying *
$\bar{f}(0) = 1 = \langle\langle\ \rangle\rangle$
$\bar{f}(Sy) = \bar{f}(y) * \langle\langle g(\bar{f}(y))\rangle\rangle \quad \Rightarrow \bar{f}$ is pr

$f(y) = (\bar{f}(y+1))_y \quad \Rightarrow f$ is pr

Fact: $x * y$ $\quad x = \langle\langle a_0, \dots, a_n \rangle\rangle \quad y = \langle\langle b_0, \dots, b_n \rangle\rangle$

$x * y = \langle\langle a_0, \dots, a_n, b_0, \dots, b_n \rangle\rangle$

$x \le x * y$

$y \le x * y$

$a_i < \langle\langle a_0, \dots, a_n \rangle\rangle$

$x = 2^2 = \langle\langle 1 \rangle\rangle$

$y = 2^3 = \langle\langle 2 \rangle\rangle$

$x * y = 2^2 3^3 = \langle\langle 1, 2 \rangle\rangle$

x is the G# of a term

$\Leftrightarrow$ (1) $x = \langle\langle \#0 \rangle\rangle$

or (2) $(\exists i < x) \quad x = \langle\langle 8+i \rangle\rangle$

or (3) $(\exists y \lneq x) \quad x = \langle\langle \#S \rangle\rangle * y$ & "y is a term"

or (4) $(\exists y \lneq x)(\exists z \lneq x) \quad x = \langle\langle \#+ \rangle\rangle * y * z$ & "y is a term" & "z is a term"

or (5) $(\exists y \lneq x)(\exists z \lneq x) \quad x = \langle\langle \#\cdot \rangle\rangle * y * z$ & "y is a term" & "z is a term"

This gives a course of values prim rec of fcn h:

$h(n) = 1$ if n "is a term"

$= 0$ if n "isn't a term"

QED

Proposition:

"x is a formula" (wff) is a primitive recursive predicate

Proof:

Similar to that for "is a term"

x is a formula $\Leftrightarrow$ (1) $(\exists y)(\exists y < x)(\exists z < x)$ (y "is a term") & (z "is a term") &

$x = \langle\langle \# = \rangle\rangle * y * z$

(2) ... (negation) (3) ... (disjunction) (4) $(\exists y < x)(\exists z < x)$ ($x = \langle\langle \#\exists \rangle\rangle * y * z$

& z "is a formula" & $(\exists i < y)(y = \langle\langle 8+i \rangle\rangle)$)

We are going to define:

$Sb : \omega^3 \to \omega$

$\ni$ (1) Sb is p.r.

(2) If α is a term and x is variable (i.e. $x \geq 8$) and t is a term

~~(3)~~ $Sb(\alpha, x, t)$ is $\alpha_x[t]$

α $+xx$

x

t $SS0$

$\alpha_x t$ $+SS0SS0$

(3) If α is a formula, x is a variable, t is a term
$Sb(\alpha, x, t)$ is Godel number of $\alpha_x[t]$
(replacing <u>free</u> occurrences of x by t)

Dfn of Sb:
By course of values recursion on α for fixed x, t)

case 1: $x < 7$ (x "isn't a variable")
or t "is not a term"

$Sb(\alpha, x, t) = \alpha$

case 2: case 1 fails ~~formula~~ but
α is neither a term or a formula

$Sb(\alpha, x, t) = \alpha$

case 3: case 1 fails but α is a term
This splits into 6 subcases

case 3A: $\alpha = \langle\langle 0 \rangle\rangle$

$Sub(\alpha, x, t) = \langle\langle 0 \rangle\rangle$

case 3B: $\alpha = \langle\langle 8+j \rangle\rangle$
but $8+j \neq x$
then $Sb(\alpha, x, t) = \alpha$

case 3C: $\alpha = \langle\langle 8+j \rangle\rangle$
and $8+j = x$

$Sb(\alpha, x, t) = t$

To be Contd

Goal:

$Sb(\alpha, x, t) = * \alpha_x[t]$; Function

Cases 1 & 2:

If $x \leq 7$ or t not a term or α is neither a formula nor a term,

$Sb(\alpha, x, t) = \alpha$

Case 3: α is a term & case 1 fails

case 3.1: $\alpha = \langle\langle * 0 \rangle\rangle$ $Sb(\alpha, x, t) = \alpha$

case 3.2: $(\exists i < \alpha)(\alpha = \langle\langle 8+i \rangle\rangle$ & $8+i \neq x)$

$Sb(\alpha, x, t) = \alpha$

case 3.3: $\alpha = \langle\langle x \rangle\rangle$

Then $Sb(\alpha, x, t) = t$

case 3.4: $(\exists a < \alpha)(\exists b < \alpha)$

a is a term b is a term $\alpha = \langle\langle * + \rangle\rangle * a * b$

Let $a_\emptyset = (\mu a < \alpha)(\exists b < \alpha)(\alpha = \langle\langle * + \rangle\rangle * a * b)$

$b_\emptyset = (\mu b < \alpha)(\alpha = \langle\langle * + \rangle\rangle * a_\emptyset * b)$

$Sb(\alpha, x, t) =$

$\langle\langle * + \rangle\rangle * Sb(a_\emptyset, x, t) * Sb(b_\emptyset, x, t)$

case 3.5: ~~$Sb(\alpha, x, t) = \langle\langle * \cdot \rangle\rangle$~~ α is "$\cdot t_1 t_2$"

Left to you

case 3.6: α is "$S t_1$"

Left to you

Case 4: α is a formula and case 1 fails

Case 4.1 α is "$= t_1 t_2$"

left to you

Left to you

case 4.3 α is "$\mathrm{V}\alpha\beta$"

Left to you

case 4.4 α is "$\exists v \beta$"

case 4.4.A $v = x$ (so there are no free occurrances of x)

$$Sb(\alpha, x, t) = \alpha$$

case 4.4.B $v \neq x$

$$Sb(\alpha, x, t) = \text{"}\exists v\ Sb(\beta, x, t)\text{"}$$

$$\langle\langle \exists \rangle\rangle * \langle\langle v \rangle\rangle * Sb(\beta, x, t)$$

Proposition:

(Conv: t is a term, x is a variable (i.e. $x \geq 8$)), α is a formula)

The ~~formula~~ relation "x occurs in t" is prim recursive

Proof:

$x \in Var$, $t \in Term$ and $Sb(t, x, 0) \neq t$

Proposition:

"x occurs free in α" is prim recursive

$x \in var$, $\alpha \in Form$ $Sb(\alpha, x, 0) \neq \alpha$

Proposition:

"α is a sentence" is prim rec

$(\forall y < \alpha)(y \in var \rightarrow y$ doesnt occur free in $\alpha)$

$Sub(x, t, \alpha)$ means: Predicate

$x \in Var$, $t \in Term$

and t is substitutable for x at free

occurrances of x in α

Proposition:

Sub(x,t,α) is primitive recursive

Proof:

Will give a course-of-values primitive recursive definition for Sub by induction on α

Sub(x,t,α) iff

$x \in$ Var

$t \in$ Term

$\alpha \in$ Form

and case 1 α is atomic $(=t_1 t_2)$ only possibility

YES Sub(x,t,α) is true

or case 2 α is "$\vee \beta \gamma$"

Sub(x,t,α) iff Sub(x,t,β) and Sub(x,t,γ)

or case 3 α is "$\neg\beta$"

left to you

or case 4 α has form "$\exists v \beta$" ~~or x doesn't occur free in β~~

case 4.1 $v = x$ or x doesn't occur free in β

Then Sub(x,t,α) is true

($\because$ no free occurances of x)

case 4.2 case 4.1 fails (ie $v \neq x$ & v occurs in t)

Then some occurance of v in t will be clobbered by outer $\exists v$ in $\alpha_x[t]$

Then $\neg$ Sub(x,t,α)

case 4.3 otherwise

Sub(x,t,α) iff Sub(x,t,β)

$\boxtimes$ QED

Proposition: "ϕ is an identity axiom" is prim recursive

"ϕ is equality axiom" is prim recursive

$\hookrightarrow v_1 = v_2 \wedge v_3 = v_4 \wedge v_1 = v_3 \rightarrow v_2 = v_4$

(for the predicate $=$)

need to do all four cases

"I hame tenure"

"When the winds howl its nice"

$\exists$-intro (rule of inference)

$$\frac{\phi(x) \rightarrow c}{\exists x\, \phi(x) \rightarrow c}$$ provided x doesnt occur free in c

Proposition:

"β follows from α" by an $\exists$ introduction is prim rec.

Proof:

left to you

Proposition:

"α follows from β, γ by MP" is primitive recursive

$\beta = \gamma \rightarrow \alpha$

Proof:

clear

Friday

Logical Axioms

Identity Axioms

Equality Axioms

Substitution Axioms

Tautologys

Logical Rules of Inference

Modus Ponens

$\exists$ introduction

$\varphi_x[t] \to \exists x \phi$

Proposition:

"α is a ~~substitutition~~ axiom" is prim recursive

Proof:

This works:

$(\exists x < \alpha)(\exists t < \alpha)(\exists \phi < \alpha)$ (α has form $\varphi_x[t] \to \exists x \phi$

or

$(\exists \varphi < \alpha)$ (x doesn't occur free in φ and ~~α~~ x is "$\phi \to \exists x \phi$")

To express $\varphi_x[t]$ use $Sb[\varphi, x, t]$

Lemma:

There is a primitive recursive function $h(x,y)$ such that if $s \in Seq$, $lh(s) \le x$ and $(\forall i < lh(s))((s)_i \le y) \Rightarrow s \le h(x,y)$

$$s = \prod_{i < lh(s)} p_i^{(s)_i + 1}$$

$$\le \prod_{i<lh(s)} p_i^{y+1}$$

$$\le \prod_{i<lh(s)} p_x^{y+1}$$

$$\le p_x^{(x)(y+1)}$$

$$= h(x,y)$$

Proposition:

"α is a ~~tautology~~" is primitive recursive

Proof: (later)

Definition:

Let $s \in Seq$ and $(\forall i < lh(s))((s)_i < 2)$

then S ~~respects~~ propositional connectives if

(1) If $\cancel{\#}(\neg\beta) < lh(s) \rightarrow (s)_{\neg\beta} = 1 - (s)_\beta$

(2) If $\vee\beta\gamma < lh(s) \rightarrow (s)_{\vee\beta\gamma} = (s_{\cancel{\beta}})_\beta + (s)_\gamma - (s)_\beta (s)_\gamma$

Proposition:

"α is a tautology" is primitive recursive

Proof:

~~Following~~ means

α is a formula &

$[\forall s \leq h(\alpha+1, 2)]$ [If $s \in Seq$ & $(\forall i < lh(s))((s)_i < 2)$ & s respects prop conn]

then $(s)_\alpha = 1$

$\boxtimes$ QED

Proposition:

"α is an axiom of Q" is primitive recursive

"$\alpha = Q1$" $\vee$ "$\alpha = Q2$" $\vee$ ... "$\alpha = Q7$"

Recall P has family of induction axioms

$[\phi_x[0] \wedge (\forall x)[\phi \rightarrow \phi_x[Sx]]) \rightarrow \forall x \phi$

Proposition:

"α is an induction axiom" is prim recursive

Proof:

$(\exists \phi < \alpha)(\exists x < \alpha)$

so that α has the form

"$\varphi_x[0]$ & $(\forall x)(\phi \to \varphi_x[Sx]) \to \forall x \, \varphi_x$"

This works

Proposition: "α is the ~~Gödel number of~~ a [Q] P-proof of β" ~~proof of~~

is primitive recursive

Proof:

I'll do it for P proof (Case of Q proof is similar)

following works

$P \in Seq$ & $lh(P) > 0$

and $(P)_{lh(P)-1} = \beta$

and $(\forall i\, lh(P))((P_i))$ is a formula

and $(\forall i < lh(P))$

$[(P)_i$ is a logical axiom or

$(P)_i$ is an axiom of P or

$\exists j < i$ s.t. $(P)_i$ follows from $(P)_j$ by Ex. Intro or

$(\exists j < i)(\exists k < i)$

$(P)_i$ follows from $(P)_j$ and $(P)_k$

by modus ponens

Proposition:

If $f: \omega \to \omega$ and f is num wise rep in Q then

f is recursive

Theorem: $f:\omega^n\to\omega$ is num rep in Q
$\Rightarrow f$ is recursive (converse of prev long proof)
(some places use just this as defn)

Proof: will give proof for $n=1$

Lemma: There exists a prim recursive fcn $\mathrm{Num}:\omega\to\omega$
$\mathrm{Num}(n) = {}^\#\underline{n}$

Let $f:\omega\to\omega$
Let $\varphi(v_0,v_1)$ num rep f in Q
Define $h:\omega\to\omega$
$h(n) = (\mu y)($ ⓪ $y\in\mathrm{Seq},\ \mathrm{lh}(y)=3$
① y_0 is a Godel number of a Q proof of $(y)_1$
② $(y)_1 = {}^\#\varphi(\underline{n},\underline{(y)_2})$

Since $Q\vdash\varphi(\underline{n},\underline{f(n)})$
its clear that h is total and recursive

Claim: $(h(n))_2 = f(n)$
Granted claim clear that f is recursive

Proof: (Claim)
If $z = (h(n))_2$
$Q\vdash\varphi(\underline{n},\underline{z})$
If $z\neq f(n)$
$Q\vdash\neg\varphi(\underline{n},\underline{z})$ (since φ num rep in f)
But Q is consistent, so $z=f(n)$
QED (Claim)
QED (Theorem)

T is a consistent theory

L_T is finite (only finitely many non logical symbols)

We can Gödel number L_T much as we did for L_Q

If $\{*\varphi : \varphi$ is an axiom of $T\}$ is recursive,

the relation "π is a T proof of φ" will be recursive

If $\{0, s\}$ are among the symbols of L_T,

then one can define "num rep" as before

$T \vdash \varphi(\underline{x}, \underline{y})$ $(y = f(x))$

$T \vdash \neg\varphi(\underline{x}, \underline{y})$ $(y \neq f(x))$

one can prove just as for Q,

if T as above & T is consistent,

T num rep f, then f is recursive

(Soft Result, in Q richer statements

in fact in ${}_{\mathbb{R}}$RCF not true that rec $\Rightarrow$ num rep (?))

Theorem:

Let T be a theory in a finite language

Fix a Gödel numbering of L_T

Assume $L_Q \subseteq L_T$

$Ax_Q \subseteq Thm_T$

T consistent

Then $\{*\varphi \mid T \vdash \varphi\}$ is not recursive

Definition:

Let T be a theory with L_T finite. Then T is undecidable if $\{*\varphi \mid T \vdash \varphi\}$ is not recursive

Remark: This does not depend on Q (will return to this.

(1) T is consistent

and (2) If S is a theory with $L_S = L_T$, $Ax_S \supseteq Ax_T$

and S is consistent,

then S is undecidable

Result (next goal) is

Q is essentially undecidable

Remark:

Its quite easy to see if

$\#_1$ and $\#_2$ are two Godel numberings for T

in the style of what we did for Q

then there are $f, g : \omega \to \omega$ prim recursive

such that $f(\#_1 \varphi) = \#_2 \varphi$

$g(\#_2 \varphi) = \#_1 \varphi$

Hence the set ~~of~~ $\{\#_1 \varphi \mid T \vdash \varphi\}$ is recursive $\Leftrightarrow$

$\{\#_2 \varphi \mid T \vdash \varphi\}$ is recursive

Lemma: (Diagonal lemma)

Let $P \subseteq \omega^2$

For $b \in \omega$, let $P_b = \{n \mid \langle n, b\rangle \in P\}$

Let $D = \{n \mid \langle n, n\rangle \notin P\}$

Then $D \neq P_b$ any b

Proof: $b \in D \Leftrightarrow \langle b, b\rangle \notin P \Leftrightarrow b \notin P_b$

so $D \neq P_b$

$Thm_T(x)$ iff x is a G# of a thm of T

let $P = \langle a, b\rangle$: $Thm_T(Sb(b, \# v_0, Num(a)))$

Claim: If A is recursive then ...

$$P_b = A$$

Let $\varphi(v_0, v_1)$ num rep χ_A in Q

Let $\psi(v_0)$ be $\varphi(v_0, \underline{1})$

If $n \in A$, $\chi_A(n) = 1 \to Q \vdash \psi(\underline{n}) \to T \vdash \psi(\underline{n})$

If $n \notin A$, $\chi_A(n) \neq 1 \to Q \vdash \neg\psi(\underline{n}) \to T \vdash \neg\psi(\underline{n}) \to T \nvdash \psi(\underline{n})$ (con T)

$A = \{n \mid T \vdash \psi(\underline{n})\}$

Let $b = \# \psi$

Then clearly $A = P_b$

QED[Claim]

Let $Q = \{b \mid \neg\langle b, b\rangle \in P\} =$

$\{b \mid \neg \mathrm{Thm}_T(\mathrm{Sb}(b, \# v_0, \mathrm{Num}(b)))\}$

$Q \neq P_b$, any b

so Q isn't recursive (∵ each recursive set has form P_b)

So must be Thm_T is not recursive

i.e. $\{\# \varphi \mid T \vdash \varphi\}$ is not recursive

Next Goal: Gödel Incompleteness Theorem

$\underline{Q}$ T T consistent $\Rightarrow T$ undecidable $L_T \supseteq L_Q; Ax_T \supseteq Ax_Q$

Definition:

A set $A \subseteq \omega^n$ is recursively enumerable if there is a recursive relation $S \subseteq \omega^{n+1}$ and $\langle x_1, \ldots, x_n \rangle \in A \Leftrightarrow$ $(\exists y \in \omega)(\langle y, x_1, \ldots, x_n \rangle) \in S$

Basic example:

(1) Let f be partial recursive. Then $Dom(f)$ & $Range(f)$ are r.e. $f: \omega \to \omega$

Proof: $f(n) = U(\mu y)T(e, n, y)$ ← any partial rec fn is of this form, where e is G# of f

$Dom(f) = \{x \mid (\exists y)\, T(e, x, y)\}$

$Range(f) = \{x \mid (\exists y)\,[\,T(e, (y)_0, (y)_1) \wedge U((y)_1) = x\,]\}$

converse true (important)

(2) T Theory

L_T Finite

Ax_T is recursive (T is (recursively) axiomatizable)

Then $\{\#\varphi \mid T \vdash \varphi\}$ is r.e.
$= Thm^*_T$

$n \in Thm^*_T \Leftrightarrow (\exists y)[\,y$ is Gödel number of a T proof whose last line has G# $n\,]$

we cannot prove the set of [illegible] of [illegible]
non recursive.

Its not true that any r.e. set is range
of recursive fn
(eg If $A=\emptyset$, certainly not range of total fn)

Proposition:

If A is r.e., $A \subseteq \omega$, and $A \neq \phi$ then
$\exists$ rec $h : \omega \to \omega$ with range $h = A$

Proof:

Let $A = \{x \mid (\exists y)\, S(x,y)\}$

Let $a \in A$

$h(y) = (y)_0$ if $S((y)_0, (y)_1)$

$h(y) = a$ if $\neg S((y)_0, (y)_1)$; to force totality

Then this works (?!)

Closure properties of r.e. sets:

(1) $A \subseteq \omega$, A recursive $\Rightarrow$ A is r.e.

$x \in A \Leftrightarrow (\exists y)\,\underbrace{(x \in A \wedge y = y)}_{\text{recursive predicate}}$; to cater to that vulgar prejudice

(2) If $A, B \subseteq \omega$ are r.e.

So is $A \cup B$, $A \cap B$

$A = \{x \mid (\exists y)\, S_1(x,y)\}$ S_1 recursive
$B = \{x \mid (\exists y)\, S_2(x,y)\}$ S_2 recursive

$A \cup B = \{x \mid (\exists y)(S_1(x,y) \text{ or } S_2(x,y))\}$

$A \cap B = \{x \mid (\exists y)(S_1(x,(y)_1) \text{ and } S_2(x,(y)_2))\}$; pedagogically confusing aesthetic since subscripts

Proposition:

$$K = \{ e \mid (\exists y)\, T(e,e,y) \} \quad \text{is r.e.}$$

Corollary:

There exists an r.e. set which is not recursive

Proposition:

Let $A \subseteq \omega^{n+1}$ be r.e.

Let $B = \{ \langle x_0, \dots, x_{n-1} \rangle \mid (\exists y) \langle x_0, \dots, x_{n-1}, y \rangle \in A \}$

Then B is r.e.

Proof:

$A = \{ \vec{x} \mid (\exists z) S(\vec{x}, z) \}$ S recursive

$B = \{ \langle x_0, \dots, x_{n-1} \rangle \mid (\exists y)(\exists z)\, S(x_0, \dots, x_{n-1}, y, z) \}$

$= \{ \langle x_0, \dots, x_{n-1} \rangle \mid (\exists u)\, S(x_0, \dots, x_{n-1}, (u)_0, (u)_1) \}$

Proposition:

Suppose $A \subseteq \omega^n$, A is r.e.

$h_1, \dots, h_n : \omega^m \longrightarrow \omega$ recursive

Then $\{ \langle x_1, \dots, x_n \rangle : \langle h_1(\vec{x}), \dots, h_n(\vec{x}) \rangle \in A \}$ is r.e.

Follows from proof that $(\exists)$r.e. is r.e.

Proof:

$\langle x_1, \dots, x_n \rangle \in B \Leftrightarrow (\exists y_1 \dots \exists y_n)\ y_1 = h_1(\vec{x}) \wedge \dots y_n = h_n(\vec{x})\ \& \ \langle y_1, \dots, y_n \rangle \in A \}$

Not true:

A r.e. $\Rightarrow \omega - A$ r.e.

$A \subseteq \omega^2$ is r.e. $\{ x \mid (\forall y)(\langle x, y \rangle \in A) \}$ is r.e.

Proposition:

Let $A \subseteq \omega^2$ (works for ω^n, $n \geq 2$) be r.e.

Let $B = \{\langle x,y\rangle \mid (\exists z < y)(\langle x,z\rangle \in A\}$

Let $C = \{\langle x,y\rangle \mid \forall z < y\ (\langle x,z\rangle \in A)\}$

Then B, C are r.e.

$B = \{\langle x, y\rangle \mid (\exists z)[\underbrace{z < y \wedge \langle x, z\rangle \in A}_{\text{r.e.}}]$

r.e. closed under existential quant

$\Rightarrow B$ is r.e.

$A = \{\langle x, z\rangle \mid (\exists u)\, S(x,z,u)\}$ with S rec

$\langle x,y\rangle \in C \Leftrightarrow (\exists w)[(\forall z < y)\, S(x, z, (w)_z)]\}$

(You should think this through)

So clearly C is r.e.

Proposition: Let $A \subseteq \omega$

TFAE

(1) A recursive

(2) A and $\omega - A$ are r.e.

Proof:

(1) $\Rightarrow$ (2) is trivial

A recursive $\Rightarrow A$ is r.e.

A recursive $\Rightarrow \omega - A$ is recursive

$\Rightarrow \omega - A$ is r.e.

Proposition:

Let $A \subseteq \omega$

Then A is recursive $\Leftrightarrow$ A is r.e. and $\omega - A$ is r.e.

Proof:

($\Rightarrow$) Trivial

($\Leftarrow$) Suppose A, $\omega - A$ are r.e.

$A = \{x \mid (\exists y)\, S(x,y)\}$ S recursive

$\omega - A = \{x \mid (\exists y)\, S^*(x,y)\}$ S^* recursive

$(\forall x)(\exists y)\, [S(x,y) \vee S^*(x,y)]$

Define $h : \omega \to \omega$, recursive by

$h(x) = \mu y\, [S(x,y) \vee S^*(x,y)]$

$S(x, h(x)) \to x \in A$

$x \in A \Rightarrow x \notin \omega - A$

$\Rightarrow \neg S^*(x, h(x))$

$\Rightarrow S(x, h(x))$

So $x \in A \Leftrightarrow S(x, h(x))$

$\Rightarrow A$ is recursive

$\square$ QED

Corollary:

$\omega - K$ is not r.e. (K "halting problem" $= \{x \mid (\exists y)\, T(x,x,y)\}$)

Proof: If it were then since K is r.e. K would be recursive. So r.e. sets aren't closed under compl.

$K = \{x \mid (\exists y) T(x,x,y)\}$

$\bar{K} = \{x \mid (\forall y) \neg T(x,x,y)\}$

$W = \{(x,y) \mid \neg T(x,x,y)\}$

W is recursive but $\forall y\, W$ is not r.e. (its W-K)

Back to theories:

Convention:

Until further notice all theories have languages with finitely many logical symbols and are Gödel numbered.

Proposition:

Let T be a complete (also consistent) rec axiomatizable theory. Then T is decidable.

Proof:

Let $\mathrm{Thm}_T = \{\# \varphi \mid \varphi$ is a sentence & $T \vdash \varphi\}$

We know Thm_T is r.e.

Since T is complete (also con) if φ is a sentence of L_T (only for sentences)

$T \vdash \varphi \longleftrightarrow T \nvdash \neg\varphi$

$\omega - \mathrm{Thm}_T = \{n \mid$ Either n is not the G# of a sentence, or $\langle \# \neg \rangle * n \in \mathrm{Thm}_T\}$

So $\omega - \mathrm{Thm}_T$ is r.e. so Thm_T is recursive

Simple Fact:

$h: \omega \to \omega$ is recursive

A is r.e.

$h^{-1}(A) = \{n \mid h(n) \in A\}$ is r.e. $h^{-1}(A) = \{x \mid (\exists y)\, h(x) = y \wedge y \in A\}$ is r.e.

Corollary: RCF (Real Closed Fields) ...
obviously complete (specific structure is $\mathbb{R}$)

Lemma:
There is a prim rec. fn $h : \omega \to \omega$ such that if n is the G# of a formula φ, $h(n)$ is G# of a closure of φ.

Proof:
$h(n)$ will be G#

$\forall v_0 \forall v_1 \ldots \forall v_n \varphi$

Proposition:
TFAE
(1) $Thm_T = \{{}^{\#}\varphi \mid \varphi$ is a sentence & $T \vdash \varphi\}$ is recursive
(2) $thm'_T = \{{}^{\#}\varphi \mid \varphi$ is a formula & $T \vdash \varphi\}$ is recursive

Proof: (2) $\Rightarrow$ (1) Trivial
(1) $\Rightarrow$ (2)
$\varphi \in Thm'_T \Leftrightarrow$ (1) φ is a formula
(2) $h(\varphi) \in Thm_T$

QED

Theorem: (Gödel's First Incompleteness Theorem)
Let T be a theory
suppose (1) T is consistent
(2) T is rec. axiomatizable
(3) Q "$\subseteq$" T i.e. $L_Q \subseteq L_T$
$Q \vdash \varphi \Rightarrow T \vdash \varphi$

Then T is incomplete.

Proof:

By earlier results

(1) & (3) $\Rightarrow$ T is not decidable

(2) & T complete $\Rightarrow$ T is decidable

So if T satisfies (1), T is incomplete

Corollary:

Peano arithmetic is incomplete

Proof:

Need Q "C" P

Only problematical axiom:

$x \neq 0 \rightarrow (\exists y)(x = Sy)$ (Prove by ind on x)

$P \vdash x \neq 0 \rightarrow (\exists y)(x = Sy)$ interesting...

Definition:

Let T_1, T_2 be theories; T_1 a finite extension of T_2 if

(1) $L_{T_1} = L_{T_2}$

(2) $Ax_{T_1} = Ax_{T_2} \cup \{\varphi_1, \dots, \varphi_k\}$

Proposition:

If T_2 is decidable & T_1 is a finite extension of T_2 then T_1 is decidable

Proof: $T_1 = T_2 \cup \{\varphi_1, \dots, \varphi_k\}$

Let φ_i^* be closure of φ_i

Let $\Phi = \varphi_1^* \wedge \dots \wedge \varphi_n^*$

$T_2[\Phi] \vdash \psi \Leftrightarrow T_2 \vdash \Phi \rightarrow \psi$

so if T_2 ~~&~~ is decidable, so is T_1

Corollary

If T_1 is a finite extension of T_2 and T_1 is undecidable, so is T_2.

Example:

$T_1 = Q$

$T_2 = \langle L_Q, \emptyset \rangle$

So $\{ \# \varphi \mid \varphi$ is a logically valid sentence of $L_Q\}$ is not recursive

Example: $(+, \cdot, 0, 1)$

T_2 = Theory of commutative rings

$T_1 = T_2[\forall x (x=0 \vee x=1)]$

Then T_1 is obviously decidable (has only one model up to isomorphism i.e. $\mathbb{F}_2$)

But T_2 is not decidable

Finite extension of an undecidable thy which is decidable (!)

Fact:

If $x \in \omega$ then $\exists y_1, y_2, y_3, y_4 \in \omega$:

$$x = y_1^2 + y_2^2 + y_3^2 + y_4^2 \quad \text{(Lagrange)}$$

$\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$

In $\mathbb{Z}$ we can define

$\omega = \{x \in \mathbb{Z} \mid \exists y_1 \ldots \exists y_4 \quad x = y_1^2 + y_2^2 + y_3^2 + y_4^2\}$

$\therefore$ P is interpretable in $Th(\mathbb{Z}; 0, 1, +, \cdot)$

Z.F. construct integers $0 \approx \{\}$ $Sx \sim x \cup \{x\}$

$+, \cdot$ are complicated. "is an integer" $\sim \ldots$

T_1, T_2 theories

goal:

Define T_1 ~~is~~ is interpretable in T_2

We will assume there is some language L_T with

$L_{T_1} \subseteq L_T$; $L_{T_2} \subseteq L_T$

this means:

(1) Each logical symbol of L_{T_1} is the same logical symbol for L_{T_2} (same symbol for "$\exists$")

(2) Each non logical symbol of $L_{T_1} \cap L_{T_2}$ is of the same type (eg 2ary fn symbol) in both languages

eg (ZF : $\in, =$ cant have terms for $a+b$ since only terms are variables!)

but can do by $\exists f$ on $x+1 \ni f(0) = x+1$ ... $f(y) = z$ etc)

Definition:

A preinterpretation of T_1 into T_2 is a map $\pi : \{\overset{\exists}{\cancel{\Xi}}$, non logical symbols of $T_1\} \to$ Formulas of L_{T_2} such that

(1) $\Pi(\exists)$ is a formula having free at most v_0

and $T_2 \vdash \exists v_0 \Pi(\exists)(v_0)$

(2) If P is an n-ary predicate symbol of L_{T_1} (other than $=$) $\Pi(P)$ is a formula of L_{T_2} having at most $v_0, \dots, v_{n-1}$ free

(3) If f is an n-ary fcn symbol of L_{T_1}, then $\pi(f)$ is a formula having at most $v_0, \dots, v_n$ free such that

$T_2 \vdash (\forall v_1)(\forall v_2) \cdots (\forall v_n)$ if $\left(\Pi(\exists)(v_0) \wedge \cdots \wedge \Pi(\exists)(v_{n-1})\right)$ then

$\exists v_{n+1} [\Pi(\exists)(v_{n+1})$ &

$(\forall v_n)[\Pi(\exists)(v_n) \to (v_n = v_{n+1} \Leftrightarrow \Pi(f)(v_0, \ldots, v_n)\}$

This says: if $v_0, \ldots, v_{n-1}$ are in $\{x \mid \Pi(\exists)(x)\}$,
there is exactly one v_i: $v_n \in \{x \mid \Pi(\exists)(x)\}$
& $\Pi(f)(v_0, \ldots, v_n)$

eg $\varphi_+(a, b, c)$ c is a fn of a, b else if
$2+2 = \psi_1$ $2+2 = \psi_2$ then laws of
logic would lead to a contradiction
$(\psi_1 = \psi_2 \to 1 = 0)$

fn means there is exactly one $c = a+b$

Technicalities:
(1) constant symbols are treated as 0-ary fcns
(2) Could be a problem
v_i might not be substitutable $\Pi(\exists)(v_0)$
way around this relable bound variables
to get an alphabetic variant of
$\Pi(\exists)(v_0)$ in which v_i is substitutable
(missed by Enderton)

Next Goal:

$\mathcal{B}$ a model for T_2

Π a pre interpretation of T_1 in T_2

we want to define an L_{T_1} structure

$\mathcal{B}^{\pi}$

$|\mathcal{B}^{\pi}| = \{x \in |\mathcal{B}| \mid \mathcal{B} \models \Pi(\forall)(\underline{x})\}$

Notice $|B^{\pi}| \neq \emptyset$ since $T_2 \vdash \exists v_0 \Pi(\forall)(v_0)$

$=^{B^{\pi}}$ is what it must be

Let P be an nary predicate of L_{T_1}.

$P^{B^{\pi}} = \{\langle x_0, \ldots, x_{n-1}\rangle \in |B^{\pi}|^n : B \models \pi(P)(\underline{x_0}, \ldots, \underline{x_{n-1}})\}$

If f is an nary function symbol of L_{T_1}

$f^{B^{\pi}} = \{\langle\langle x_0, \ldots, x_{n-1}\rangle, x_n\rangle$: ① $x_0, \ldots, x_n \in |B^{\pi}|$ &
② $B \models \pi(f)(\underline{x_0}, \ldots, \underline{x_n})\}$

("The long defn we gave before:")

By our assumption on $\pi(f)$

and $B \models T_2$,

$f^{B^{\pi}} : |B_{\pi}|^n$

Definition:

Let Π be as a tove

π is an interpretation of T_1 in T_2

if for every $B \models T_2$

$B^{\pi} \models T_1$

example: $T_2 = \mathrm{Th}(\langle \mathbb{Z}, +, \cdot, \rangle)$ ...

$T_1 = \mathrm{Th}(\langle \omega, S, 0, < \rangle)$

$\pi(\mathbb{N})(v_0)$ is $\exists v_1 \exists v_2 \exists v_3 \exists v_4 (v_0 = v_1 \cdot v_1 + v_2 \cdot v_2 + v_3 \cdot v_3 + v_4 \cdot v_4)$

$\oplus$

~~$\pi(\mathbb{N})(0) \wedge \pi(v_0)$~~

$\pi(0)(v_0)$ $\quad v_0 = v_0 + v_0$

If $\langle \mathbb{Z}, +, \cdot \rangle \models$ ~~...~~ $\underline{x} = \underline{x} + \underline{x}$, then $\underline{x} = 0$

Definition of Preinterpretation:

(d) To each constant symbol c of L_T, a formula $\pi(c)$ having only v_0 free

$T_2 \vdash \exists! v_0\, \pi(c)(v_0)$

~~$\pi(\mathbb{N})(\exists w)$~~ $(*)$ ~~$\exists w \forall v_0 (\pi(c)(v_0) \leftrightarrow v_0 = w)$~~

(in case not clear: that constants are 0 place function symbols)

$(*)$ $(\exists w)(\pi(\mathbb{N})(w) \wedge (\forall v_0)\, \pi(\mathbb{N})(v_0) \rightarrow [v_0 = w \leftrightarrow \pi(c)(v_0)]$

$\pi(S)(v_0, v_1)$ $\quad (\exists v_2)(v_2 = v_2 \cdot v_2 \wedge v_2 \neq v_2 + v_2 \wedge v_1 = v_0 + v_2)$

$\pi(<)(v_0, v_1)$ $\quad (\exists v_2)(v_1 = v_0 + v_2 \wedge v_2 \neq v_2 + v_2 \wedge \pi(\mathbb{N})(v_2))$

$\oplus$ Not quite evident that this $\pi \rightarrow$ gives an interpretation.

Theorem:

There is a recursive procedure which assigns to each formula ϕ of L_{T_1} a formula ϕ^* of L_{T_2} $\ni$ φ and φ^* have same free variables

if $x_1, \ldots, x_n \in |B^{\pi}|$

$$B^{\pi} \models \varphi(\underline{x_1}, \ldots, \underline{x_n}) \Leftrightarrow B \models \varphi^*(\underline{x_1}, \ldots, \underline{x_n})$$

Theorem:

Let Π be a preinterpretation of T_1 in T_2

Then there is a recursive procedure:

$\varphi \to \varphi^*$ such that:

(1) $F.V.(\varphi^*) \subseteq F.V.(\varphi)$ Free variables

$\subseteq \{v_0, \dots, v_{n-1}\}$

(2) If $B \models T_2$, B^π is the L_{T_1} structure determined by Π, φ is a formula of LT_1

$\varphi(v_0, \dots, v_{n-1})$ (Free Variables of φ are in $\{v_0, \dots, v_{n-1}\}$

and $x_0, \dots, x_{n-1}$ are elts of B^π

$B^\pi \models \varphi(\underline{x}_0, \dots, \underline{x}_{n-1}) \iff B \models \varphi^*(\underline{x}_1, \dots, \underline{x}_{n-1})$

Important special case:

φ is a sentence

so φ^* is a sentence and $B^* \models \varphi \iff B \models \varphi^*$

First consider special case:

T_1 has no function

We will be running into following -

$\Theta(v_0, \dots, v_{n-1})$ $v_{i_0}, \dots, v_{i_{n-1}}$ are another

lis of n variables (possible repetitions)

$\Theta(v_{i_0}, \dots, v_{i_{n-1}})$

obvious try

$\Theta_{v_0, \dots, v_{n-1}}[v_{i_0}, \dots, v_{i_{n-1}}]$

This may not work [illegible]

bound variables.

I'll define

$\theta(v_{i_0}, \ldots, v_{i_{n-1}})$

as follows

Let m be least integer $> i_0, \ldots, i_{n-1}, n-1$

Let $\theta'(v_0, \ldots, v_{n-1})$ be obtained from θ by replacing bound occurrences of v_j by v_{m+j}

$\theta'_{v_0, \ldots, v_{n-1}}[v_{i_0}, \ldots, v_{i_{n-1}}]$ will be $\theta(v_{i_0}, \ldots, v_{i_{n-1}})_{**}$

This procedure clearly recursive!

Definition of φ^*:

case 1 φ is atomic

φ is $P v_{i_1} \ldots v_{i_n}$

φ^* is $\Pi(P)(v_{i_1}, \ldots, v_{i_n})$

case 2 φ is $\psi \cup \chi$

φ^* is $\psi^* \cup \chi^*$

case 3 φ is $\neg \chi$

φ^* is $\neg \chi^*$

case 4 φ is $(\exists x)\psi$

φ^* is $(\exists x)[\Pi(\exists)(x) \wedge \psi^*]$

QED Case 1 (no fn symbols)

f, g function symbols

"Become a member of the Club" Alg Complexity Theory

$f\, g\, x = y$

$(\exists z)(\exists w)(g\,x = z \wedge \cancel{f(x,z)}\ f\,z = w \wedge w = y)$

$\ldots\ \Pi(g)(x,z) \wedge \Pi(f)(z,w) \wedge \cdots$

Basically eliminate terms one by one

Definition:
A formula is special if all atomic subformulas are either
(1) without any fcn symbols
or (2) of form $f v_{i_1} \ldots v_{i_n} = v_{i_{n+1}}$

<u>Case 2</u> T_1 has function symbols

Proposition: There is a recursive procedure: $\varphi \to \varphi^*$:
if φ is an atomic formula of LT_1,
φ^* is an equivalent special formula

If φ is atomic
let rank(φ) be number of function symbols in φ

Definition of φ^*
(0) φ not atomic formula : GARBAGE $\varphi^* = \varphi$
(1) rank(φ) = 0, φ^* is φ
(2) φ has form $f v_{i_0} \ldots v_{i_n} = v_{i_{n+1}}$, $\varphi^* = \varphi$
(3) otherwise
(so φ contains some function symbol)

Let q be rightmost fn symbol...

q starts a term

$q\, v_{i_1}, \ldots v_{i_r}$

Let v_m be least variable not appearing in φ

Let φ' be obtained from φ by replacing (starting ~~...~~ with our given occur of q)

$q\, v_{i_1} \ldots v_{i_r}$

by v_m

φ^* is $(\exists v_m)(q\, v_{i_1} \ldots v_{i_r} = v_m \;\&\; (\varphi')^*)$

This works

Define φ^*

Case 1 ~~...~~ φ is special

case 1A φ atomic

φ has one of forms

$P\, v_{i_1} \ldots v_{i_n}$

in which case φ^* is $\Pi(P)(v_{i_1} \ldots v_{i_n})$

or φ of form

$q\, v_{i_1} \ldots v_{i_n} = v_{i_{n+1}}$

φ^* $\Pi(q)(v_{i_1}, \ldots, v_{i_n}, v_{i_{n+1}})$

case 1B φ is $\psi \vee \chi$

case 1C φ is $\neg\psi$

case 1d φ is $\exists x\, \psi$

General case

If φ is special already done

If φ is atomic

$$\varphi^* = (\varphi^*)^*$$

case 2B $\quad \varphi$ is $\psi \vee \chi$

case 2C $\quad \varphi$ is $\neg \psi$

case 2D $\quad \varphi$ is $(\exists x)\,\psi$

As before.

Π last time

pre interpretation of $Th(\langle \omega; S, 0, < \rangle)$ in $Th(\langle \mathbb{Z}; +; \circ \rangle)$

Claim: Π is an interpretation

Need to see $\mathcal{B} \models T_2 \rightarrow \mathcal{B}^{\Pi} \models T_1$

Let $T_1 \vdash \theta$,

θ a sentence

To see

$\mathcal{B}^{\pi} \models \theta$

$\mathcal{B} = \langle \mathbb{Z}; +, \circ \rangle$

$\eta = \langle \omega, 0, S, < \rangle$

$\mathcal{B}^{\pi} \models \theta \Leftrightarrow \mathcal{B} \models \theta^*$

But $T_1 \vdash \theta \Leftrightarrow \eta \models \theta \Leftrightarrow \mathcal{B} \models \theta^* \Leftrightarrow T_2 \vdash \theta^*$

$\Rightarrow B \models \theta^*$

So $\mathcal{B}^{\pi} \models \theta$ $\cancel{\Leftrightarrow \mathcal{B} \models \theta^*}$ as desired!

Cor Π is an interpretation

Cor π is an interpretation iff

$\forall \theta \quad T_1 \vdash \theta,$ $\quad\Big|\quad$ $\forall$ axioms θ of T_1

$T_2 \vdash \theta^*$ $\quad\Big|\quad$ $T_2 \vdash (\theta^{\text{clos}})^*$

Final version of 1^{st} incompleteness theorem:

Let T be a theory (L_T finite)

Suppose (1) T is recursively axiomatizable
(2) Q is interpretable in T

Then T is incomplete

Proof: If T is not consistent T is not complete

So we may assume T consistent

If T is complete & rec axiomatizable

T is decidable

So enough to show T is undecidable

Let π be the given interpretation of Q in T

Let $\varphi \to \varphi^*$ be the recursive transition map of sentences of L_Q into sentences of L_T, constructed last time

Define a theory T^* as follows:

$L_{T^*} = L_Q$

$Ax_{T^*} = \{\varphi \mid \varphi$ is a sentence of L_Q & $T \vdash \varphi^*\}$

$\varphi \in AxT^* \Leftrightarrow B^\pi \models \varphi$
for all $B \models T$

From this, following are clear

(1) T^* is consistent

(2) If $T^* \vdash \theta$, θ is a sentence, $\theta \in AxT^*$

(3) $Q \subseteq$ ...

But Q is essentially undecidable

So Thm T^* is not recursive

But by ② Thm $T^* = \{ \phi \mid T \vdash \phi^* \}$

If T were decidable, T^* would be decidable

Upshot: T is not decidable

(Trivial as a proof. The hard one was incompleteness of extensions of Q)

Very useful (compare with set theory, etc ...)

Gödel numbering of infinite languages:

L_T a language with at most countably many symbols

Pathology (1) Even w/ countable lang may not be able to give ~~...~~ G.\#

(2) W/ countable lang can have 2 G\# maps but the map between them is non rec

~~OR would~~ results is essentially ...

Enough to Gödel Number the alphabet, Σ_T, of L_T

Definition:

A Gödel numbering of Σ_T is a map $\varphi : A \to \Sigma_T$ such that

(1) A is a recursive subset of ~~...~~ ω

(2) φ is 1-1 and onto Σ_T

(3) Map $i \to \# v_i$ is recursive & has recursive range

(4) There is a recursive function h

$\ni$ if $n \in A$ then $h(n) = 2k$ $k > 0$

iff $\varphi(n)$ is a k-ary predicate symbol

if $n \in \bar{A}$ then $h(n) = 2k+1$

iff $\varphi(n)$ is an k-ary fcn symbol

One can give unnatural examples of countable languages w/ no recursive Gödel numbering

Definition:

Two Gödel numberings

$\varphi : A \to \Sigma_T$

$\psi : B \to \Sigma_T$

are equivalent if $\psi^{-1} \circ \varphi$ are partially equivalent
and $\varphi^{-1} \circ \psi$

An equivalence class of Gödel *'ings is a recursive structure on L_T

There are easy examples of L_T's without any recursive structure

(He said something about non recursive permutations and how you get different G*ings!)

One frequently sees in the literature, dialects of Peano Arithmetic w/ a fcn symbol for each primitive recursive function

Back to convention:

All languages have finitely many non logical symbols.

Theorem:

(1) Let T_1 be finitely axiomatizable and essentially undecidable.

(2) Let T_2, T_3 be theories $L_{T_2} = L_{T_3}$ $(T_2 \subseteq T_3)$

$\text{Thm } T_2 \subseteq \text{Thm } T_3$

T_3 consistent

and T_1 is interpretable in T_3

Conclusion:

T_2 is undecidable theory

Corollary:

There is T_4 with $T_4 \subseteq T_3$ & T_4 is finitely axiomatizable & essentially undecidable.

Theorem:

T_1, T_2 theories Assume

(1) T_1 is finitely axiomatizable and essentially undecidable

(2) T_1 is interpretable in some ~~finite~~ consistent extension, T_3 of T_2 with $L_{T_2} = L_{T_3}$

Conclusion: T_2 is undecidable

Claim: There is a finite set Σ of sentences of L_{T_2} such that if

$L_T = L_{T_2}$ and $\Sigma \subseteq \mathrm{Thm}_T$,

then T_1 is interpretable in T (via same pre interp as used in ~~statement~~ of thm)

Proof: (Claim)

For an interp Π of T_1 in T_3 and let $\varphi \to \varphi^*$ be corresponding translation of formulas of L_{T_1} into L_{T_3}.

What do we need in Σ:

(1) Sentence: $\exists x \, \Pi(\exists)(x)$

(2) For each fn symbol f of L_{T_1}, I need a sentence φ_f that asserts $\Pi(f)$ does determine a fcn (φ_f was explicitly described when we defined pre interpretations)

(3) If θ is an axiom of T_1 $(\theta^{cl})^* \in \Sigma$ ← translation procedure

(θ^{cl} is the closure of θ)

This does it!

Notice:

$T_2[\Sigma] \subseteq T_3$

so $T_2[\Sigma]$ is consistent

By an argument already given,

T_1 essentially undecidable,

T_1 interperable in $T_2[\Sigma]$

$T_2[\Sigma]$ consis

$\to T_2[\Sigma]$ is undecidable

Recall the proof:

Let $L_S = L_{T_1}$

$Ax_S = \{\varphi \mid \varphi$ is a sentence of L_S and $T_2[\Sigma] \vdash \varphi^*\}$

Then S is undecidable, since $S \supseteq T_1$, T_1 essentially undecid

but a decision procedure for $T_2[\Sigma]$ would yield one for S

Let Φ be a conjugation of all axioms in Σ

$T_2[\Sigma] \vdash \varphi \iff T_2 \vdash \Phi \to \varphi$

So since $T_2[\Sigma]$ is undecidable, so is T_2

Application 1

$T_1 = Q$ (finitely axiomatizable, essentially undecidable)

T_2 $\not{\#}$: $L_{T_2} = \{\in\}$

T_3 : ZFC $-$ {power set axiom}

Remark: one can prove in ZFC that ZFC $-$ {power set ax}

is consistent.

Model is $\langle HC; \in \rangle$

where HC is those sets $x \ni$ /* hereditarily countable
* sets
*/

(1) X is countable
(2) members of x are countable
(3) members of members of x are countable
etc

Hence $\{\varphi \mid \varphi$ is logically valid sentence of $L_\in\}$

is not recursive.

Easy to see: If Σ is as in proof,
Σ is essentially undecidable
If T is a consistent extension of Σ,
T_1 is interp in T so
T is undecidable

Gossip:
The following baby set theory is essentially undecidable
(1) Axiom of extensionality
$(\forall z)(z \in x \leftrightarrow z \in y) \to x = y$
(2) Null set
$(\exists x)(\forall y)(y \notin x)$

(3) $a \cup \{b\}$ exists
$(\forall a)$ $(\exists c)(\forall d)(d \in c \Leftrightarrow (d \in a \vee d = b))$
Interp Q in this hard

Applications 2

$T_1 = Q$

$T_2 =$ Thy of comm Rings
$(+, \cdot)$ (can allow 0, 1 still works)

$T_3 = $ Thy $(\langle \mathbb{Z}; +, \cdot \rangle)$

I've already sketched an interp of T_1 in T_3
(4 squares Lagrange)
eg $\omega = \{ x \in \mathbb{Z} \mid \exists y_1 \ldots y_4 \ x = y_1^2 + \cdots y_4^2 \}$ etc

Conclusion: Thy of comm Rings is undecidable.

Not essentially undecidable
$\because$ there are finite commutative rings
$(\langle \mathbb{Z}_2; +; \cdot \rangle)$

Deep Theorem of Ax
~~The set of~~ $\{ \varphi \mid \varphi$ holds in all finite fields $\langle F; +, \cdot \rangle \}$
is recursive.

Notation:
$\mathbb{Q} =$ rationals
$Q =$ our finitely axiomatizable, essentially undecidable

Deep theorem of Julia Robinson:
$\{ n \mid n \in \omega \} = \{ n \mid \langle \mathbb{Q}; + \cdot \rangle \models \varphi(\underline{n}) \}$ for some formula φ

Application:
$T_1 = Q$
$T_2 =$ Thy of fields
$T_3 =$ Thy $(\langle \mathbb{Q}; +, \cdot \rangle)$

By Robinson's theorem, T_1 is interp in T_3 hence T_2 is undecidable

+ Theorem:

Let L be a 1^{st} order language w/ a single binary relation. Then $\{\varphi \mid \varphi$ is a valid sentence of $L\}$ is not recursive

Proof:

We will use same basic approach as last time

T_1 : Q

T_2 : $L_{T_2} = L_\epsilon$ $AxT_2 = \emptyset$

T_3 : $Thy(\langle E; R \rangle)$

where E is a certain set

R is "$\epsilon \restriction E$" "ϵ restricted to E"

$R = \{\langle a,b \rangle : a \epsilon E$ and $b \epsilon E$ and $a \epsilon b\}$

We start by defining V_n for $n \in \omega$

$V_0 = \{\}$

$V_{n+1} = P(V_n) = \{a \mid a \subseteq V_n\}$

$V_\omega = \bigcup_{n<\omega} V_n$ (= hereditarily finite sets)

$E = P(V_\omega)$ $(= V_{\omega+1})$

Proposition: $i<j \rightarrow V_i \subseteq V_j$

Proof: By induction on i

$V_0 = \emptyset \subseteq V_j$ any j

if $i>0$ $i<j$ $V_{i-1} \subseteq V_{j-1}$ (ind hyp) So $P(V_{i-1}) \subseteq P(V_{j-1})$

i.e. $V_i \subseteq V_j$

~~Proposition~~: $V_\omega \subseteq E$

Proof: If $x \in \cancel{V_\omega}\, V_\omega$, $x \in V_n$ for some $n > 0$

But $V_{n-1} \subseteq V_\omega$ ~~so $V_n \in E$~~

$P(V_{n-1}) \subseteq P(V_\omega)$

ie $V_n \subseteq E$

so $x \in E$

Proposition:

Let $a, b \in V_\omega$. Then $a \cup b$, $\{a,b\}$, $\langle a,b \rangle \in V_\omega$

Proof:

If $a, b \in V_\omega$ $a, b \in V_n$ some large n

so $b, a \in P(V_{n-1})$

$a \subseteq V_{n-1}$

$b \subseteq V_{n-1}$

so $a \cup b \subseteq V_{n-1}$

$\Rightarrow a \cup b \in V_n \subseteq V_\omega$

$a, b \in V_\omega \Rightarrow a, b \in V_n$ some large n

$\{a,b\} \in V_{n+1} = P(V_n)$

$\langle a,b \rangle = \{\{a\}, \{a,b\}\}$

so $a, b \in V_\omega \Rightarrow$ ~~$\{a\}$~~ $\{a, a\}$

$\{a,b\} \in V_\omega$

$\Rightarrow \langle a,b \rangle \in$ ~~V_n~~ V_ω

$= 1024 \times 64$

6136

440

65576

Proposition:

If $a \in V_\omega$ then $a \subseteq V_\omega$

Proof:

$a \in V_\omega \Rightarrow a \in V_n$ some $n > 0$

$\Rightarrow a \subseteq V_{n-1} \subseteq V_\omega$

so $a \subseteq V_\omega$

Definition:

We define Δ_n for $n \in \omega$ as follows

$\Delta_0 = \emptyset$

$\Delta_{n+1} = \Delta_n \cup \{\Delta_n\}$

Proposition: $\Delta_n \in V_\omega$ all n (∵ of closure properties of V_ω)

Proposition: $\Delta_n = \{\Delta_i \mid i < n, i \in \omega\}$

Proof: By induction on n

$\Delta_0 = \{\Delta_i \mid i < 0, i \in \omega\}$

$\Delta_{n+1} = \Delta_n \cup \{\Delta_n\} = \{\Delta_i \mid i < n\} \cup \{\Delta_n\}$

$= \{\Delta_i \mid i \leq n\}$

$= \{\Delta_i \mid i < n+1\}$

□ QED

Proposition: $i \neq j \rightarrow \Delta_i \neq \Delta_j$ $(i, j \in \omega)$

WLOG $i < j$

Pick i least such that for some $j > i$, the proposition fails.

Since $i<j$

But $\Delta_i = \Delta_j$,

so $\Delta_i \in \Delta_i$

so $\Delta_i \in \{\Delta_k \mid k<i\}$

$\therefore \exists k < i : \Delta_k = \Delta_i$

This contradicts leastness of i.

QED

Now going to copy $\langle \omega; +, \cdot, S, 0\rangle$ onto

$\langle \bar{\omega}; \bar{+}, \bar{\cdot}, \bar{S}, \bar{0}\rangle$

$\bar{\omega} = \{\Delta_i \mid i \in \omega\}$

Now going to define an interpretation Π of L_a into L_ϵ

so that $\langle E; R\rangle^{\Pi} = \langle \bar{\omega}; \bar{+}, \bar{\cdot}, \bar{S}, \bar{0}\rangle$

This will complete the proof

$\Pi(0)(x)$ $\quad (\forall y)(y \notin x)$ $\quad$ expresses $x=0$

so in $\langle E; R\rangle$, $\Pi(0)(\underline{x})$ iff $x=\emptyset$ $\quad$ i.e. $\langle E;R\rangle \models \Pi(0)(\underline{x})$ iff $x=\emptyset$

$\Pi(S)(x,y)$: $(\forall z)(z \in y \Leftrightarrow (z \in x \text{ or } z = x))$

$\Pi(\exists)(x)$:

$(\forall y)([\,y$ "$0 \in y$" and $(\forall z)(z \in y \to Sz \in y)] \to x \in y)$

"just because of the meanness of my character"

Clear $\{x \mid \langle E, R\rangle \models \Pi(\exists)(\underline{x})\} = \{\Delta_n \mid n \in \omega\}$

$\Pi(+)(a,b,c)$

$(\exists f)$ (① f is a function (all. for $\forall a \in f$ (a is an ordered pair $\forall x,y,z$ ($\langle x,y\rangle \in f$ & $\langle x,z\rangle \in f$ $\rightarrow y=z$.

⋮

② $(\forall x)(x \in \mathrm{dom}(f) \Leftrightarrow$

$\exists y\, \exists z : \langle y, z\rangle$ &

$\Pi(\exists)(y)$ & $\Pi(\exists)(z)$

(i.e. domain$(f) = \bar{\omega}^2$)

③ $(\forall x)\ \Pi(\exists)(x) \longrightarrow f(\langle x, 0\rangle) = x$

④ $\forall x\, \forall y\ (\Pi(\exists)(x) \wedge \Pi(\exists)(y) \rightarrow f(\langle x, Sy\rangle) = S f(x,y)$)

⑤ $f(\langle a,b\rangle) = c$

$\Pi(\cdot)(a,b,c)$:

$(\exists f)(\exists g)$

① f, g are fcns

② $\mathrm{Dom}(f) = \mathrm{Dom}(g) = \bar{\omega}^2$

③ f satisfies recursion equations for $+$

④ $(\forall x)(\Pi(\exists)(x) \rightarrow g(\langle x, 0\rangle) = 0$

⑤ $(\forall x)(\forall y)(\Pi(\exists)(x) \wedge \Pi(\exists)(y) \rightarrow$
$g(x, Sy) = f(g(x,y), x)$

⑥ $g(\langle a,b\rangle) = c$

∴ non recursiveness of set of validities follows

Our "first principles" proof of undecidability of validities ($\vdash_E$) is complete

Friday

ZFC + (¬Con ZFC) is consistent

∴ ↳ has models

∴ has integers

but they can't be isomorphic to regular integers!!!

← one of these is GN of a proof (which it thinks is finite but is really infinite)

good models | bad models ω

Precise Statement of Gödel's Second Incompleteness Theorem:

Let T be a theory such that:

(1) T has a (primitive) recursive set of axioms — still true w/o but need some technicalities

(2) P is interpretable in T

(3) T is consistent

Then T can't prove "T is consistent"

We are going to construct a certain sentence Con(T) of L_P

"T is consistent" will be $[\mathrm{Con}(T)]^*$ (translation under given interpretation of P in T)

Remarks:

(1) Can weaken primitive recursive set of axioms all the way to r.e. set of axioms (But if they is consistent (?) can get rec set from r.e.?)

(2) Can replace P by tiny fragments of P

$L_{Exp} = \{0, 1, S, +, \cdot, E\}$

Ax_{EXP} : Q + Induction for Δ_0 formulas + recursion equations + $x E 0 = S0$

$x E Sy = (x E y)\cdot x$

Fact (non trivial) Finitely Axiomatizable!!

(1) construct a primitive recursive predicate

$Prf_T(x,y)$ (in L_P)

which expresses "x is the G# of a T proof"
and "y is G# of last line of proof"

(as we did for P.A.) Peano Arithmetic $\equiv P$)

(2) Let F be $\exists x (x \neq x)$

Con(T) is $\forall x \neg Prf_P(x, \underline{\#F})$

$T \nvdash [Con(T)]^{\#}$ G. C. Thm 2

Epistles of Paul: Joe the Cretin says all Cretins are liars

Can construct Φ : $Q \vdash \Phi \leftrightarrow \varphi(\underline{e})$

$e = \# \Phi$

Monday

r.e. $A \subseteq \omega$ is r.e.

Showing how to replace prim rec set of axioms by rec enum

printin $\overset{\text{rec}}{\wedge}$ pred

$B(x,y)$

$A = \{x \mid (\exists y) B(x,y)\}$

φ is a sentence

$\varphi^k = \varphi \wedge (\varphi \wedge \ldots \varphi)$ k times

$\varphi^1 = \varphi$
$\varphi^{k+1} = \varphi \wedge \varphi^k$

A is our r.e. set of axioms

$\{\varphi^y : \varphi = x \ \& \ B(x,y)\}$

$= \{\varphi \mid (\exists x \leq \varphi)(\exists y \leq \varphi) \ B(x,y) \ \& \ \varphi = [\varphi_x]^y$

Self Referential lemma:

Theorem:

Let $\varphi(v_0)$ be a formula of L_P
Then there is a sentence Φ with G#e:

$Q \vdash \Phi \Leftrightarrow \varphi(\underline{e})$

("φ hold of me")

Going to construct a series of sentences θ_n which will have short nicknames

Case 1 n is a Gödel Number of a formula $\varphi(v_0)$ having at most v_0 free

Then $\theta_n = \varphi(\underline{n})$

case 2 (garbage) o.w. θ_n is "$0=0$" To make it prim rec

Lemma

The map $n \to \# \Theta_n$ is primitive recursive

Proof:

Basically char. use: $n \to \# \underline{n}$ is primitive recursive

substitution is prim. rec. etc

Let $\alpha(v_0, v_1)$ numeralwise functionally represent the function

$n \to \# \Theta_n$

Let $\psi(v_0)$ be $(\exists v_j)[\alpha(v_0, v_j) \wedge \varphi(v_j)]$

Let r be Gödel number of ψ

$\Phi = \Theta_r = \psi(\underline{r})$ $\quad Q \vdash \alpha(\underline{r}, \underline{e})$ (numeralwise fully represents)

$e = \# \Phi = \# \Theta_r$

1) so $Q \vdash (\forall v_j)[\alpha(\underline{r}, v_j) \leftrightarrow v_j = \underline{e}]$ (defn of numeralwise fnc rep.)

② $\Phi : (\exists v_j)[\alpha(\underline{r}, v_j) \wedge \varphi(v_j)]$

Self Reference:
"Goes back to Gödel or God whichever way you look @ it"

From (1) & (2), by first order logic,

$Q \vdash \Phi \leftrightarrow \varphi(\underline{e})$

[QED]

Application 1:

Let $\eta = \langle \omega; 0; S, +, \cdot \rangle$ standard model

Let $Tr = \{e \mid e$ is G# of a formula φ:

$\eta \models \varphi\}$

Let $\varphi(v_0)$ be a formula of L_P w/ only v_0 free

$Tr \neq \{n \mid \mathfrak{N} \models \varphi(\underline{n})\}$

("Truth is undefinable")

careful careful careful

[Idea of proof:
Deny and go for contradiction
Use a sentence which says
"I am untrue"]

By self reference lemma,
there is a sentence Ψ with $G\Psi = e$
$Q \vdash \Psi \leftrightarrow \neg\varphi(\underline{e})$

$\mathfrak{N} \models \Psi \leftrightarrow \neg\Psi(\underline{e})$

So $e \in Tr \Leftrightarrow \mathfrak{N} \models \Psi \Leftrightarrow (\mathfrak{N} \models \neg\Psi(\underline{e})) \leftrightarrow \neg(\mathfrak{N} \models \Psi(\underline{e}))$

Defn: A theory T ($L_T = L_P$) is ω-inconsistent if for some formula $\varphi(v_0)$ of L_P having only v_0 free

$T \vdash \exists v_0\, \varphi(v_0)$

$T \vdash \neg\varphi(\underline{0})$

$T \vdash \neg\varphi(\underline{1})$,

i.e. $(\forall n \in \omega)\ T \vdash \neg\varphi(\underline{n})$

(not inconsistent but not modelled by $\mathfrak{N}$)

Defn: T is ω consistent if it is not ω-inconsistent

Remark: If T is inconsistent $T \vdash \alpha$ any α. so T is ω-inconsistent. So T is ω-consistent $\Rightarrow T$ is consistent

Example: If T is $P + \neg\mathrm{Con}\,P$

then T is consistent

(since by Gödel $P \nvdash \mathrm{Con}(P)$)

But T is $\underline{\underline{\omega}}$inconsistent

$$\varphi(x) = \mathrm{Prf}_{T_P}(x, \ulcorner(\exists y)(y \neq y)\urcorner)$$

Theorem (Gödel) → can do w/ Q also!

(1) set of $T \supseteq P$, (2) T has a prim rec set of axioms

(3) T is ω consistent

then T is incomplete

(Self Referential Lemma:

Given $\varphi(v_0)$ in L_P. There is a sentence Φ of L_P with G.# e s.t.

$Q \vdash \Phi \leftrightarrow \varphi(\underline{e})$

Excercise:

Given T,

P interpretable in T. Formulate and prove a version of self ref lemma for L_T. ie given $\varphi(v_0)$ in L_T, $\exists$ sentence Φ of L_T $\ni$

$T \vdash \Phi \leftrightarrow$ "φ holds of #Φ"

(1) T has a primitive recursive axiomization

(2) P is interpreted in T

By self referential lemma, get a sentence Φ of L_P which asserts:

"My translation is not a theorem of T"

More precisely,

let Trans(x,y) numerically represent the function (prim rec) which maps G.# of a sentence φ of L_P to G# of its translation φ^*

$Prf_T(x,y)$ numerically represent x is G# of a proof whose last line is y.

$\varphi(x)$: $(\forall y)(\forall z)\,[\mathrm{Trans}(x,y) \rightarrow \neg \mathrm{Prf}_T(z,y)]$

Let Φ be a sentence w/ G# e

$\exists\; Q \vdash \Phi \leftrightarrow \varphi(\underline{e})$ a theorem whose existence is guaranteed by self ref lemma

Gödel has nothing to do w/ "provabl" any property will do

If e_1 is G# of Φ^*

$Q \vdash \mathrm{Trans}(\underline{e},y) \leftrightarrow y = \underline{e_1}$

So $Q \vdash \Phi \leftrightarrow (\forall z) \neg \mathrm{Prf}_T(z, \underline{e_1})$

Claim 1: If T is consistent, $T \nvdash \Phi^*$ ← translation (Φ^*)

Claim 2: If T is ω consistent, then $T \nvdash \neg\Phi^*$

idea:
If Φ^* were provable, that's a certain fact ...

suppose T proves Φ^*
let proof have G# f
so since $\mathrm{Prf}_T(x,y)$ num rep proof predicate

$Q \vdash \mathrm{Prf}_T(\underline{f}, \underline{e_1})$

$Q \vdash (\exists x)\,\mathrm{Prf}_T(x, \underline{e})$

By our choice of Φ,

$Q \vdash \neg\Phi \leftrightarrow (\exists x)\,\mathrm{Prf}_T(x, \underline{e_1})$

So $Q \vdash \neg\Phi$ $\quad$ ($Q \vdash A \leftrightarrow B$; $Q \vdash B \Rightarrow Q \vdash A$)

But Π interprets Q in T

$\therefore T \vdash \neg\Phi^*$ $\quad$ so $T \vdash \Phi^*$, $T \vdash \neg\Phi^*$

so T is inconsistent $\quad$ QED (Claim 1)

~~Now ~~assert~~ assume T is ω-consistent and~~

~~$T \vdash \neg \Phi^*$ will get a contradiction.~~

~~since T is consistent~~

defn: T is ω inconsistent, if for some formula $\varphi(x)$ of L_T, $T \vdash (\exists x)(\pi(\exists)(x) \wedge \varphi(x))$

For each $n \in \omega$

$$T \vdash (\forall x)(\pi(\exists)(x) \wedge [``x = \underline{n}"]^* \rightarrow \neg \dot{\varphi}(x))$$

$$(\text{ie } T \vdash ``\neg \varphi(\underline{n})")$$

Suppose T is ω-consistent and $T \vdash \neg \Phi^*$ will go for a contradiction

For each $f \in \omega$, f is not G# of a proof of Φ^* (by claim f) since T is consistent

$$\therefore Q \vdash \neg \mathrm{Prf}_T(\underline{f}, \underline{e_1})$$

Let $\varphi(x) = [\neg \mathrm{Prf}_T(x, \underline{e_1})]^*$

so $T \vdash \varphi(\underline{f})$ all f

but also $T \vdash \neg \Phi^*$

which this amounts to $T \vdash (\exists x)(\pi(\exists)(x) \wedge \varphi(x))$

So T is ω inconsistent

Works only because there are non standard models of arithmetic. (Else actually complete)

Recall Godel II says:

If $T \vdash con(T)$ T is inconsistent. (P rel. interp in T, axioms of T recursive)

Proof of Godel II:

(※) By formalizing proof of claim 1 in P,

we see

$P \vdash con(T) \to$ "Φ^* is not provable in T"

But $Q \vdash \Phi \leftrightarrow$ "Φ^* is not provable in T"

So $P \vdash con(T) \to \Phi$; P rel interp in T

So $T \vdash con(T)^* \to \Phi^*$

If $T \vdash con(T)^*$ then $T \vdash \Phi^*$

so by claim 1 T is inconsistent

Lemma:

The <u>formalizations</u> of the following are provable in P:

(1) If $Q \vdash \varphi$ (φ a sentence of L_P), then $T \vdash \varphi^*$

(2) If $T \vdash \varphi$ then $Q \vdash (\exists x)\, Prf(x, \ulcorner\varphi\urcorner)$

let φ be a sentence of L_T

(3) a If $Q \vdash \varphi$, $Q \vdash \varphi \to \psi$ then $Q \vdash \psi$ (φ, ψ sentences of Q)

b (let φ, ψ be sentences of L_T)

If $T \vdash \varphi$,

$T \vdash \varphi \to \psi$

then $T \vdash \psi$

Lemma:

The following (when suitably formalized) are provable in P

$\boxed{1}$ If $Q \vdash \varphi$ then $T \vdash \varphi^*$ ($\varphi \in \mathrm{Sent}\, P$)

$\boxed{2}$ If $T \vdash \varphi$, then $Q \vdash (\exists x)\, Prf_T(x, \underline{\# \varphi})$ ($\varphi \in \mathrm{Sent}\, T$)

$P \vdash \mathrm{Cont}(T) \to$ "Φ^* is not provable"

$\boxed{3a}$ If $Q \vdash \varphi$ and $Q \vdash \varphi \to \psi$ then $Q \vdash \psi$ ($\varphi, \psi \in \mathrm{Sent}\, P$)

$\boxed{3b}$ If $T \vdash \varphi$, $T \vdash \varphi \to \psi$ then $T \vdash \psi$ ($\varphi, \psi \in \mathrm{Sent}\, T$)

$\boxed{4}$ $(\neg\varphi)^* = \neg(\varphi)^*$ immediate from defn of translation of negation

$\boxed{5}$ If $\Theta(x, x_1, \dots, x_n)$ is a standard translation of a primitive recursive predicate, and $a_1, \dots, a_n \in \omega$ and
$\eta \models (\exists x)\, \Theta(x, \underline{a_1}, \dots, \underline{a_n})$
Then $Q \vdash (\exists x)\, \Theta(x, \underline{a_1}, \dots \underline{a_n})$

Since $\eta \models (\exists x)\, \Theta(x, \underline{a_1}, \dots, \underline{a_n})$ for some $a \in \omega$,
$\eta \models \Theta(\underline{a}, \underline{a_1}, \dots, \underline{a_n})$ But Θ is num decidable in Q and $\eta \models Q$

So $Q \vdash \Theta(\underline{a}, \underline{a_1}, \dots, \underline{a_n})$
So $Q \vdash (\exists x)\, \Theta(x, \underline{a_1}, \dots, \underline{a_n})$

Now assume lemma and show:
$P \vdash \mathrm{Con}(T) \to$ "Φ^* is not provable"

Enough to see:
$P +$ "Φ^* is ~~not~~ provable" $\vdash \neg \mathrm{Con}(T)$
$\to \exists x\, Prf_T(x, \underline{\# \Phi^*})$

Work in $P +$ "Φ^* is provable in T"
$Q \vdash \Phi \Leftrightarrow$ "Φ^* is not provable" (Fact 5)
$Q \vdash$ "Φ^* is provable in T" $\to \neg\Phi$
We have $(\exists x)\, Prf_T(x, \underline{\# \Phi^*})$

By fact $\boxed{2}$ $Q \vdash (\exists x) Prf_T(x, \underline{\# \Phi^*})$
By fact $\boxed{3a}$ $Q \vdash \neg\Phi$

By facts $\boxed{1}$ & $\boxed{4}$ $T \vdash (\neg\Phi)^*$ $T \vdash \neg(\Phi^*)$

By fact $\boxed{5}$ $\quad T \vdash \neg\Phi^* \to (\Phi^* \to F)$; Tautology

By fact $\boxed{3b}$ $\quad T \vdash \Phi^* \to F$ $\quad$ (From standard falsehood $0=$

Note: $P \vdash (\forall x)$ (If "x is G* of a sentence of T" and

* $(\exists v_0)\, Prf_T(v_0, x)$

$(\exists y)\, Prf_T(y, x)$

then $(\exists z)\, Prf_Q(z, h(x))$

$(\exists z)(\exists y)\,[y = h(x) \wedge Prf_Q(z, y)]$

But also $T \vdash \Phi^*$ (an axiom of P_t "$T \vdash \Phi^*$"

By fact 3b, $T \vdash F$

i.e. $\neg Con(T)$

Now we will handwave through topic:

What can one formalize in P

Phase 1.

Basic arithmetical facts

(1) $x + (y+z) = (x+y) + z$

(2) $x + y = y + x$

(3) $x \cdot (y+z) = x\cdot y + x \cdot z$

(4) $x \cdot (y \cdot z) = (x \cdot y) \cdot z$

(5) $x \cdot y = y \cdot x$

$$x \le y \equiv_{dfn} (\exists z)(y = x + z)$$

(6) $x \le y$ or $y \le x$

(7) $[x \le y \;\&\; y \le x] \to x = y$

$\boxed{8}$ $\quad x \le z \to (x + y \le y + z)$

$\boxed{9}$ $\quad x \le y \to x \cdot z \le y \cdot z$ $\qquad$ Dfn $1 = S0$

$\boxed{10}$ $\quad x \ne 0 \quad 1 \le x$

Sample Proofs: Induct on last variable (∵ our rec defns were on the last variable)

1) Case 1 $z=0$

$$\cancel{x+y} \quad x+(y+0) = (x+y)+0 = x+y$$

$$(x+y)+0 = x+y$$

Totally formalizable in P! using induction

Assume $(x+y)+z = x+(y+z)$

To see $(x+y)+Sz = x+(y+Sz)$

$$(x+y)+Sz =$$
$$S((x+y)+z) =$$
$$S(x+(y+z)) =$$
$$x+S(y+z) =$$
$$x+(y+Sz))$$

① $\varphi(z)$: $(x+y)+z = x+(y+z)$

Ind. Schm $\varphi(0) \wedge \forall x[\varphi(x) \to \varphi(Sx)] \to \forall x\, \varphi(x)$

$x \cdot y = y \cdot x$ Assume Facts 1–4 Done

Prove by induction on y

$y=0$ $\quad X \cdot 0 = 0 \cdot X$

well $X \cdot 0 = 0$ (2nd defn of mult)

ETS $0 \cdot x = \cancel{x}\ 0$

Lemma $0 \cdot x = 0$

$0 \cdot 0 = 0$

if $0 \cdot x = 0$

$$0 \cdot Sx = 0x + 0$$
$$= 0 + 0 \quad \text{(IH)}$$
$$= 0$$

Now assume $x \cdot y = y \cdot x$

To see $x \cdot Sy = Sy \cdot x$

$x \cdot Sy = x \cdot y + x$

ETS $Sy \cdot x = y \cdot x + x$

If so we would be done

Lemma: $Sy \cdot x = y \cdot x + x$

Proof: induction on x

$\underline{x=0}$ $Sy \cdot 0 = 0 = y \cdot 0 + 0$

assume $Sy \cdot x = y \cdot x + x$

To see $Sy \cdot Sx = y \cdot Sx + Sx$

$(Sy) \cdot (Sx) =$

$(Sy) \cdot x + Sy =$

$[(y \cdot x) + y] + [y + 1] =$

comm & assoc laws: $y \cdot x + y + x + 1 =$

$y \cdot Sx + Sx$

Peano Arithmetic Elem facts about $+, \cdot, \le$ (Phase 1)

Phase 2 Prove standard facts about β-function

1) Prove as a theorem schema least number principle

$$(\exists x)\ \varphi(x) \to (\exists x)[\varphi(x) \wedge (\forall y)(y < x \to \neg\varphi(y))]$$

Usual Proof Works

(2) ~~Pf~~ Division w/ remainder

$\forall x\ \forall y > 0\ \exists q\ \exists r :\ x = qy + r$ and $0 \le r < q$ and

q and r are unique

Proof: By induction on x for existence

Uniqueness is easy

GCD Theorem is provable in P

If $a, b \in \omega$ $\exists \alpha, \beta \in \mathbb{Z}$: if $u = \alpha a + \beta b$

$\Rightarrow$ u divides a and u divides b

Usual Proof Works (Use least # principle!)

3) PA $\vdash$ CRT for two variables

$x > 0, y > 0$ are rel prime

$u < x \quad v < y$

$\exists a :$ rem$(a, x) = u$

rem$(a, y) = v$

Usual proof works

$\beta(c, d, i) = \mathrm{rem}(c, 1 + (i+1)d)$ [illegible]

(∵ of uniqueness of remainder

$B(c, d, i, u)$ $\qquad u < 1 + (i+1)d$

$u = \beta(c, d, i)$ $\qquad \& \; \exists q : c = q \cdot (1 + (i+1)d) + u$

$\Phi(n) : \forall c \, \forall d \, \forall x \, \exists c' \exists d' : \cancel{(\forall i < n)\, B(c, d, i)}$

(1) $\forall u \, (\forall i < n) \; B(c, d, i, u) \leftrightarrow B(c', d', i, u)$

$[(\forall i < n) \; \beta(c, d, i) = \beta(c', d', i)]$ ← translation

(2) $B(c', d', n, x)$

↑ whole thing is $\Phi(n)$; only free variable is n!

We will prove in P $\forall n \, \Phi(n)$

By least number principle, it suffices to show

$\Phi(n)$ assuming $\underbrace{(\forall j < n)\, \Phi(j)}_{*}$

$*$ allows us to handle sequences of length $\leq n$

and now the usual proof works

↑ Need lemma

$\forall n \, \exists m :$

$(\forall i \leq n)(i \text{ divides } m)$

↳ effectively using exponentiation

(∵ m would be $\simeq 2^n$)

Next goal:

Our given formalized defns of prim rec fns define total functions in P, and usual defining equations are thms

Next:

(1) Gödel number prim recursive fcns
(Not every number will be a Gödel number)

(2) If n is a G# of a k place fcn
$n = 2^k \cdot \text{odd}$

(3) "n is G# of a prim rec fcn" is prim rec

$2^1 3^1$ is G# of $Z : \omega \to \omega$ $\quad Z(n) = 0 \ \forall n$
$2^1 3^2$ is G# of $S : \omega \to \omega$ $\quad S(n) = n+1$
$2^n 3^3 5^i$ (if $n > i \geq 0$) is G# of π^n_i (Recall $\pi^n_i(x_0, \dots, x_{n-1}) = x_i$)

If r is G# of $g : \omega^m \to \omega$,
and $s_1, \dots, s_m$ are G#s of $h_1, \dots, h_m : \omega^n \to \omega$,
$2^n \cdot 3^4 \cdot 5^r \cdot 7^{s_1} \cdot \dots \cdot p_{m+2}^{s_m}$ is G# of composition
$f(\vec{x}) = g(h_1(\vec{x}), \dots, h_m(\vec{x}))$

If $c \in \omega$ and s is G# of $g : \omega^2 \to \omega$
$2^1 \cdot 3^5 \cdot 5^c \cdot 7^s$ is G# of f

$f(0) = c$
$f(n+1) = g(n, f(n))$

If $g : \omega^{n-1} \to \omega$ has G# r,
$h : \omega^{n+1} \to \omega$ has G# s
$2^n \cdot 3^5 \cdot 5^r \cdot 7^s$ is G# of f,
where $f(x_1, \dots, x_{n-1}, 0) = g(x_1, \dots, x_{n-1})$
$f(x_1, \dots, x_{n-1}, Sy) = h(x_1, \dots, x_{n-1}, y, f(\vec{x}, y))$

Our prior work associates to each n, the prim rec G^* of $f: \omega^k \to \omega$, a formula $\Theta_n(v_0, \dots v_k)$ which numerically represents f

$$\Theta(v_0, v_1, \dots, v_k) \sim v_0 = f(v_1, \dots, v_k)$$

By induction on n (in metatheory) can show

$$P \vdash \forall v_1 \dots \forall v_k \exists! v_0 \; \Theta_n(v_0, v_1, \dots, v_k)$$

Showing it works for prim rec → use β fn, sequence of values

The defining equations of f_n are theorems of P

$\Theta_Z(x, y)$ $\quad x = 0$

$\Theta_S(x, y)$ $\quad$ "$x = Sy$"

$\Theta_{\Pi_i^n}(v_0, v_1, \dots v_n) =$ "$v_0 = v_{i+1}$"

f's defining equation is

$$f(x) = g(h(x))$$

$$P \vdash \Theta_f(z, x) \leftrightarrow (\exists y)[\Theta_g(z, y) \wedge \Theta_h(y, x)]$$

If
$$f(x, 0) = g(x)$$
$$f(x, Sy) = h(x, y, f(x, y))$$

$$P \vdash \Theta_f(z, x, 0) \leftrightarrow \Theta_g(z, x)$$

$$P \vdash \Theta(z, x, Sy) \wedge \Theta_f(w, x, y) \rightarrow \Theta_h(z, x, y, w)$$

Wednesday

- Finish talking about formalizing in P
- there is a finitely axiom sub thry of P (much more powerful than Q) in which all this can be formalized

Following summarizes where we are:

Let P^+ be theory with:

$L_{P^+} = L_P +$ a k-ary fcn symbol f_u whenever u is a G# of a k-ary prim rec fcn

Axioms of P^+: axioms of P +

$\forall x_1 \cdots \forall x_k [f_u(x_1, \ldots, x_k) = x_0 \Leftrightarrow \theta_u(x_0, \ldots, x_k)]$

P numeric represent fn

Any model of P uniquely extends to a model of P^+

(clear since $P \vdash \forall x_1 \ldots \forall x_k \exists! \, \theta_u(x_0, \ldots x_k)$)

2) Corollary: If $P^+ \vdash \Phi$

Φ a sentence of L_P,

then $P \vdash \Phi$

i.e. P^+ is a "conservative extension" of P"

Proof:

If $P \nvdash \Phi$,

by completeness theorem,

$P + \neg\Phi$ has a model M

M can be expanded to a model of $P^+ + \neg\Phi$

so $P^+ \nvdash \Phi$

③ P^+ ~~#~~ can prove defining equations of all the fn's

upto interp, P^+ is a subthy of P

P^+ is "primitive recursive arithmetic"

"finitist" something is finitist if it can be formalized in $P^+ \equiv$ P.R.A. (prim rec arith)

Discussion of formalization of our four facts -

Fact ④ $(\neg\theta)^* = \neg(\theta^*)$ $\quad\quad$ $Prf_T(x,y)$

immediate

Fact 3a: if $Q \vdash \varphi$, $Q \vdash \varphi \to \psi$ then $Q \vdash \psi$

Need Lemma: If Π_1 and Π_2 are Q proofs, so is $\Pi_1 * \Pi_2$ a Q proof

If Π_1 proves φ

Π_2 proves $\varphi \to \psi$

then $\Pi_1 * \Pi_2 * \langle\psi\rangle$ is a proof of ψ

Fact 3b: is totally similar

Fact 2: if $T \vdash \varphi$, then $Q \vdash \exists x\, Prf_T(x, \underline{\#\varphi})$

By induction on n,

formalizing our earlier work on nu up in Q,

one shows: if n is G# of a p.r.-fcn

then $P \vdash \theta_n(x_0, x_1, \ldots, x_k) \to$ "$Q \vdash \theta_n(\underline{x_0}, \ldots, \underline{x_k})$"

formula of P

$Prf_T(x,y)$

$f(x,y)=1$

prim fcn h : $H(y, x_0, \ldots, x_k)$

$h(x_0, \ldots x_k) = (\underline{y = h(x_0, \ldots, x_k)})$

G# of $\Theta_n(\underline{x_0}, \ldots, \underline{x_n})$

$\exists y\, Prf_Q(y, h(x_1, \ldots, x_n))$

$\exists y \exists z\, (Prf_Q(y,z) \wedge H(z, x_0, \ldots, x_k)]$

In particular

(a) $P \vdash \forall xy\; Prf_T(x,y) \rightarrow$ " $Q \vdash Prf_T(\underline{x}, \underline{y})$"

(b) If φ is (G#) of a sentence of L_T

and $\exists y\, Prf_T(y, \varphi) \rightarrow$ " $\exists y\; Q \vdash Prf_T(\underline{y}, \underline{\varphi})$"

(c) $P \vdash$ If $\exists y\; Q \vdash Prf_T(y, \#\varphi) \rightarrow Q \vdash (\exists y) Prf_T(y, \underline{\#\varphi})$

$\therefore$ Get $P \vdash \exists y\, Prf_T(y, \underline{\#\varphi}) \rightarrow Q \vdash (\exists y) Prf_T(y, \underline{\#\varphi})$ <u>Fact 2</u>

<Fact 1> If $Q \vdash \varphi$ then $T \vdash \varphi^*$ (φ a sentence)

Have to prove by induction on the length of a Q proof.

Actually prove:

(*) If $Q \vdash \varphi(v_0, \ldots, v_k)$ then $T \vdash \forall v_0 \ldots \forall v_k\, [\Pi(\exists)(v_0) \wedge \ldots \Pi(\exists)v_k \rightarrow \varphi^*(v_0, \ldots, v_k)]$

Proof actually yields a ~~proof~~ prim rec fcn F:

if c is a Θ proof of $\varphi(v_0, \ldots, v_n)$

$F(c)$ is a T proof of (*)

Defn of F splits into cases accordingly according to type of last line of c

Etc. Details will not be given

OS upgrade X 2 and use next unit of time

$\Psi(\vec{x},$

$\cancel{\Psi(\varphi_P}$

$\Psi(P_\varphi, \vec{x}, t, y)$ iff

$(y = T_n(P_\varphi, x_1, \ldots, x_n, t))$

$$\exists t \left\{ [y = T_n(P_\varphi, x_1, \ldots, x_n, t)] \wedge \right.$$

$$\Psi(x_1, \ldots, x_n, y) \leftrightarrow$$

$$(\exists t)\left[H(T_n(P_\varphi, \vec{x}, t)) \wedge (y = U(T_n, \vec{x}, t)) \right]$$

→ Mackey

→ Bell's ineq

→ Von Neumann

→ Mermin "Rootmus all the way down"