R Solovay
M125A
717 Evans
Off Hrs MWF 3^{10} – 4 pm

Logic

Xams Underlying $\neg$ (not) $\vee$ (or) $\exists$ (there exists) } > fund. different

– Formal Language for Propositional Logic (so Alphabet) (for finite alph)

Have to describe (1) alphabet
(2) which sequences of symbols from our alphabet are wffs:

$A\emptyset$, $A1$, $A2$, ... $\rightarrow$ $A(1)$, $A(2)$, ...

Sequence: fn whose domain is a finite initial segment of integers

$s : \{0, \dots, n-1\} \rightarrow \underline{X}$

$\langle s(0), s(1), \dots, s(n-1) \rangle$

Difference between x and sequence of length 1 $\langle x \rangle$

alphabet

(left parens
) right parens
$\neg$ not
$\vee$ or
$\wedge$ and
$\rightarrow$ if then
$\leftrightarrow$ iff

infinite list of propositional constants.

$A_0 \ A_1 \ A_2 \ \dots$

Inductive Defn of "well formed formulae"

(1) For any i in ω $(=\{0,1,2,\ldots\})$
$\langle A_i \rangle$ is a well formed formula

(2) If P and Q are wffs then
$$\left.\begin{array}{ll} (\neg P) & \text{is a formula} \\ (P \vee Q) & \text{"} \\ (P \wedge Q) & \text{"} \\ (P \rightarrow Q) & \text{"} \\ (P \leftrightarrow Q) & \text{"} \end{array}\right\} \text{are wff}$$

By $(P \rightarrow Q)$ I really mean
$\langle (\rangle \frown P \frown \langle \rightarrow \rangle \frown Q \frown \langle) \rangle$

(3) The only wff are those given by rules (1) and (2)

<u>Key fact</u>: Unique Readability
If R, P, Q, S, T are wff
if $R = (P \wedge Q)$
then $R \neq (S \vee T)$ etc

<u>Also</u> if $R = (P \vee Q)$
and $R = (S \vee T)$
then $P = S$; $Q = T$

We will name
two elements T "true"
F "false"

Let $S = \{A_0, A_1, A_2, \ldots\}$
$\bar{S} = \{I \mid I \text{ is a wff}\}$

Suppose we are given
$\phi: S \rightarrow \{T, F\}$

I'm now going to define its canonical prolongation to $\bar{S}$
$\bar{\phi}$ to $\bar{S}$.

(1) $\bar{\varphi}(A_i) = \varphi(A_i)$

(2) (a) If $P = \neg Q$,

$\bar{\varphi}(P) = T$ if $\bar{\varphi}(Q) = F$

$= F$ if $\bar{\varphi}(Q) = T$;

(b) $\bar{\varphi}((P \cup Q)) = T$ if either $\bar{\varphi}(P) = T$ or $\bar{\varphi}(Q) = T$

$= F$ otherwise

(c) $\bar{\varphi}((P \wedge Q)) = T$ if $\bar{\varphi}(P) = T$ and $\bar{\varphi}(Q) = T$

$= F$ otherwise

(d) $\bar{\varphi}((P \to Q)) = F$ if $\bar{\varphi}(P) = T$ and $\bar{\varphi}(Q) = F$

$= T$ otherwise

(e) $\bar{\varphi}((P \leftrightarrow Q)) = T$ if $\bar{\varphi}(P) = \bar{\varphi}(Q)$

$= F$ otherwise

P	¬P
T	F
F	T

P	Q	$P \cup Q$	$P \wedge Q$	$P \to Q$	$P \leftrightarrow Q$
T	T	T	T	T	T
T	F	T	F	F	F
F	T	T	F	T	F
F	F	F	F	T	T

M125a

Sept 6, 1991
Friday

HW Due Fri 13 (Sept)
Props 2, 3, 6, 7^+, 11^- (look at 5, 8)
pp 38-39 Enderton

$S = \{A_i \mid i \in \omega\}$
$\bar{S} = \{P \mid P \text{ a wff}\}$
$v: S \to \{T, F\}$
then v has a canonical prolongation to
$\bar{v} : \bar{S} \to \{T, F\}$
following truth tables

<u>Example</u> (of Tautology) $\quad P \vee (\neg P) \leftarrow$ tautology

P	$\neg P$	$P \vee (\neg P)$
T	F	T
F	T	T

<u>Defn</u> Let Σ be a set of wffs
let τ be a wff
then Σ tautologically implies τ
($\Sigma \models \tau$ or $\tau \Dashv \Sigma$)
If:
Whenever $v: S \to \{T, F\}$ is a valuation
and $\bar{v}(\varphi) = T$ for each $\varphi \in \Sigma$,
then $\bar{v}(\tau) = T$.

<u>Examples</u> Let A, B be wffs
If $\Sigma = \{A, \neg A\}$ then $\Sigma \models B$.
Need to see: if $v: S \to \{T, F\}$ is a val
and $\bar{v}(A) = T$
$\bar{v}(\neg A) = T$ then $\bar{v}(B) = T$

Fourier

If a system proves a sentence and its negation, then it proves everything.

$\{A, A \to B\} \models B$

Let τ a wff

<u>Def</u> τ is a tautology
if for any valuation $v: S \to \{T, F\}$
$\bar{v}(\tau) = T$

<u>Example</u> $A \cup (\neg A)$

<u>RK</u> The following are equivalent:
(1) τ is a tautology
(2) $\phi \models \tau$
set w/ no members

<u>Proof</u> (1) $\to$ (2) clear
(2) $\to$ (1) Let $v: S \to \{T, F\}$ be a valuation
To see $v(\tau) = T$.
But clearly for any $u \in \phi$, $\bar{v}(u) = T$ (no such ϕs
and $\phi \models \tau$
so $\bar{v}(\tau) = T$

———x———

$(\neg (P \cup Q)) \leftrightarrow (\neg P \wedge \neg Q)$
- check by method of Truth Tables

P	Q	$\neg P$	$\neg Q$	$P \vee Q$	$\neg(P \vee Q)$	$(\neg P) \wedge (\neg Q)$	original wff
T	T	F	F	T	F	F	T
T	F	F	T	T	F	F	T
F	T	T	F	T	F	F	T
F	F	T	T	F	T	T	T

More examples of tautologies:
all T's so a tautology

(1) $A \vee (\neg A)$

(2) $\neg\neg A \Leftrightarrow A$

(3) $\neg(A \vee B) \Leftrightarrow (\neg A) \wedge (\neg B)$

(4) $\neg(A \wedge B) \Leftrightarrow (\neg A) \vee (\neg B)$

(5) $(A \rightarrow B) \Leftrightarrow (\neg B \rightarrow \neg A)$

Remarks on Proof of Unique Readability:

- all wffs: number of left parens = number of right parens
  "Balanced"

M125a - Lecture

Sept 9, '91
Monday

- Prove Unique Readability
- Algorithm to test for well-formedness
- Disjunctive Normal Form

Suppose $(P \wedge Q)$ is $(R \wedge S)$

Then P is an initial segment of R or
R is an initial segment of P

<u>Key pt</u>: If P and R are wffs, and P is an initial segment of R then $P = R$.

<u>Lemma 1</u>: If P is a wff, then P is 'balanced' (i.e. has the same number of left and right parentheses)

<u>Proof</u>: Already known

<u>Lemma 2</u> If P is a wff and Q is a nonempty proper (i.e. $\neq P$) initial segment of P, then Q has more left parens than right parens! (so Q is not balanced)

<u>Proof</u>: By induction on length of P.

(<u>case 1</u>) $P = A_i$ Utterly clear ($l = 0 \Rightarrow$ empty)

(<u>case 2</u>) P is of form $(\neg R)$

So Q is one of the following

✓ (a) "("

✓ (b) "$(\neg$"

(c) "$(\neg R_0$" where R_0 is a proper non empty initial segment of R

✓ (d) "$(\neg R$"

wff $\Rightarrow$ same # left & right

case (c) R_0 has more left parens than right parens (by IH)

$(\neg R_0$ has more left ('s than R_0

same # of right)'s

so it's o.k.!

Case 3 ~~R is~~ P is of form $(R \alpha S)$
where R, S are wffs and α is one of $\{\cup, \cap, \rightarrow, \leftrightarrow\}$

Possibilities for Q

(a) $Q =$ "("

(b) $Q =$ "$(R_0$" ; R_0 proper non empty initial part of R

(c) $Q =$ "$(R$"

(d) $Q =$ "$(R\alpha$"

(e) $Q =$ "$(R \alpha S_0$" ; S_0 " " " " " S

(f) $Q =$ "$(R \alpha S$"

$\mathcal{A}$ = set of all atomic formulae $= \{A_i \mid i \in \omega\}$

$\bar{S} = \{P \mid P \text{ is a wff}\}$

$E_\neg : \bar{S} \longrightarrow \bar{S}$

$E_\neg(\text{"}P\text{"}) = \text{"}(\neg P)\text{"}$

If α is a binary connective

$E_\alpha : \bar{S} \times \bar{S} \rightarrow \bar{S}$

$E_\alpha(\text{"}P\text{"}, \text{"}Q\text{"}) = \text{"}(P \alpha Q)\text{"}$

here $\alpha \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$

Unique Readability:

(1) S and range $(E_\neg)$ and rang $(E_\cup)$ and ... and range $(E_\leftrightarrow)$ are pairwise disjoint

(E.g. no wff is both A_i and of the for $(\neg P)$)

(2) Each of the functions $E_\neg, E_\vee, E_\wedge, \ldots E_\leftrightarrow$

Proof (1) Its clear no wff P is in S and also in range E.

Why something in S has length 1

" " $rg(E_\neg)$ has length ≥ 3

(2) If a wff has the form $(\neg P)$ it can't have form $(R \alpha S)$ for α a binary connective

Why: If so first symbol of R is "$\neg$" But it is clear that the first symbol of a wff is either A_i some i or "("

(3) If "$(\neg P)$" = "$(\neg Q)$"
then "P" = "Q"
Get P (or Q) from "$(\neg P)$" by stripping off first two symbols and last symbol.

So $E_\neg$ is 1-1

We'll be done if we prove:
"$(P \alpha Q)$" = "$(R \beta S)$"
where P, Q, R, S are wffs and α, β are b.c.'s (binary connectives) then "P" = "R" "α" = "β" "Q" = "S"

shows disjointness

will show completely equal!

If "$(P \alpha Q)$" = "$(R \beta S)$"
then "$P \alpha Q)$" = "$R \beta S$"

So either "P" = "R" or "P" is a proper initial segment of "R"
or "R" is a proper initial segment of "P"

But P, R are wffs, so are balanced (Lemma 1)
So P can't be a proper (non-empty) initial seg of R (Lemma 2)
and also R can't be a proper (" ") " " " P
So $P = R$
and "$\alpha Q)$" = "$\beta S)$"
so "α" = "β"
and "Q" = "S"

See the impt of parens: $A_1 \cup A_2 \cup A_3$

not same as RP

Endorton: ordered tuple would work

$P = A_1 \quad Q = A_2 \cup A_3$
$R = A_1 \cup A_2 \quad S = A_3$

M125a

11 Sept '91
Wednesday

<u>Goal</u>: To show $\exists$ an algorithm for determining if a string of symbols is a "wff"

non wff $((\neg A_1)$

alphabet $\Sigma = \{ (,), \neg, \cup, \cap, \rightarrow, \leftrightarrow, A_0, A_1, A_2, \ldots \}$

<u>Lemma 1</u>: Let P be a string of symbols from Σ

$: (A_1 \cup A_5)$ Let P' be the result of replacing A_i in P by A_0

$: (A_0 \cup A_0)$ Then P is a wff $\Leftrightarrow$ P' is a wff

<u>Proof</u>: Notice length(P) = length(P')
(Proof) By induction on length (P)

<u>Case 1</u> : $P = A_i$ clear
<u>case 2</u> : $P' = A_0$ clear (P is some A_i)
<u>case 3</u> : P is a wff of the form $(\neg Q)$ where Q a wff
$P' = (\neg Q')$ By IH Q' is a wff. So P' is a wff
<u>Case 4</u> : P' is a wff of form $(\neg Q')$ (Q' a wff)
P must have form $(\neg Q)$. By IH; Q a wff
so P is a wff
etc $(Q \circ R)$ $\leftarrow$ apply in same way

<u>Lemma 2</u> Let P be a string from Σ
Let P' be result of replacing in P
each occurrance of "$\cap$", "$\rightarrow$", "$\leftrightarrow$", by "$\cup$"
Then P is a wff $\Leftrightarrow$ P' is a wff
<u>Proof</u> similar to preceding lemma

<u>algorithms</u>
Eg 1) If m, n are decimal representations of integers, there is an algorithm for finding decimal representations of $m+n$, $m-n$, and provided $n \neq 0$ the quotient and remainder after dividing m by n.

Algorithm (1) always produces the right result
(2) no creativity required
(3) definite recipe telling what to do at each step

⇒ Computable functions: Post, Turing, Church, Kleene, Gödel, Herbrand

Alternate defn of algorithm:
Define idealized digital computer & program for it.
Algorithms are what those can do.
Our theoretical notion of algorithm allows arbitrarily long (but finite) running times

Euclidean Algorithm
finding gcd of two numbers

Questions not solvable by algorithms:
(1) Given a formulae of first order logic, is it logically valid.
(2) Given a sentence about number theory, is it true?
(for $\mathbb{R}$, there is such a procedure $\langle \mathbb{R}, +, \times, 0, 1, < \rangle$ there is an algo) Tarski
Hilbert's 10th problem (3) There is no algorithm to determine if a polynomial (in $\mathbb{Z}$) eqn $P(x_1, \ldots, x_n)$ has a soln in the integers $\mathbb{Z}^n$.
(there is a decision procedure for soln in $\mathbb{R}^n$)

(4) Groups can be represented by generators and relations:
$g_1, \ldots, g_n$ are generators
$R_1, \ldots, R_m = e$ are relations
Given a presentation and a word on $g_1, \ldots, g_n$, $g_1^{-1}, \ldots, g_n^{-1}$
Q: ("Word Problem") Is some given word $w = e$ in the group determined by gen $g_1 \ldots g_n$ & $R_1 \ldots R_m$
→ Known ~~to be~~ not to have a d...

Sept 13, 1991
Friday

Algorithm for testing if a string of symbols from $\{_z (,), \neg, \vee, A_x\}$ is a wff

<u>Input</u> : string of symbols P
<u>Output</u> : Yes (It is a wff)
No (It isn't a wff)

Step 0: List all non empty substrings in order of increasing length
$Q_0, \ldots Q_m$
with $Q_m = P$

$P = (\neg)$
(
)
$\neg$
$(\neg$
$\neg)$
$(\neg)$

(Q is a substring if $P = R^\frown Q^\frown S$)
R, S possibly empty

Our algorithm will first settle
Is Q_0 a wff?
Is Q_1 a wff?
- - - - -
Is Q_n a wff?

The effect of all this is:
⟹ In deciding if P is a wff we can assume know the answer for a proper substring of $P_{\#}$.

⟹ So we're reduced to the following question:
Determine if P is a wff knowing answer for all proper substrings of P

Here we go (around, round, round...)

① If $P = \langle A_0 \rangle$ its a wff (Answer "Yes") o.w. go to Step ②

② If first symbol of P is not a "("
answer no. If it is, then $P = (^\frown P_1$
Go to ③

③ If first symbol of P_1 is "¬" go to (4).
If not go to (5)

/*Negations*/ ④ So P_1 has form $\neg^\frown P_2$
If P_2 doesn't end w/ a ")" answer no.
If it does, P_2 has form "$P_3{}^\frown$)"
If P_3 is a wff answer yes
If not answer no.

⑤ In this case $P = $ "$(^\frown P_1$

Look for shortest non empty balanced initial segment of P_1 call it Q
If no such Q, answer "No".
If it is see if Q is also a wff. If no answer "No"
If yes $P_1 = Q^\frown R$
If R doesn't have form "$\vee^\frown R_2{}^\frown$)",
answer "no".
If does form, test if "R" is a wff
If so answer "yes"
If not answer "no"

End of algorithm

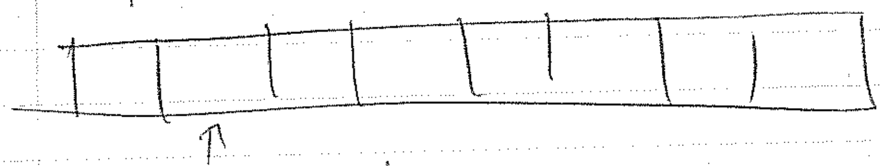
(One must check by induction on length of P)
① Halts clearly
② Algo always gives right answer
Time estimate: $O(n^3)$ steps

Work Tape:

Symbols on ~~input~~ work tape: "B, W,), V, —"

Symbols on input tape: "(,), ¬, V, A_0, —"

Tape is ∞ # of squares, each sq. cons a symbol



Reading head

Never write or erase on input tape (Read only tape)

Only move to right on input tape

on work tape we will always look at rightmost non blank symbol.

<u>Algorithm</u>: At start Word P is on input tape we are scanning 1^{st} symbol of P. work tape is blank.

0. Write BW on work tape
 Transition head of work tape to be on W
 Goto Step 1.

<u>Step 1.</u>

<u>Case B</u>: If symbol read on input tape is blank, answer yes, if not answer no.
If symbol read on i/p tape answer Yes
If not answer no.

~~Case 1~~ <u>case 2</u> Symbol on work tape is ")"
If symbol on input tape is not ")"
answer "No" and halt.

<u>case 3</u> like case 2 but replace ")" by "V" throughout

case 4: symbol on work tape being read is "w"

case 4a: symbol on i/p tape is $A_\emptyset$
~~[illegible]~~
Erase w and go 1-step to left on work tape
Go one square to right on input tape

case 4b: Symbol under tape is neither "$A_\emptyset$" or "C"
Answer No and halt

case 4c: Symbol under tape is a "C". move i/p tape one square to the right

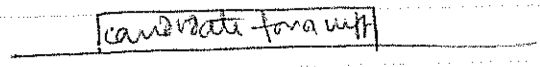
(To be contd)

M25a (Lecture)

Sept 16 '91
Monday

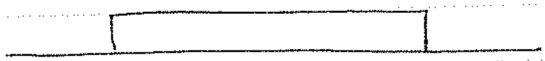
Midterm (in class) Fri Oct 4

HW (due this Friday) p 43 #2 p 52 #2 p 52 #4 (Why deg 2 is correct)



← i/p tape

read only tape
move only to right



← work tape

), v, B, W

↑ well formed formula

case (4) W is scanned symbol on work tape

case (4a) If A_ϕ scanned on input tape
move input tape head 1 square to right
(4b) Erase w on work tape end and move one square to the left. Go to start of step 1.

case (4B) If symbol not an A_ϕ and not a "C" then answer is No and halt

case (4C) Symbol scanned on i/p tape is "C"
First move input head one square to the right

case (4c.1) If scanned symbol now is "$\neg$"
(a) move input head 1 square to right
(b) erase w on work tape with)w in first two blank squares on work tape

case (4.c.2) Erase 'w' on work tape with) w v w on work tape in first 4 blank squares. Move work head 3 squares to right

'Proof by Erasure'

<u>Example</u> $((\neg A_\emptyset) \vee A_\emptyset)$ Input Tape

now treated?

B W
B) W V W
B) W V) W
B) W V)
B) W V
B) W
B)
B

Worktape

———— x ————

§1.5 DNF (Sentential Connectives)

skip (1.6 Switching Circuits
1.7 Effectiveness and compactness

T 1
F 0

$\{T, F\} = \{0, 1\} = 2$ $\therefore 2^n = \{T, F\}^n$

↑ set theorists view 2 as $\{0, 1\}$

$\{T, F\}^n = \{ \langle u_1, \ldots, u_n \rangle \mid \text{each } u_i \in \{T, F\} \}$

$\mathrm{card}(\{T, F\}^n) = 2^n$

Let W be a wff containing at most atomic symbols $A_1, \ldots, A_n$

Want to define ~~$\phi(W)$~~ $\phi_W(\{T, F\}^n) \to \{T, F\}$

$\phi_W(\langle v_1, \ldots, v_n \rangle) = \bar{v}(W)$

where $v(A_i) = v_i$

$W = A_1 \vee A_2$

$\phi_W(\langle T,T\rangle) = T$

$\phi_W(\langle F,F\rangle) = F$

etc

Theorem Let $\Psi : \{T,F\}^n \to \{T,F\}$

There is a wff W containing at most symbols from $\{\neg, \vee, \wedge, A_1, \dots, A_n\}$ such that

$\phi_W = \Psi$

Proof: case (1) $\Psi(\langle v_1, \dots, v_n\rangle) = F \quad \forall \vec{v} \in \{T,F\}^n$

take $W = A_1 \wedge (\neg A_1)$

This works!

case (2) $\Psi(\vec{v}) = T$ for exactly one $\vec{v} \in \{T,F\}^n$

$\Psi(\langle w_1, \dots, w_n\rangle) = T$

Let $\beta_i = A_i$ if $w_i = T$

$\beta_i = (\neg A_i)$ if $w_i = F$

Let $W = \beta_1 \wedge \beta_2 \wedge \dots \wedge \beta_n \quad (\equiv ((\beta_1) \wedge (\beta_2) -)))$

$\bar{v}(W) = T \Leftrightarrow \bar{v}(\beta_i) = T \ \forall\ 1 \le i \le n$

choose β_i

$\bar{v}(\beta_i) = T \Leftrightarrow \bar{v}(A_i) = w_i$

$\bar{v}(W) = T \Leftrightarrow v(A_i) = w_i \ \forall\ 1 \le i \le n$

case (3) $\Psi(\vec{v}) = T$ for exactly k $\vec{v}$'s in $\{T,F\}^n$ $\quad k \geq 1$

Say there are

$\vec{v}_1 = \langle w_1^1, \dots, w_1^n\rangle$

$\vdots$

$\vec{v}_k = \langle w_k^1, \dots, w_k^n\rangle$

Let $\beta_i^j = A_j$ if $w_i^j = T$

$= \neg A_j$ if $w_i^j = F$ $\quad$ Let $W = \gamma_1 \vee \gamma_2 \vee \dots \gamma_k$

DNF contd: (copy from Neophytos)

- formula w/ n ~~vars~~ length has ($^n/\log n$) variables

Can write all wff using $\{\neg, \vee\}$, $\{\neg, \wedge\}$, $\{\neg, \rightarrow\}$

<u>Thm</u> Let P be a wff involving $\{A_1, \ldots, A_n\}$ and $\{\wedge, \neg, \vee\}$
Then $\exists$ a tautologically equiv wff Q using $\{A_1, \ldots A_n\}$ and $\{\neg, \vee\}$
i.e. $P \models\!\dashv Q$ (will write as $P \approx Q$)

<u>Proof</u>: Main Pt $A_1 \wedge A_2 \approx (\neg((\neg A_1) \vee (\neg A_2)))$
Just keep using $\nwarrow$ to get of ~~ands~~ ANDs
No problem w/ convergence
iff $\underset{10+1}{(A \Leftrightarrow B)}$ $\underset{10+10}{(A \to B) \wedge (B \to A)}$ diverges

Formally: By induction on length of wff P.
If P involves only $\{\wedge, \vee, \neg\}$ $\exists$ wff Q, $Q \approx P$
Q involves only $\{\neg, \vee\}$

case 1 P is A_i take $Q = A_i$
case 2 $P = (P_1 \vee P_2)$
By IH $\exists Q_1, Q_2$ w/ $P_1 \approx Q_1$ $P_2 = Q_2$

Q_1 Q_2 involve only $\{\vee, \neg\}$ Take $Q = (Q_1 \vee Q_2)$

case 3 $P = (\neg P_1)$
similar: left to you

case 4 $P = (P_1 \wedge P_2)$ By IH $P_1 \approx Q_1$ $P_2 \approx Q_2$
Q_1, Q_2 involve no $\wedge$s

Let $P = (\neg((\neg P_1) \wedge (\neg P_2)))$

QED

One says $\{\neg, \vee\}$ is a complete set of connectives (i.e. every wff is taut. equiv to one only using $\{\vee, \neg\}$)

Prop $\{\neg, \wedge\}$ is a complete set of propositional connectives

Pf $A_1 \vee A_2 \approx (\neg((\neg A_1) \wedge (\neg A_2)))$

Rest of Proof as Before

Prop $\{\neg, \rightarrow\}$ is a complete set

Proof $P \vee Q \equiv \neg P \rightarrow Q$

etc

Prop $\{\vee, \wedge\}$ is not a complete set of prop conn.

1. Show that any wff involving $\{\vee, \wedge\}$ is true when variables are set to T.
   (Easy, use induction)

2. $\neg A_1$ is F is $A_1 = T$

So $\neg A_1 \not\approx P$ for P involving only $\{\vee, \wedge\}$

QED

Prop $\{\neg\}$ is not complete

Easy to show by induction that any wff P involving only $\{\neg\}$ is equiv to either A_i or $(\neg A_i)$ & clearly no such formula is equiv to $(A_1 \vee A_2)$

QED

Problem 7/p38 Given a sequence of wffs
$\alpha_0, \alpha_1, \alpha_2, \ldots$

Define a map from $\bar{S} \to \bar{S}$
$\phi \to \phi^*$ as follows:
$(A_i)^* = \alpha_i$
$(\neg P)^* = (\neg P^*)$
$(P \alpha Q)^* = (P^* \alpha Q^*)$ $\alpha \in \{\vee, \wedge, \to, \leftrightarrow\}$

Claim: If $\bar{\phi}$ is a tautology, so is ϕ^*

Ex $((P \vee (\neg P))$ easy to check for tautology for any wff

Lemma: Let $v: S \to \{T, F\}$
Define $u: S \to \{T, F\}$
by $u(A_i) = \bar{v}(\alpha_i)$
Then $\bar{u}(\theta) = \bar{v}(\theta^*)$ for all wffs θ.

Proof of Lemma: (By induction on length of θ)
Case 1. θ is A_i
$\bar{u}(A_i) = u(A_i) = \bar{v}(\alpha_i)$
$= \bar{v}((A_i^*))$

Case 2 $\theta = (\neg \chi)$
$\bar{u}(\theta) = H_\neg(\bar{u}(\chi))$
$= H_\neg(\bar{v}(\chi^*))$
$= \bar{v}((\neg \chi^*))$
$= \bar{v}(\theta^*)$

Case 3 $\theta = (\chi \alpha \psi)$
α is a bin connective
$\bar{u}(\theta) = H_\alpha(\bar{u}(\chi), \bar{u}(\psi))$
$= H_\alpha(\bar{v}(\chi^*), \bar{v}(\psi^*))$
$= \bar{v}((\chi^* \alpha \psi^*))$
$= \bar{v}(\theta^*)$ QED

<u>Claim</u> If ϕ is a tautology so is ϕ^*

Let $v: S \to \{T, F\}$

To see $\bar{v}(\phi^*) = T$

Define u as before

so by Lemma

$\bar{v}(\phi^*) = \bar{u}(\phi)$

But ϕ is a tautology

so $\bar{u}(\phi) = T$

so $\bar{v}(\phi^*) = T$

<u>QED</u>

"Not a believer in the wisdom of suffering"

HW (next Fri) §2.1, 2.3 p78 problems 1, 2, 5, 8 p100 1, 2

First order Languages

Several new ingredients -

infinite ~~stock~~ of variables

$v_1, v_2, v_3, \ldots$

Quantifier Symbols

$\forall$ ~ "for all", "for every"

$\exists$ ~ "there is a", "there exists"

; when we actually define our formal language, we
; will "define" $\exists$ in terms of "$\forall$" "$\exists x$ —" abbreviates "$\neg \forall x \neg$ —"

Alphabet of a first order language

Logical Symbols

1. Parentheses: "(", ")"
2. variables: $v_1, v_2, v_3, \ldots$
3. Propositional connectives: $\neg$, $\rightarrow$
4. Two place predicate for $=$: $=$ ($\approx$)

Parameters

1. Quantifier: $\forall$
2. Predicate symbols: For each positive integer n will have a possibly empty set of n-ary predicate symbols.
3. A possibly empty set of constant symbols
4. For each positive n ~~on~~ a possibly empty set of n-ary functions symbols.

Two examples:

(1.) Language of set theory

One binary predicate:

$\in$

[Interpretation: variables range over sets
$a \in b$ ($\in ab$) means "a is a member of b"]

(2) <u>Number Theory</u> $[\omega = \{0,1,2,...\}]$

One binary predicate: $<$
One constant symbol: 0
One 1ary fn symbol S $[Sx = x+1]$
Three 2ary functions symbols
$+, \cdot, E$ (here xEy stands for x^y)

(i) There is no set of which every set is a member
(ii) $\neg$ [There is a set of which every set is a member]
(iii) $\neg$ [$\exists v_1$ (every set is a member of v_1)]
(iv) $\neg [\exists v_1 \forall v_2 (v_2 \in v_1)]$; replace $\exists$ by $\neg\forall\neg$

$v_2 \in v_1$ $\in v_2 v_1$

$(\neg(\neg(\forall v_1 (\neg \forall v_2 \in v_2 v_1))))$

<u>Pair Axiom</u>

For every pair of sets a, b, there is a set whose members are precisely a and b

$\forall v_1 v_2 \exists v_3$[members of v_3 are precisely v_1 and v_2]

$\forall v_1 \forall v_2 \exists v_3 \forall v_4 [v_4 \in v_3 \Leftrightarrow (v_4 = v_1 \vee v_4 = v_2)]$

etc

(See Enderton)

$(2+3)\circ 5$

$23+5\circ$ RPN

$\circ + 235$ PN

In our notation: no 5, 3, 2 ⇒ need S

$\circ + SS0SSS0SSSSS0$

$2+2=4$

$+22=4$

~~$= + SS\emptyset SS\emptyset = SSSS\emptyset$~~

$= + SS\emptyset SS\emptyset\ SSSS\emptyset$

(1) any non zero natural number is the success of some numb

(2) any number n, if $n \neq 0$ then n is the successor of some number

$\forall v_1(\neg(v_1=0) \longrightarrow (\exists v_2)(v_1 = S v_2))$: Polish : ∃ replaced

etc

Example P_x abbreviates x is prime

Solovay

M125a

Sept 29 '71
Wednesday

Alphabet of a first order logic
$v_0, v_1, v_2, \ldots$: variables
$\neg, \rightarrow$.
$=$
( )
$\forall$
n-ary predicate symbols $n \geq 1$
some (maybe $\emptyset$) n-ary function symbols $n \geq 1$
constant symbols

< 2-ary
$S, +, \cdot, E$
0

* everything in class
* in HW/Reading Asgn
* upto Fri
*/
§2.1 → Unique Readblty

Limited Quantifiers
$P(x)$: x is a prime
$Q(x)$: x is even

Every prime is even   irreducible commonsense
$(\forall x)(P(x) \rightarrow Q(x))$
There is a prime which is even
$(\exists x)(P(x) \wedge Q(x))$

Let L be a first order language
We now define (by a inductive defn) what a term is.
(i) Every variable is a term
(ii) Every constant symbol is a term
(iii) If f is an n-ary function symbol and $t_1, \ldots, t_n$ are terms then $ft_1 \ldots t_n$ is a term.
eg $\emptyset$ is term; $S\emptyset$ is a term; "$+SS0S0$" is a term
(iv) No terms except by rules (i) (ii) (iii)

Defn If P is an n-ary predicate symbol (could be =) and $t_1 \ldots t_n$ are terms then $Pt_1 \ldots t_n$ is an atomic formulas.

Examples $<S0SS0$ $(1<2)$
$=+SS0SS0\ SSSS0$ $(2+2=4)$

Inductive defn of wff follows -
(i) atomic formulae are wffs
(ii) if P is a wff, so is $(\neg P)$
(iii) if P, Q are wffs, so is $(P \to Q)$
(iv) if P is a wff and x is a variable then $\forall x\, P$ is a wff
(v) No others than by (1) – (4)

Structure
$\mathfrak{N} = \langle \omega; <, +, \cdot, E, S \rangle$
$< S0\,SS0$ is true on $\mathfrak{N}$
$< S0\,S0$ is false
$< x\, y$ ← not a sentence (∵ not bound)
$\forall x \forall y < x\, y$ is false (x and y are ~~not~~ bound by quant)

In
$(\forall x)\ (x=0) \to x=0$
First two occurances of x are bound
last is free

$(\exists x)(x=y) \to (x=y)$
Both y's are free in this formulae
also last occurance of x is free

We now formally define:
Variable x occurs free in the wff P
(By induction on length of P)
(i) Case 1 P atomic: x occurs free in $P \Leftrightarrow x$ occurs in P
(ii) Case 2 $P = (Q \to R)$ (Q, R wffs)
x occurs free in $P \Leftrightarrow x$ occurs free in Q or x occurs free in R

(iii) <u>case 3</u> P is $(\neg Q)$ Q a wff
x occurs free in $P \Leftrightarrow x$ occurs free in Q

(iv) <u>case 4</u> P is $\forall v_i Q$ Q a wff
x occurs free in $\forall v_i Q$
if ① $v_i \neq x$
② x occurs free in Q

$(\forall x)(x=0) \rightarrow x=y$

$x=0$ free x
$(\forall x)(x=0)$ no free occurances of x
$x=y$ free x (free y too)
$\therefore x$ is free in $(\forall x)(x=0) \rightarrow x=y$

Here is an inductive defn of which occurances of x in P are free (the others are said to be bound)

(1) All occurances of x in an atomic formulae are free
(2) Free occurances of x in $(P \rightarrow Q)$ are those free occurances of x in P and free occ. of x in Q
(3) Free occurances of x in $(\neg P)$ are precisely those free occurances of x in P.
(4) If $v_i = x$, x has no free occurances in $\forall v_i P$
If $v_i \neq x$, free occurances of x in $\forall v_i P$ are free occurances of x in P.

<u>Defn</u> A sentence of L is a formulae of L with no free variables

$a \to b$ $\quad a\bar{b}$

$a \vee \bar{b}$

$a \vee \bar{b}$

$\bar{a} \vee \bar{b}$

Abbreviations and Conventions: (cf Enderton)

$(\alpha \vee \beta)$	$((\neg\alpha) \to \beta)$
$(\alpha \wedge \beta)$	$(\neg(\alpha \to (\neg\beta)))$
$(\alpha \leftrightarrow \beta)$	$((\alpha \to \beta) \wedge (\beta \to \alpha))$
$\exists x\, \alpha$	$(\neg \forall x (\neg \alpha))$

For binary predicates like $\in, <, =$

$t_1 = t_2$ for "$= t_1 t_2$"

$t_1 \neq t_2$ for "$\neg = t_1 t_2$"

M125a

HW - due next ~~Friday~~ Monday: Enderton pp 95-97 prob 2, 8, 9, 12a, 12b, 19

Read §2.2 (compl)

Aziz

Friday 27 Sept 91

Ambiguityless rules for omitting (restoring) parentheses

1. Outermost set of parens may be dropped

   $\forall x\, \alpha \to \beta$ for $(\forall x\, \alpha \to \beta)$

2. Unary operators $\neg, \forall, \exists$ have as small a scope as possible

   $\neg \alpha \vee \beta$ means $(\neg \alpha) \vee \beta$ and not $\neg(\alpha \vee \beta)$

   $\forall x\, \alpha \to \beta$ means $(\forall x\, \alpha) \to \beta$ and not $\forall x\, (\alpha \to \beta)$

   $\exists x\, \alpha \wedge \beta$ mean $(\exists x\, \alpha) \wedge \beta$ not $\exists x\, (\alpha \wedge \beta)$

3. Rules for binary operations:

   $\wedge$ binds tighter than $\vee$ which binds tighter than $\to$ which is tighter than $\leftrightarrow$

4. Same operator several times associate to the right

   $P \vee Q \vee R$ means $(P \vee (Q \vee R))$; $P \to Q \to R$ means $(P \to (Q \to R))$

Ordered n-tuple If $a_1, \ldots, a_n$ are sets then $\langle a_1, \ldots, a_n \rangle$ is "ordered n tuple"

Key Property If $\langle a_1, \ldots a_n \rangle = \langle b_1, \ldots b_n \rangle$ then $a_1 = b_1 \ \ldots \ a_n = b_n$

Dispensible Blather: How to define $(a, b) = \{\{a\}, \{a, b\}\}$

$(a, b) = (c, d)$ iff $a = c$ & $b = d$

A function $f: X \to Y$ is a set of ordered pairs (1) $\ni \forall x \in X\ \exists y \in Y \ni (x,y) \in f$

(2) If $(x, y_1) \in f$ and $(x, y_2) \in f \to y_1 = y_2$

vector $n \in \omega$ is $\{0, 1, 2, \ldots, n-1\}$ $\quad 0 = \emptyset \quad 1 = \{0\} \quad 2 = \{0, 1\} \ldots$

$\langle a_0, a_1, \ldots a_{n-1} \rangle$ is the function f:

  $\mathrm{domain}(f) = \{0, 1, 2, \ldots n-1\} = n$

  $a_i = f(i)$

End of digression

If X is a set, X^n = set of all ordered n-tuples $\{\langle a_1, \ldots a_n \rangle\}$ s.t. all $a_i \in X$

very important!

Let L be a 1st order language. An L structure, $\mathfrak{a}$, consists of the following

(1) A nonempty set $|\mathfrak{a}|$

(2) Each n-ary predicate symbol, P, of L (other than "=") is assigned a subset $P_{\mathfrak{a}} \subseteq |\mathfrak{a}^n|$

(3) Each n-ary function symbol f is assigned an n-ary fn

  $f_{\mathfrak{a}}: |\mathfrak{a}|^n \to |\mathfrak{a}|$

(4) Each constant c of L is assigned an element $c_{\mathfrak{a}} \in |\mathfrak{a}|$

Example 1 L a language of no. theory

Symbols of $L = \{ \ldots, 0, S, +, \cdot, E, < \}$

Structure 1: $\mathfrak{N}$ $|\mathfrak{N}| = \omega = \{0, 1, 2, \ldots\}$; $0^0 = 1$ * of fns from set of size a to size b $\Rightarrow a^b$

$0_{\mathfrak{N}} = 0$ $\cdot_{\mathfrak{N}} = \cdot$

$+_{\mathfrak{N}} = +$ $E_{\mathfrak{N}}(a,b) = a^b$

Example 2 $L = \{ \ldots, 0, +, \cdot, < \}$

Structure: $|\mathbb{R}|$ = usual real numbers

~~Example 3~~ $0, +, \cdot, <$ are define as usual

Example 3: $L = \{ \ldots, \in \}$ language of set theory

$|\mathfrak{a}| = \omega$ $\in_{\mathfrak{a}}$ = usual < relation i.e. $\{ \langle m, n \rangle \mid m < n \}$

Let σ_1 be $\forall x \exists y (y \notin x)$

Is σ_1 true in $\mathfrak{a}$?

This amounts to asking:

For every $x \in \omega$ is there a y which is not $< x$.

Certainly there is: $x \nless x$

Let σ_2: singletons exist

$\forall x \exists y \forall z (z \in y \leftrightarrow z = x)$

claim σ_2 is False in $\mathfrak{a}$

For if we take $x = 1$ there is no y s.t. $\forall z \; z < y \leftrightarrow z = 1$

If σ is $\sigma_1 \to \sigma_2$

σ_1 a sentence can do

If $\sigma = \neg\sigma_1$, can figure out if σ is true in $\mathfrak{a}$ if we know σ_1 is true in $\mathfrak{a}$.

σ is $\forall x \sigma_1$ Now what?

Let $V = \{v_0, v_1, \ldots\}$ An assignment function is a map $S: V \to |\mathfrak{a}|$

We will define inductively:

For formulae ϕ

ϕ is true in $\mathfrak{a}$ relative to the assignment S.

$x < y$

Solovay

M125a - Lecture

Adnan Aziz

30 Sept 1991 (?) Monday

iT Fri Oct 4
Bring Blue Books
pen book (No open notes)

L First order Language
$\mathfrak{a}$ L structure

ioal - If σ is a sentence of L,
define precisely: "σ is true in $\mathfrak{a}$"
Cant use induction: because strip away $\forall$ hard?

$V = \{v_i \mid i \in \omega\}$
(= variables of L)

Defn - An assignment function is a map $s: V \to |\mathfrak{a}|$

-ubgoal - If ϕ is a wff and $s: V \to |\mathfrak{a}|$ is an assignment ~~for~~ fn, want to define $\mathfrak{a}$ satisfies ϕ wrt s.
(Tarski)

Sub subgoal - Let $T = \{t \mid t$ is a term of $L\}$
will define $\bar{s}: T \to |\mathfrak{a}|$

intuitively,
$\bar{s}(t)$ is the value of t under assignment s

sp.
hat will you
o in the next
ame of scrabble).

define $\bar{s}(t)$ by induction on length of t

(1) $\bar{s}(v) = s(v)$ for v a variable
(2) $\bar{s}(c) = c_{\mathfrak{a}}$ for c a constant symbol
(3) t is $f t_1 \ldots t_n$ for some n-ary fn symbol and terms $t_1, \ldots, t_n$

Set $\bar{s}(t) = f_{\mathfrak{a}}(\langle \bar{s}(t_1), \ldots, \bar{s}(t_n)\rangle)$; need unique readability to have only one way of [illegible]

Notation Let ϕ be a wff

$s : V \to |a|$ is an assignment function

Write $a \models \phi[s]$ means "a satisfies ϕ wrt s"

We now by induction on the length of ϕ define

$a \models \phi[s]$

Formally, one is defining for each ϕ, the set $S_\phi = \{s \mid s : V \to |a|,$ and $a \models \phi[s]\}$

Case 1 ϕ is $= t_1 t_2$ "snow is white" $\Leftrightarrow$ snow is white is true

Then $a \models \phi[s] \Leftrightarrow \bar{s}(t_1) = \bar{s}(t_2)$

Case 2 ϕ is $p t_1 t_2 \ldots t_n$ with p an n-ary predicate symbol (not "=")

then a yields $\phi[s]$ (i.e. $a \models \phi[s]$) $\Leftrightarrow$

$\langle \bar{s}(t_1), \ldots, \bar{s}(t_n) \rangle \in P^a$

Case 3 ϕ is $(\neg \psi)$

$a \models \phi[s] \Leftrightarrow$ it is not the case that $a \models \psi[s]$

Case 4 ϕ is $(\psi \to \theta)$, ψ and θ wff <Note impt of unique readability>

$a \models \phi[s] \Leftrightarrow$ either (i) $a \models \psi[s]$ is not true or (ii) $a \models \theta[s]$ or both

Defn Let $s : V \to |a|$

Let $x \in V$

Let $d \in |a|$

then $s(x|d)$ is the following assignment function

$s(x|d)(x) = d$

$s(x|d)(y) = s(y) \quad \forall y \neq x, y \in V$

case 5 ϕ is $\forall x \psi$

$a \models \phi[s] \Leftrightarrow$ for every $d \in |a|$, $a \models \psi[s(x|d)]$ End of Defn

Proposition - Let ϕ be a wff

Let $s_1 : V \to |a|$

$s_2 : V \to |a|$

be assignment functions

Suppose $s_1(v) = s_2(v)$ for all v occurring free in ϕ

$a \models \phi[s_1] \Leftrightarrow a \models \phi[s_2]$

Proof - By induction on length of ϕ

case 1 ~~By induction on length of ϕ~~

ϕ is $= t_1 t_2$

Since s_1, s_2 agree on variables appearing in t_1, t_2

$\bar{s}_1(t_1) = \bar{s}_2(t_1)$
$\bar{s}_1(t_2) = \bar{s}_2(t_2)$

$\bar{s}(t_1) = \bar{s}(t_2)$ (Prove by induction on t that $\bar{s}_1(t) = \bar{s}_2(t)$ if s_1, s_2 agree on variables in t.)

Case 2-4 Left to you

case 5 ϕ is "$\forall x \psi$"

variables free in ψ are included among variables of $\psi \cup \{x\}$

So $s_1(x|d)$ and $s_2(x|d)$ agree on all variables occurring free in ψ

(for any $d \in |a|$)

So $a \models \phi[s_1] \Leftrightarrow (\forall d \in |a|)\ a \models \psi[s_1(x|d)] \Leftrightarrow$ (by IH)

$a \models \psi[s_2(x|d)]$ for all $d \in |a|$

$\Leftrightarrow a \models \phi(s_2)$

Cor - Let σ be a sentence of L (i.e. σ is a wff w/ no free variables)

then either (1) $a \models \sigma[s]$ for all assignment functions s

or (2) $a \models \sigma[s]$ for no assignment functions s

Oct 1991 EECS 290h ADNAN
Tuesday AZIZ

Oct 1991 M125a ADNAN
Wednesday AZIZ

L (FOL) a (structure)

Let σ be a sentence of L,
we defined "σ is true in a" (inductively)

$a \models \varphi(s)$

Alternate treatment:

Skeleton of alternate treatment

Step 1 - Enlarge L to new 1st order language L_a
by adding a new constant symbol $\underline{x}$ for each $x \in |a|$

Step 2a - Assign in the obvious way to each term t of L_a without variables its value in a, $a(t)$

Step 2b - Define by induction on length of φ (for φ a sentence of L_a) the truth value of φ in a. ($a(\varphi)$)

Eg case 1 φ is "$= t_1 t_2$"

$a(\varphi) = T \Leftrightarrow a(t_1) = a(t_2)$

case 5 φ is $\forall x \Psi$

$a(\varphi) = T \Leftrightarrow \forall d \in |a| \ a(\Psi_x[\underline{d}]) = T$

Here $\Psi_x[\underline{d}]$ is obtained from Ψ by replacing all free occurances of x in Ψ by $\underline{d}$

▫ End of sketch

/* Tarski chose this approach ∵ he didnt want to
* deal with uncountable languages
*/

Defn

Let φ be an L-formula

Let $\mathfrak{a}$ be an L-structure

Then φ is valid in $\mathfrak{a}$ $\Leftrightarrow$

$\forall s: V \to |\mathfrak{a}|$, $\mathfrak{a} \models \varphi[s]$

Example: $\mathfrak{N} = \langle \omega, +, \cdot, \ldots \rangle$

"$x+y = y+x$" is valid in $\mathfrak{N}$

Defn: φ is valid (logically valid)

if for every L structure $\mathfrak{a}$, φ is valid in $\mathfrak{a}$

Defn: Let Σ a set a formulas φ = formula (all from L)

Then $\Sigma \models \varphi$ (à la Enderton)

$\Leftrightarrow$ for every L-structure $\mathfrak{a}$ and every assignment function $s: V \to |\mathfrak{a}|$,

if $(\forall \theta \in \Sigma)(\mathfrak{a} \models \theta[s])$ then $\mathfrak{a} \models \varphi[s]$

Defn $\Sigma \models_s \varphi$ if for every structure $\mathfrak{a}$ if each $\theta \in \Sigma$ is valid in $\mathfrak{a}$, then φ is valid in $\mathfrak{a}$.

Examples $Px \models_s Py$ schoenfield

$Px \not\models Py$ (Eg Px: x is even $s(x)=0$ $s(y)=1$)

Defn Let Σ be a set of sentences

Let $\mathfrak{a}$ be a structure

$\mathfrak{a}$ is a model ~~for~~ of Σ if every sentence in Σ is true in $\mathfrak{a}$

Prop Let Σ a set of sentences. Let φ be a formula. Then TFAE

1) $\Sigma \models \varphi$

2) $\Sigma \models_s \varphi$

3) φ is valid in every model of Σ

Proof - Defn pushing

You should show 1) $\to$ 3)

"Theory" I just mean a set of sentences

example L: I, e, $\circ$

1 ary operation, const symbol, binary connective

Let Gr be following theory in this language

(1) $(\forall x)\ (e\cdot x = x\cdot e = x)$

(2) $(\forall x)\ (x\cdot Ix = Ix\cdot x = e)$

(3) $(\forall x)(\forall y)(\forall z)\ (x\cdot(y\cdot z) = (x\cdot y)\cdot z)$

- Models of Gr are group
- Similarly there ~~are~~ is a theory whose models are commutative Rings
- No such theory for topological spaces

Next Goal

Give defn of a "formal proof" from Σ

(1) A proof is a finite sequence of wffs

(2) If we have an algorithm for which formulas are in Σ, we can tell effectively what the proofs from Σ are.

Here is a provisional defn of proof from Σ

We will define a set Λ of logical axioms

$\Sigma \vdash \varphi \iff$ there is a finite sequence of formulas $\langle \varphi_1, \dots, \varphi_n \rangle$:

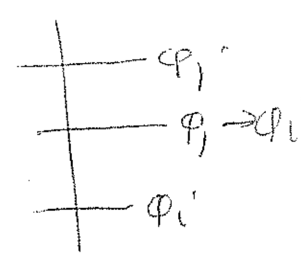
(1) $\varphi_n = \varphi$

(2) For all $1 \le i \le n$ either (a) $\varphi_i \in \Sigma$

or (b) $\varphi_i \in \Lambda$

or (c) $\exists j < i\ \exists k < i$

$\varphi_k = \varphi_j \to \varphi_i$



TPT $x=y,\ y=z \vdash x=z$

$\vdash (x=y) \to (y=z) \to (x=z)$

$a \to b \to c \quad = \bar{a} + \bar{b} + c \ \checkmark$

$a \to (b \to c) = \bar{a} + \bar{b} + c \ \checkmark$

$\vdash x=y \to \forall z P x z \to \forall z P y z$

$\vdash x=y \to P x z \to P y z$

$x=y \vdash P x z \to P y z$

$x=y \vdash \forall z (P x z \to P y z)$

$x=y \vdash \forall z (P x z \to P y z) \to \forall z P x z \to \forall z P y z)$

$x=y \vdash (\forall z P x z \to \forall z P y z)$

$\vdash x=y \to (\forall z P x z \to \forall z P y z)$

7 Oct '91
Monday

Γ - set of formulae

$\Gamma \vdash \varphi \iff \Gamma \models \varphi$

(φ is a theorem from axioms of Γ)

<u>Thm</u> If φ, ψ are formulae of L, then

ψ is a generalization of φ

if ψ is $\forall x_1 \dots \forall x_n \varphi$ ($n=0$ is legal, $x_1, \dots x_n$ not all distinct)

Logical axioms are all generalizations of formulae of the following types (x, y are variables, α and β are wffs)

1. All tautologies
2. All formulas of the form $\forall x\, \alpha \to \alpha^x_t$ (where t is substitutable for x in α)

/* t is a term! shouldn't have other vars in it which are being quantified */

3. $\forall x (\alpha \to \beta) \to (\forall x \alpha \to \forall x \beta)$
4. $\alpha \to \forall x \alpha$ provided x does not occur free in α
5. $x = x$
6. $x = y \to (\alpha \to \alpha')$ where α is atomic and α' is obtained from α by replacing some x's by y's

α is a wff

Idea α^x_t is obtained from α by replacing every free occurance of x by t

α $(\exists x)(x=4)$ or $x=5$

$t = y^2$

α^t_x $(\exists x)(x=4)$ or $(y^2=5)$

Define α^x_t formally by induction on length of α

(1) α atomic

just replace x by t throughout α to get α^x_t

(2) $\alpha = (\neg\beta)$

$\alpha^x_t = (\neg\beta^x_t)$

(3) $\alpha = (\beta \to \gamma)$ (β, γ wff)

$\alpha^x_t = (\beta^x_t \to \gamma^x_t)$

(4) α is $\forall y \beta$

case 4a $y = x$
Then α^x_t is α (no free occurances of x)

case 4b $y \neq x$
$\alpha^x_t = \forall y\, \beta^x_t$

α is $\neg \forall y(x=y)$

~~$\forall/\alpha/\alpha$~~ $\forall x\, \alpha \to \alpha^x_z$

$\forall x\, \neg\forall y\,(x=y) \to \neg\forall y\,(z=y)$ o.k.

$\forall x\, \alpha \to \alpha^x_y$

$\forall x\, \neg\forall y(x=y) \to \neg\forall y(y=y)$

is false in any model $\mathfrak{a}$ such that $|\mathfrak{a}|$ has at least two elements.

Here is an inductive defn of t is substitutable for x in α.

1. α is atomic. Any t is substitutable for x in α

2. $\alpha = (\neg\beta)$ is substitutable for x in α $\Leftrightarrow$ t is substitutable for x in β

3. $\alpha = (\beta \to \gamma)$ (β, γ wffs)

t is substitutable for x in α iff

(1) t is sub for x in β

and (2) t is subs for x in γ

4. α is $\forall y\, \beta$

t is substitutable for x in α if either

(i) α has no free occurrences of x

or (ii) y does not appear in t and ~~t~~ is substitutable for ~~t~~x in β

Tautologys

A prime formula is either an atomic formula or one of the form $\exists x \beta$

~~$\forall x \beta$~~ $Pr = \{\varphi \mid \varphi \text{ is a prime formulas}\}$

$wff = \{\varphi \mid \varphi \text{ is a wff}\}$

Like what we did before.

any $v : Pr \to \{T, F\}$

prolongs to $\bar{v} : wff \to \{T, F\}$

$\bar{v}(\varphi) = v(\varphi)$ if φ is prime

$\bar{v}((\neg\varphi)) = H_\neg(\bar{v}(\varphi))$

$\bar{v}((\alpha \to \beta)) = H_\to(\bar{v}(\alpha), \bar{v}(\beta))$

Let $\varphi \in wff$

φ is a tautology if for any $v : Pr \to \{T, F\}$,

$\bar{v}(\varphi) = T$

Recall: A Γ proof is a finite sequence of formulas

$\varphi_1, \ldots, \varphi_n$:

Each φ_i is either (a) a formula in Γ

(b) a logical axiom (Λ = logical axioms)

(c) there are $j < i$, $k < i$:

φ_k is $\varphi_j \to \varphi_i$

(Modus ponens)

$$\frac{\begin{matrix}\varphi_j \\ \varphi_j \to \varphi_i\end{matrix}}{\varphi_i}$$

φ is a Γ thm if there is a Γ process whose last line ~~is~~ is φ

Suppose

Suppose S is a set of wff

suppose also

1) $\Gamma \subseteq S$

2) $\Delta \subseteq S$

3) if φ, $\varphi \to \psi$ are in S

then ~~$\varphi \in S$~~ $\psi \in S$

Then if $\Gamma \vdash \varphi$, ~~$\psi \in$~~ $\varphi \in S$

<u>Proof</u> Let $\varphi_1, \ldots, \varphi_n$ be a Γ proof of φ

By induction, show $\varphi_i \in S$

so can prove facts about thms of P using this approach

M125a

Oct 8 '11
Wednesday

<u>Goal</u> $\Gamma \models \varphi \iff \Gamma \vdash \varphi$

$\uparrow$ using Γ sentences, axioms, modus ponens to write proof

"metaphysics" = "nonsense"
"thats my punishment for being poor and having to teach this course" $\Leftarrow$ complete Polish notation solution

<u>Generalization Theorem</u>

suppose $\Gamma \vdash \varphi$
suppose x doesnt occur free in any formulae Γ

Then $\Gamma \vdash \forall x\, \varphi$ (x <u>can</u> occur free in φ)

we'll prove this by induction on Γ-theorems
Need to check 3 things
(1) Thm is true if $\varphi \in \Gamma$
(2) Thm is true if $\varphi \in \Lambda$
(3) If thm true for φ, and $\varphi \to \psi$, its true for ψ

Case (i) $\varphi \in \Gamma$ Notice x is not free in φ

so $\Gamma \vdash \varphi$

$\Gamma \vdash \varphi \to \forall x\, \varphi$ (logical axiom) x is not free in φ

so $\Gamma \vdash \forall x\, \varphi$ (modus ponens)

case (ii)
$\varphi \in \Lambda$
Then clearly, ~~XXXX~~ $\forall x\, \varphi$ is in Λ
(its a generalization of the same formula, φ was a generalization) so $\Gamma \vdash \forall x\, \varphi$

Case (iii) $\Gamma \vdash \varphi$

$\Gamma \vdash \varphi \to \psi$

and we know that for φ, $\varphi \to \psi$

To see: $\Gamma \vdash \forall x \psi$

(1) $\Gamma \vdash \forall x \varphi$
(2) $\Gamma \vdash \forall x (\varphi \to \psi)$ } Ind Hyp

(3) $\Gamma \vdash (\forall x)(\varphi \to \psi) \to [\forall x \varphi \to \forall x \psi]$ (logical axiom)

(4) $\Gamma \vdash \forall x \varphi \to \forall x \psi$ (MP)

$\Gamma \vdash \forall x \psi$ (MP applied to (1), (4))

Defn Let $\alpha_1, \ldots, \alpha_n, \beta$ be wffs

β is a tautological consequence of $\alpha_1, \ldots, \alpha_n \iff$

$\alpha_1 \to \alpha_2 \to \cdots \to \alpha_n \to \beta$ is a tautology

(Rule T)
Prop If Γ proves $\alpha_1, \ldots,$ i.e. $\Gamma \vdash \alpha_1, \ldots$ $\Gamma \vdash \alpha_n$ and β is a taut. consequence of $\alpha_1, \ldots, \alpha_n$ then $\Gamma \vdash \beta$

Recall $\alpha_1 \to \alpha_2 \to \cdots \to \alpha_k \to \beta$ is $\alpha_1 \to (\alpha_2 \to (\cdots (\alpha_k \to \beta))))$

Proof $\Gamma \vdash \alpha_1 \to \cdots \to \beta$ (Taut)

$\Gamma \vdash \alpha_1$

$\Gamma \vdash \alpha_2 \to \cdots \alpha_n \to \beta$ (MP)

$\Gamma \vdash \alpha_2$

$\Gamma \vdash \alpha_3 \cdots \to \beta$)

Deduction theorem: If $\Gamma ; \alpha \vdash \beta$

then $\Gamma \vdash \alpha \to \beta$

Proof By induction on the $\Gamma ; \alpha$ proof of β

case(i) $\beta \in \Gamma \cup \Delta$

then certainly $\Gamma \vdash \beta$

But $\beta \to (\alpha \to \beta)$ is a tautology

So $\Gamma \vdash (\alpha \to \beta)$ (T)

case(ii) $\beta = \alpha$

To see

$\Gamma \vdash \alpha \to \alpha$

But $\alpha \to \alpha$ is a tautology

case(iii) $\Gamma ; \alpha \vdash \varphi$ To see $\Gamma \vdash \alpha \to \psi$

$\Gamma ; \alpha \vdash \varphi \to \psi$

and we know by ind hyp

$\Gamma \vdash \alpha \to \varphi$

$\Gamma \vdash \alpha \to (\varphi \to \psi)$

claim $(\alpha \to \varphi) \to (\alpha \to (\varphi \to \psi)) \to (\alpha \to \psi)$

is a tautology

Granted this, $\Gamma \vdash \alpha \to \psi$ by rule T

Lemma

If $\Gamma ; \varphi \vdash \neg \psi$

$\Gamma ; \psi \vdash \neg \varphi$

Proof $\Gamma ; \varphi \vdash \neg \psi$

$\Gamma \vdash \varphi \to \neg \psi$ (deduction)

$\Gamma \vdash \psi \to \neg \varphi$ (T) TMP

~~XXX~~ $\Gamma ; \psi \vdash \neg \varphi$ (MP)

Defn Γ is inconsistent if for some β,

$\Gamma \vdash \beta$, $\Gamma \vdash \neg \beta$

Prop If Γ is inconsistent

$\Gamma \vdash \alpha$ (any α)

Proof claim

$\beta \to (\neg \beta \to \alpha)$ is a tautology

so if $\Gamma \vdash \beta$

$\Gamma \vdash \neg \beta$

$\Gamma \vdash \alpha$ (by T)

$\vdash \exists x \forall y\, Pxy \to \forall y \exists x\, Pxy$

$\exists$: Player has winning move

$\forall$: Player has all losing moves

M125a

Oct 11, 1991
Friday

<u>Thm</u> $\vdash \exists x \forall y\, \varphi \to \forall y \exists x\, \varphi$

E.T.S. – "It's enough to see"

By ~~DRP~~ Deduction Thm E.T.S.

$\exists x \forall y\, \varphi \vdash \forall y \exists x\, \varphi$

By generalization ~~the~~ rule,

E.T.S. $\exists x \forall y\, \varphi \vdash \exists x\, \varphi$

use definition of $\exists$:

$\neg \forall x \neg \forall y\, \varphi \vdash \neg \forall x \neg \varphi$

$P \vdash \neg Q$

iff $\vdash P \to \neg Q$

Recall T: $\vdash (P \to \neg Q) \to (Q \to \neg P)$

$\therefore$ By MP $P \vdash \neg Q$ iff $Q \vdash \neg P$

$P \vdash \neg Q$ $\qquad$ $\neg P \vdash \neg Q$

then $Q \vdash \neg P$ $\qquad$ then $Q \vdash \neg\neg P$

i.e. $Q \vdash P$

ETS $\forall x \neg \varphi \vdash \neg\neg \forall x \neg \forall y\, \varphi$

ETS $\forall x \neg \varphi \vdash \forall x \neg \forall y\, \varphi$

by generalization

$\forall x \neg \varphi \vdash \neg \forall y\, \varphi$

ets $\forall x \neg \varphi, \forall y\, \varphi \vdash 0=1$

~~$P \vdash \neg\varphi$~~

$$\boxed{\begin{array}{l} P \vdash \neg\varphi \\ \text{ets. } P, \varphi \vdash 0=1 \end{array}}$$

$P, Q \vdash 0=1$

$\Rightarrow P, Q \vdash$

Claim: But $\forall x \neg \varphi \vdash \neg \varphi$

Proof $\vdash \forall x \neg \varphi \to \neg(\varphi)$ (Axiom)

$\forall x \neg \varphi \vdash \forall x \neg \varphi$ (Gen)

$\forall x \neg \varphi \vdash \neg \varphi$ (MP)

Similarly $\forall y\, \varphi \vdash \varphi$

$\forall x \neg \varphi, \forall y \varphi \vdash \varphi, \neg \varphi$ $\qquad$ $(\forall x) \neg \varphi, \forall y\, \varphi \vdash 0=1$

Need one Lemma:

If $\Gamma, \varphi \vdash \psi$ $\qquad$ $\Gamma \vdash \varphi \to \psi$

$\Gamma, \varphi \vdash \neg\psi$ $\qquad$ $\Gamma \vdash \varphi \to \neg\psi$

But $(\varphi \to \psi) \to (\varphi \to \neg\psi) \to \neg\varphi$

then

$\Gamma \vdash \neg\varphi$

$\therefore\ \Gamma \vdash \neg\varphi$

Gödel Completeness Theorem $\Gamma \models \varphi \leftrightarrow \Gamma \vdash \varphi$

Thm $\Gamma \vdash \varphi$

Suppose c is a constant appearing in φ,
but doesn't appear in any formula in Γ.
Then there is a variable y:

$\Gamma \vdash (\forall y)\ \varphi^c_y$

Proof Let $\alpha_1, \ldots, \alpha_n$ is a proof of φ from Γ
Let y be a variable not appearing in this proof

Claim $(\alpha_1)^c_y, (\alpha_2)^c_y, \ldots, (\alpha_n)^c_y$ is a proof of φ^c_y

case (i) $\alpha_i \in \Gamma$ $\quad$ c doesn't occur in α_i,
But then since
then $(\alpha_i)^c_y = \alpha_i$

$\therefore\ (\alpha_i)^c_y \in \Gamma$ so ok

case (ii) $\alpha_i \in \Lambda$

Then a routine check (using y as a fresh variable)
shows $(\alpha_i)^c_y \in \Lambda$

case (iii) For some $j < i$, $k < i$

α_k is $\alpha_j \to \alpha_i$

$(\alpha_k)^c_y = (\alpha_j)^c_y \to (\alpha_i)^c_y$

so $(\alpha_i)^c_y$ follows from $(\alpha_j)^c_y$, $(\alpha_k)^c_y$ by M.P.

Let Φ be set of axioms from Γ used in $\langle \alpha_1, \ldots, \alpha_n \rangle$

$\Phi \vdash (\varphi)^c_y$

But y doesnt appear free in Φ

By generalization,

$\Phi \vdash \forall y \, (\varphi)^c_y$

Corollary Suppose $\Gamma \vdash \varphi^x_c$

c doesnt appear in either Γ or φ

then $\Gamma \vdash \forall x \, \varphi$

By result just proved

for some fresh variable y not appearing in φ,

$\Gamma \vdash (\forall y)(\varphi^x_c)^c_y$ $\cancel{(\forall x)(\varphi^x_y)}$; ~~$\Gamma \vdash \varphi^x_c$ but Result $\Rightarrow \Gamma \vdash \forall x \varphi^x_c$~~

Clearly since c is not in φ

$(\varphi^x_c)^c_y = \varphi^x_y$

so $\Gamma \vdash (\forall y)(\varphi)^x_y$

Lemma

but y not appear in φ

$$(\forall y)\ \varphi^x_y \vdash \forall x\, \varphi$$

$$(\forall y)\varphi^x_y \vdash (\forall y)(\varphi^x_y)$$

$$\vdash (\forall y)(\varphi)^x_y \rightarrow ((\varphi)^x_y)^y_x$$

(Notice x is substitutable for y in $(\varphi)^x_y$)

$$(\varphi^x_y)^y_x = \varphi$$

since y was fresh

$$(\forall y)\ \varphi^x_y \vdash \varphi \quad \text{(M.P.)}$$

$$(\forall y)\ \varphi^x_y \vdash \forall x\, \varphi \quad \text{(Gen)}$$

So $\Gamma \vdash (\forall y)(\varphi)^x_y$

$$\vdash (\forall y)\ \varphi^x_y \rightarrow \forall x\ \varphi \quad \text{(deduction Thm + Lemma)}$$

$$\Gamma, \exists x\ \Phi(x) \vdash \sigma$$

So $\Gamma \vdash (\forall y)\ \varphi^c_y$

$$\Gamma, \Phi(c) \vdash \sigma$$

$$\Gamma, \exists x\ \Phi \vdash \sigma$$

$\varphi = (x = c)$ $\quad \psi = x = c$

$\varphi^x_c = (c = c)$ true

$(y = b)^y_c \vdash (b = c)$

but $\exists y\ (y = b) \nvdash (b = c)$

Existential Introduction (EI)

$$\Gamma, (\varphi)^x_c \vdash \psi$$

and suppose c doesn't appear in Γ, φ, ψ

then

$$\Gamma, \exists x \varphi \vdash \psi$$

By deduction thm $\quad \Gamma \vdash (\varphi)^x_c \rightarrow \psi$

$(P \to Q) \leftrightarrow (\neg Q \to \neg P)$

$\Gamma \vdash \neg \psi \to \neg (\varphi)^x_c$

$\Gamma, \neg \psi \vdash \neg (\varphi)^x_c$

$\Gamma, \neg \psi \vdash (\forall x) \neg (\varphi)^x_x$

$\therefore \Gamma, \neg (\forall x) \neg (\varphi)^x_x \vdash \psi$

$\therefore \Gamma, \exists x\ \varphi \vdash \psi$

done

R Solovay — Math 125a Lecture — 14 Oct '91 Monday

Due at 21
3,4/91 ; 10,15/123
1,2/139

Going to have proof of variant theorem for you to read in Enderton.

Next Goal L - FOL
Γ - set of formulae
φ an L formulae

If $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$ (Easy dirn of Godel Compl Thm)

Recall: $\Gamma \models \varphi$ for every L structure $\mathfrak{a}$, and every map $s : Var \to |\mathfrak{a}|$,
if for every $\psi \in \Gamma$ $\mathfrak{a} \models \psi[s]$
then $\mathfrak{a} \models \varphi[s]$

Theorem If $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$

Proof Use the following lemma:

Lemma If φ is a logical axiom then $\phi \models \varphi$ (ie φ is valid in all structures)

Proof of thm modulo lemma: (ie assuming lemma)

Proof by induction on proof of φ

case 1: $\varphi \in \Gamma$ Trivial

case 2: $\varphi \in \Lambda$ Immediate from lemma

case 3: φ follows from earlier lines of proof say ψ, $\psi \Rightarrow \varphi$ by modus ponens

Let $\mathfrak{a}$ be an L-structure

Let $s : \text{var} \longrightarrow |\mathfrak{a}|$

To see if for all $\theta \in \Gamma$, $\mathfrak{a} \models \theta[s]$ then $\mathfrak{a} \models \varphi[s]$

Assume for all $\theta \in \Gamma$, $\mathfrak{a} \models \theta[s]$

To see ~~...~~ is $\mathfrak{a} \models \varphi[s]$

By IH $\mathfrak{a} \models \psi[s]$ (*)

and $\mathfrak{a} \models \psi \to \varphi[s]$

But $\mathfrak{a} \models (\psi \to \varphi)[s]$

iff $\mathfrak{a} \not\models \psi[s]$ or $\mathfrak{a} \models \varphi[s]$

But $\mathfrak{a} \models \psi[s]$ by (*)

so $\mathfrak{a} \models \varphi[s]$

QED (case 3)

Remains to prove lemma:

Remark 1: If φ is valid in $\mathfrak{a}$, so is $\forall x\, \varphi$

Recall φ is valid in $\mathfrak{a}$ iff $\forall s: Var \to |\mathfrak{a}|$, $\mathfrak{a} \models \varphi[s]$

Proof: /* Assume φ is valid in $\mathfrak{a}$ */ Let $s: Var \to |\mathfrak{a}|$

~~Then~~ To see $\mathfrak{a} \models \forall x\, \varphi[s]$

i.e. need to see $\forall d \in |\mathfrak{a}|$

$$\mathfrak{a} \models \varphi[s(x|d)]$$

But φ is valid

So for any $t: Var \to |\mathfrak{a}|$,

$$\mathfrak{a} \models \varphi[t]$$

In particular, for any d, $\mathfrak{a} \models \varphi[s(x|d)]$

Cor If φ is logically valid, so is any generalization of φ

So it remains to check for each of six types of basic logical axioms that they are logically true

"Principle of Procrastination" — leave the hardest for the last

case 4 $\alpha \to \forall x\, \alpha$ provided α is a wff and x is not free in α

Left to you ☺

<u>case 3</u> $\forall x(\alpha \to \beta) \to \forall x \alpha \to \forall x \beta$

Left to you

<u>case 5</u> $x = x$

Let $\mathfrak{a}$ be a structure

Let $s : \text{var} \to |\mathfrak{a}|$

To see $\mathfrak{a} \models x = x\,[s]$?

i.e. $s(x) = s(x)$ Clear

Digression - Earthquakes, MSRI, Nephew

<u>Axiom Group 1</u> Tautologys

Let $\mathfrak{a}$ be a structure. Let φ be a tautology

Let $s : \text{var} \to |\mathfrak{a}|$

To see $\mathfrak{a} \models \varphi[s]$

Define v: Prime Formula $\to \{T, F\}$

$v(\pi) = T$ iff $\mathfrak{a} \models \pi[s]$

$= F$ otherwise

Easy to check: for any formula θ,

$\bar{v}(\theta) = T$ iff $\mathfrak{a} \models \theta[s]$

$= F$ otherwise

In particular $\mathfrak{a} \models \varphi[s]$ iff $\bar{v}(\varphi) = T$

But φ is a tautology so $\bar{v}(\varphi) = T$

so $\mathfrak{a} \models \varphi[s]$

Case 6 $\quad x=y \to (\alpha \to \alpha')$

α is atomic

α' is obtained from α by replacing some of the x's in α by y's

Proof Enough to see

$\{x=y, \alpha\} \models \alpha'$

Lemma Let t be a term. Let t' be a term obtained from t by replacing some x's by y's

Then if $s(x) = s(y)$ where $s: \text{var} \to |a|$

then $\bar{s}(t) = \bar{s}(t')$

Proof Easy induction on t. (length of t)

~~α(1)~~

$\alpha = t_1 t_2$

$\alpha' = t_1' t_2'$

Let $s: \text{var} \to |a|$

$a \models (x=y)[s] \quad (1)$

$a \models \alpha[s] \quad (2)$

(1) says $s(x) = s(y)$

By lemma,

$\bar{s}(t_1) = \bar{s}(t_1')$

$\bar{s}(t_2) = \bar{s}(t_2')$

(2) $\bar{s}(t_1) = \bar{s}_1(t_2)$

So $\bar{s}(t_1') = \bar{s}(t_2')$

so $q \models \alpha'[s]$

—— x ——

Case when $\alpha = P\, t_1 \ldots t_n$
is entirely similar

—— x ——

Axiom Group 2

~~First~~ Finally, case 2

$\forall x\, \varphi \to (\varphi)^x_t$ where t is substitutable for x in

Oct 16 '91
Wednesday

proving

If $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$

Reduced this to a lemma:

If φ is a logical axiom
$\mathfrak{a}$ an L-structure
$s : V \to |\mathfrak{a}|$
then $\mathfrak{a} \models \varphi[s]$

Done most of lemma

Remains to see: if α is a wff and t is substitutable for x in α then $\forall x \alpha \to \alpha^x_t$ is logically valid

To motivate needed lemma, look at special case

$\forall x Px \to Pt$

Let $\mathfrak{a}$ an L structure
$s : V \to |\mathfrak{a}|$

Need to see

$\mathfrak{a} \models (\forall x Px \to Pt)[s]$

"unwrapping the defns"

Need to see
if $\mathfrak{a} \models \forall x Px [s]$
then $\mathfrak{a} \models Pt[s]$

$\mathfrak{a} \models Pt[s]$ iff $\bar{s}(t) \in P^{\mathfrak{a}}$

But $\mathfrak{a} \models \forall x\, Px[s]$

so $\mathfrak{a} \models Px[s(x|\bar{s}(t))]$

ie $\bar{s}(t) \in P_a$ which is what we used

<u>Lemma</u> Let t substitutable for x in φ

$\mathfrak{a} \models \varphi^x_t [s]$ iff

$\mathfrak{a} \models \varphi[s(x|\bar{s}(t))]$

We will prove the lemma by induction of φ

For atomic case need a sublemma as follows

<u>Sublemma</u> Let u be a term

Let $s: V \to |\mathfrak{a}|$

$\bar{s}(u^x_t) = \overline{s(x|\bar{s}(t))}\,(u)$

Proof of Sublemma: induction on u

<u>case 1</u> u is a constant symbol or variable $\neq x$

so $u^x_t = u$ (since x does not appear)

$\bar{s}(u^x_t) = \bar{s}(u)$

on the other hand,

$\overline{s(x|d)}\,(u) = \bar{s}(u)$

(since x not in u!)

$\overline{s(x|\bar{s}(t))}\,(u) = \bar{s}(u)$

—x— /* case 1 */

<u>case 2</u> $u = x$,

$u^x_t = t$

so $\bar{s}(u^x_t) = \bar{s}(t)$

$\overline{s(x|\bar{s}(t))}(u) = \bar{s}(t)$

so qed case 2

<u>case 3</u> $u = f t_1 \ldots t_n$

Left to you

Use I.H.

QED (sublemma)

<u>Proof of lemma</u>

<u>Case 1</u> φ atomic

Look at case $\varphi = P u$

$\mathfrak{a} \models \varphi^x_t [s]$ iff

$\mathfrak{a} \models (Pu)^x_t [s]$ iff

$\mathfrak{a} \models P u^x_t [s]$ iff

$\bar{s}(u^x_t) \in P^{\mathfrak{a}}$

But by the lemma $\bar{s}(u^x_t) = \overline{s(x|\bar{s}(t))}(u)$

iff $\overline{s(x|\bar{s}(t))}(u) \in P^{\mathfrak{a}}$

iff $\mathfrak{a} \models P u [s(x|\bar{s}(t))]$

(* did for 1-ary predicates, same proof for n-ary predicates ... subscripts *)

case 2 φ is $\neg\psi$

Left to you (use IH)

case 3 φ is $\psi \to \chi$

Left to you (use IH)

case 4 φ is $\forall y\, \psi$ (includes $x=y$!)

and x is not free in φ

so φ^x_t is just φ

$\mathfrak{a} \models \varphi^x_t\,[s]$ iff

$\mathfrak{a} \models \varphi[s]$

since x not free in φ

$\mathfrak{a} \models \varphi[s]$ iff $\mathfrak{a} \models \varphi[s(x|d)]$ (any $d \in |\mathfrak{a}|$)

so $\mathfrak{a} \models \varphi[s(x|\bar{s}(t))]$ iff $\mathfrak{a} \models \varphi[s]$

so QED case 4

case 5 φ is $\forall y\, \psi$

& x does occur free in φ

since x occurs free in φ

and t is substitutable for x in φ

(i) y does not occur in t

(ii) $y \neq x$

By (1) $\quad \bar{s}(t) = \overline{s(y|d)}(t)$ for any $d \in |a|$ $\quad (*)$

$a \models \varphi^x_t [s]$ iff

$\forall d \in |a|$

$a \models \psi^x_t [s(y|d)]$

$a \models \psi^x_t [s(y|d)] \quad (\forall d \in |a|)$

$a \models \psi[s(y|d)(x|\overline{s(y|d)}(t))]$; IH applied to ψ $(\forall d \in |a|)$

By $(*)$, iff

$a \models \psi[s(y|d)(x|\bar{s}(t))] \quad (\forall d \in |a|)$

iff $a \models \psi[s(x|\bar{s}(t))(y|d)] \quad (\forall d \in |a|)$

iff $a \models \forall y\ \psi[s(x|\bar{s}(t))]$

iff $a \models \varphi[s(x|\bar{s}(t))]$

QED ~~Lemma~~ cases

QED Lemma

Now check:

if α a wff

t is subs. for x in α

$s : V \to |a|$

$a \models (\forall x\, \alpha \to \alpha^x_t)[s]$

enough to see:

if $a \models (\forall x\, \alpha)[s]$, $\quad a \models \alpha^x_t [s]$

Assume $\mathfrak{a} \models \forall x\, \alpha\, [s]$ (※)

To see: $\mathfrak{a} \models \alpha^x_t\, [s]$

By lemma, this holds iff

$\mathfrak{a} \models \alpha[s(x|\bar{s}(t))]$

But for any $d \in |\mathfrak{a}|$,

$\mathfrak{a} \models \alpha\, [s(x|d)]$ (by ※)

so in particular,

$\mathfrak{a} \models \alpha[s(x|\bar{s}(t))]$

———x———

Next go for other direction

$\Gamma \models \varphi \to \Gamma \vdash \varphi$ (*)

Special case of this

$\Gamma \models 0=1 \to \Gamma \vdash 0=1$

If $\Gamma \nvdash 0=1$ then $\Gamma \not\models 0=1$

($\Rightarrow \Gamma$ has a model!)

(*) is equivalent to:

"If Γ is consistent then Γ has a model"

R Solovay M125a Oct 17, 199 Friday

"0 = 1" shorthand for $\neg \forall x (x = x)$

Important fact $\vdash \neg 0 = 1$

Dfn Let Γ is a set of formulae

Then Γ is inconsistent if for some formula A,

$\Gamma \vdash A$ and $\Gamma \vdash \neg A$

Γ is consistent $\Leftrightarrow$ Γ is not inconsistent

Prop TFAE

(1) Γ is inconsistent

~~(2) Γ proves A for~~

(2) For every A (a wff in language of Γ),

$\Gamma \vdash A$

(3) $\Gamma \vdash 0 = 1$

Proof (1)$\to$(2)

Suppose $\Gamma \vdash A$, $\Gamma \vdash \neg A$

To see $\Gamma \vdash B$ (any wff B)

~~$\neg A/\ell$~~ $\neg A \to (A \to B)$ is a tautology

By rule T $\Gamma \vdash B$

(2)$\to$(3) Trivial

(3)$\to$(1) We know $\vdash \neg 0 = 1$

So if $\Gamma \vdash 0 = 1$,

Γ is inconsistent (Take "0 = 1" for A)

Gödel completeness Theorem Version 1:

If $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$

GCThm Ver 2

If Γ is consistent, then there is an L structure $\mathfrak{A}$ and an assignment of meaning to variables $s: V \to |\mathfrak{A}|$ such that $\mathfrak{A} \models \varphi[s]$, any $\varphi \in \Gamma$

at GCT1 $\Leftrightarrow$ GCT2 ; easy to show both ways

<u>Claim</u> GC2 implies GC1

So assume GC2

Will show GC1

Assume $\Gamma \nvdash \varphi$

We will show $\Gamma \not\models \varphi$

<u>Claim</u> $\Gamma; \neg\varphi$ is consistent

Proof: Suppose not. Go for contradiction

If $\Gamma; \neg\varphi$ is inconsistent

$\Gamma; \neg\varphi \vdash \varphi$

So $\Gamma \vdash \neg\varphi \to \varphi$

Note $(\neg\varphi \to \varphi) \to \varphi$ is a tautology

So $\Gamma \vdash \varphi$ /* contradiction */

By GC2, there is an L structure $\mathfrak{a}$,

and an $s: V \to |\mathfrak{a}|$

for every $\theta \in \Gamma; \neg\varphi$,

$\mathfrak{a} \models \theta[s]$

so in particular

(1) $\theta \in \Gamma$, $\mathfrak{a} \models \theta[s]$

(2) $\mathfrak{a} \models \neg\varphi[s]$

(3) $\mathfrak{a} \not\models \varphi[s]$

So (1), (3) show $\Gamma \not\models \varphi$

As was to be shown ■

Dfn Let Γ be a set of sentences in language L

Let $\mathfrak{a}$ an L structure. Then $\mathfrak{a}$ is a model of Γ

if each $\sigma \in \Gamma$ is true in $\mathfrak{a}$

Cor (to GC2) If Γ is a consistent set of sentences, Γ has a model

Dfn Let A be a set. Then A is countable if $A = \emptyset$ or there exists a map $F: \omega \to A$ $(\omega = \{0,1,2,\ldots\};$ which maps ω onto A

(equiv to say $\exists$ 1-1 into map from set to ω)

<u>Theorem</u> Let A be countable

And let $A^* = \{f \mid f$ is a finite sequence from $A\}$

$= \{f \mid f$ is a fn, $\mathrm{dom}(f) = \{0, 1, \dots, n-1\}$ $n \in \omega$,

range$(f) \subseteq A\}$

Then A^* is countable

<u>Proof</u> If $A = \emptyset$, $A^* = \{\emptyset\}$ A^* has 1 elt $\Rightarrow A^*$ is countable

If A is non empty, problem reduces to showing

ω^* is countable

I'll assign a number to each sequence in A^*

$\emptyset \longrightarrow 1$; empty sequence

$\langle 0 \rangle \to 2^{0+1} = 2$

$\langle 3 \rangle \longrightarrow 2^{3+1} = 16$

$\langle 1, 1 \rangle \to 2^{1+1} 3^{1+1} = 36$

In general the sequence of integers

$\langle a_1, \dots, a_{n-1} \rangle$ will be assigned the sequence

number

$p_0^{a_0+1} \, p_1^{a_1+1} \cdots p_{n-1}^{a_{n-1}}$; p_i is the i^{th} prime

; ie $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, ...

Basic Fact: This map $\omega^* \to \omega$ is 1-1

(so called fundamental thm arithmetic)

very important encoding

Now prove A^* is countable –

let $F : \omega \to A$ be onto

define $G : \omega \to A^*$ as follows

<u>Case</u> If n is a sequence number (say for $\langle a_0, \ldots, a_{r-1} \rangle$)

$$G(n) = \langle F(a_0), \ldots, F(a_{r-1}) \rangle$$

<u>Case 2</u> o.w. $G(n) = \emptyset$

$\therefore$ clear that G is onto

<u>Prop</u> If A is countable, $B \subseteq A$, then B is countable

<u>Pf</u> left to you $F : \omega \to A$ /* special case $B = \emptyset$ */

$G : \omega \to B$

<u>Prop</u> Let L be a first order language, $\bar{c}$ countable alphabet Σ.

Then $\{\varphi \mid \varphi \text{ is a wff from } L\}$ is countable

<u>Proof</u> should be clear (Σ countable $\to \Sigma^*$ countable $\to$ $\{wffs\}$ countable)

<u>Defn</u> L is <u>countable</u> iff the alphabet of L is countable

<u>High level description</u> (of proof of G(2))

(1) Going to enlarge language by adding countably many constants $c_0, c_1, \ldots$

(2) We will arrange that every member of ~~$\mathfrak{A}$~~ $|\mathfrak{a}|$ underlying set of model we construct is some $(c_i)^{\mathfrak{a}}$)

If $\mathfrak{a} \models \exists x\, \varphi(x)$, need $\varphi(c_j)$ true some j

we will arrange that for any formula of ~~our~~

L enlarged, $\varphi(x)$ have an axiom

$$(\exists x)\, \varphi(x) \to \varphi(c_j)$$

We will arrange that for any sentence of our language,

$$T^* \vdash \sigma \text{ or } T^* \vdash \neg \sigma$$

and still keep T^* consistent

Due 10/28/91

HW pp 139-140

Q 4ab, 5a, 6

You should review "effective enumerability" p 60-63

M125a lecture

Oct 23 1991
Wednesday

"Leaving the real world"

<u>Gödel completeness Theorem</u>

T set of ~~sentences~~ formulae

If T is consistent then there is a model $\mathfrak{a}$ and an $s: V \to |\mathfrak{a}|$

such that

$\forall \varphi \in T, \quad \mathfrak{a} \models \varphi[s]$

Proof goes in two phases:

<u>Defn</u> A theory is a pair $\langle L, Ax \rangle$:

(1) L is a first order language.

(2) Ax is a subset of wff of L.

<u>1st:</u> Enlarge T to a theory $T^* = \langle L_{T^*}, Ax_{T^*} \rangle$

so that

(1) T^* is consistent

(2) T^* has "enough constants"

ie if φ is a formula of L_T,
and x is a variable, then there is a
constant symbol c (depending on φ)
so that the following is an axiom of T^*

$$\neg \forall x \varphi \to \neg \varphi^x_c$$

(3) T^* is "complete" ie if σ is a wff of L_{T^*} then
either $\sigma \in Ax_{T^*}$ or $\neg\sigma \in Ax_{T^*}$

Phase 2 Build a model for T^*

ie we will build an L_{T^*} structure $\mathfrak{a}$,

and define $s: V \to |\mathfrak{a}|$ so that

$\mathfrak{a} \models \varphi[s]$ for all $\varphi \in Ax_{T^*}$

If $T^* \vdash t_1 = t_2$ will say $t_1 \sim t_2$

$s(v) = [v]$

Will define full structure $\mathfrak{a}$ as guided by T^*

We'll prove by induction

$T^* \vdash \varphi$ iff $\mathfrak{a} \models \varphi[s]$

End of outline

Start of phase 1

Let T_0 be our original consistent theory, T

Step 1 construct a new theory T_1 as follows.

$L_{T_1} = L_{T_0} \cup \{c_0, c_1, c_2, \ldots\}$

new constant symbols

~~Ax~~ $Ax_{T_1} = Ax_{T_0}$

/* Remark: we
* assume the
* language of T
* is countable
*/

Claim T_1 is consistent

Proof Suppose not. Let π be a proof of "$0=1$" in

Since Π is finite, can find an integer N so that if c_i apperas in Π, $i \leq N$

Again since Π is finite, we can find distinct variables $y_0, \dots, y_N$ not appearing in Π.

Let Π^* be obtained from Π by replacing c_i by y_i through out for $0 \leq i \leq N$.

Then Π^* is a proof of "$0=1$" in $T_\emptyset$

But this is absurd since $T_\emptyset$ is consistent

Upshot T_1 is consistent.

Step 2 (ensuring "enough constants")

Let $\langle\langle \varphi_i, x_i \rangle \mid i \in \omega\rangle$ be a listing of all pairs $\langle \varphi, x \rangle$ such that φ is a wff of L_{T_1} and x is a variable

(Some $\langle \varphi, x \rangle$ will appear several times in list; I do not care)

We will define a series of theories $T_{1,i}$ (for $i \in \omega$) such that:

1) $L_{T_{1,i}} = L_{T_1}$

2) $Ax_{T_{1,i}} = Ax_{T_1}$ plus finitely many new axioms

3) $Ax_{T_1} \subseteq Ax_{T_{1,1}} \subseteq Ax_{T_{1,2}} \subseteq \dots\dots$

At end, we'll define T_2 thus

$$L_{T_2} = L_{T_1}$$

$$Ax_{T_2} = \bigcup_{i \in \omega} Ax_{T_{1,i}}$$

Here we go:

$T_{1,0} = T_1$

Let define $T_{1,i+1}$

Pick a fresh constant c_j not appearing in $Ax_{T_{1,i}}$ or in φ_i

(No c_i's in Ax_{T_1} so at most finitely many $Ax_{T_{1,i}}$ and at most finitely many in φ_i.

So not all c_j's are stale.

Set $L_{T_{1,i+1}} = L_{T_{1,i}}$

$$Ax_{T_{1,i+1}} = Ax_{T_{1,i}} \cup \{ \neg \forall x_i \varphi_i \to \neg (\varphi_i)^{x_i}_{c_j} \}$$

That does construction of step 2.

Define T_2 as indicated above

Claim 1 T_2 has "enough constants"

Let φ be a wff of L_{T_2}

x a variable

so for some i, $\langle \varphi, x \rangle = \langle \varphi_i, x_i \rangle$

But then the axiom we need:

$\{\neg\forall x\,\varphi \to \varphi^x_c\}$ for some suitable c ~~were~~ was thrown in to T_2 at stage $i+1$.

M125a

25 Oct 91
Friday

Gödel Completeness Theorem

If T is a consistent theory, then there is an L_T structure $\mathfrak{A}$ and an $s: V \to |\mathfrak{A}|$:

$\mathfrak{A} \models \varphi[s]$ for every $\varphi \in Ax_T$

Phase 1

Build an auxiliary theory T^*:

1) $Ax_T \subseteq Ax_{T^*}$
2) T^* is consistent
3) T^* has enough constants
4) T^* is complete

Set $T_0 = T$

$T_1 = T_0 +$ add countably many constants $\langle c_i \mid i \in \omega \rangle$

Let $\langle \langle \varphi_i, x_i \rangle \mid i \in \omega \rangle$ be an enumeration of all pairs φ, x : φ is a wff of L_{T_1} and x a variable

Let $T_{1,i}$ be defined by induction on i

$L_{T_{1,i}} = L_{T_1}$

$Ax(T_{1,0}) = Ax(T_1)$

$Ax(T_{1,i+1}) = Ax(T_{1,i}) + \{ \neg \forall x_i \, \varphi_i \to \neg \varphi_i{}^{x_i}_{c_k} \}$?

where c_k is "fresh"

ie c_k does not appear in an axiom of $T_{1,i}$ or in φ_i

$L_{T_2} = L_{T_1}$

$Ax(T_2) = \bigcup_{i \in \omega} Ax(T_1, i)$

Clear: T_2 has "enuff constants"

<u>Remains to see</u> T_2 is consistent:

Proof that T_2 is consistent

A proof of "$0=1$" in T_2 will be a finite object

and can use only finitely many axioms of T_2

Hence for i large enough, the same goal shows

$T_{1,i}$ is inconsistent

So its enough to ~~show~~ see $T_{1,i}$ is consistent,

for all i.

Will prove by induction.

$T_{1,0}$ is just T_1 which we've seen is consistent

Now suppose $T_{1,k}$ is consistent

To see: $T_{1,k+1}$ is consistent

<u>Write</u>: x for x_k

φ for φ_k

c for fresh constant used in new axioms of T_{k+1}

$Ax(T_{k+1}) = Ax(T_k) \cup \{\neg \forall x \varphi \to \neg \varphi^x_c\}$

(R.A.A) p130

T_{k+1} yields a contradiction $(T_{k+1} \vdash 0=1)$

$: \neg(\neg \forall x\, \varphi \to \neg \varphi^x_c)$ — (a)

~~$T;\theta \vdash 0=1$~~

$T;\theta \vdash 0=1$

so $T;\theta \vdash \neg\theta$

$T \vdash \theta \to \neg\theta$

but $(\theta \to \neg\theta) \to \neg\theta$ is Tautology

so $T \vdash \neg\theta$ by rule T

$\neg(P \to Q) \equiv (P \wedge \neg Q)$

By (a), rule T

(b) $T_K \vdash \neg\forall x \varphi$

$T_K \vdash \neg\neg \varphi^x_c$

(c) $T_K \vdash \varphi^x_c$

But c does not appear in the axioms of T.

So by lemma on constants

(d) $T_K \vdash \forall x \varphi$

(Here T_K is $T_{i,K}$ throughout) (from (b) & (d))

So upshot: if T_{K+1} is inconsistent so is T_K.

I.e. if T_K is consistent so is T_{K+1}

Final Upshot: T_2 is consistent

We need the following lemma:

Lemma: Let T be a consistent theory. Let σ be a wff of L_T.

Then either $T+\sigma$ or $T+\neg\sigma$ is consistent

Here $L_{T+\sigma} = L_T$

$Ax(T+\sigma) = Ax(T) \cup \{\sigma\}$

Proof suppose not

$T+\sigma \vdash 0=1$

So $T \vdash \neg\sigma$ (*)

Similarly $T+\neg\sigma \vdash 0=1$

$T \vdash \neg\neg\sigma$ (**)

$\therefore$ T is inconsistent contradiction!

So $T+\sigma$ or $T+\neg\sigma$ must be consistent

QED

Now let $\langle \sigma_i \mid i \in \omega \rangle$ be an enumeration of all wffs of L_{T_2}

Define theories $T_{2,i}$ by induction on i

$T_{2,0} = T_2$

If $T_{2,i} + \sigma_i$ is consistent set $T_{2,i+1} = T_{2,i} + \sigma_i$

o.w set $T_{2,i+1} = T_{2,i} + \neg\sigma_i$

Now define T_3

$$L_{T_3} = L_{T_2}$$

$$Ax(T_3) = \bigcup_{i \in \omega} Ax(T_2, i)$$

Because all T_2, i's are consistent so is T_3 (?!)

It is clear by our construction that T_3 is compl

Let σ a ~~fo~~ wff of L_{T_3}

So $\sigma = \sigma_n$ for some n

By our construction, either

$\sigma_n \in Ax(T_2, n+1)$

or $\neg \sigma_n \in Ax(T_2, n+1)$

so therefore $\sigma_n \in Ax(T_3)$

or $\neg \sigma_n \in Ax(T_3)$

Set $T^* = T_3$

clear: (1) $L_{T^*} \supseteq L_T$

~~(2)~~ $Ax(T^*) \supseteq Ax(T)$

(2) T^* is consistent

(3) T^* has enough constants

(since $Ax(T_2) \subseteq Ax(T^*)$ (?)

$L_{T_2} \subseteq L_{T^*}$

and T_2 had enough constants

<u>Next Goal</u>: Construct a Structure $\mathfrak{A}$ which is a model of T

Let $\mathbb{X} = \{t \mid t \text{ is a term of } L_{T^*}\}$

Define a binary relation $\sim$ on $\mathbb{X}$ as follows -

$t_1 \sim t_2$ iff $T^* \vdash t_1 = t_2$

<u>Lemma</u> (1) For any $t \in \mathbb{X}$ $t \sim t$

(2) For any $t_1, t_2 \in \mathbb{X}$ $t_1 \sim t_2 \rightarrow t_2 \sim t_1$

(3) Let $t_1, t_2, t_3 \in \mathbb{X}$

If $t_1 \sim t_2$ and $t_2 \sim t_3$,

then $t_1 \sim t_3$

<u>Proof</u> (simple)

$T^* \vdash (\forall x)(x = x)$ (axiom)

$T^* \vdash t_1 = t_1$ (any T_1)

$T^* \vdash \forall x (x = x)$

$T^* \vdash \forall x (x = x) \to (x = x)^x_{t_1}$

$T^* \vdash \forall x (x = x) \to (t_1 = t_1)$

$T^* \vdash \forall x (x = x)$ (MP)

$\therefore T^* \vdash (t_1 = t_1)$

To be continued

$\vdash (t_1 = t_2) \to (t_2 = t_1)$

$\vdash x = y \to y = x$

$\vdash \forall x \forall y \, (x = y \to y = x)$

$\vdash \forall x \forall y \, (x = y \to y = x) \to \forall y \, (t_1 = y \to y = t_1)$

$\vdash \forall y \, (t_1 = y \to y = t_1)$

$\vdash \forall y \, (t_1 = y \to y = t_1) \to (t_1 = t_2 \to t_2 = t_1)$ (Subst carefully)

$\vdash t_1 = t_2 \to t_2 = t_1$

$\vdash x = y \to (x = x \to y = x)$

$\vdash [x = y \to (x = x \to y = x)] \to [x = x \to (x = y \to y = x)]$

$\vdash [x = x \to (x = y \to y = x)]$

$\vdash (x = y \to y = x)$

QED ($\vdash (x = y \to y = x)$

M125a - Lecture

1991 October 28
Monday

No HW this week

Quasi Homework: Think carefully through details I will handwave through in proof of Gödel completeness

T theory

Built an extension T^* of T w/ some nice extra properties

Next Goal: Build a model of T^*

$X = \{ t \mid t \text{ a term of } L_{T^*} \}$ — there are "scads" of these things

Defined a relation $\sim$ on X

$t_1 \sim t_2$ iff $T^* \vdash t_1 = t_2$

Lemma The following are true

(1) (Reflexivity) If $t \in X$, then $t \sim t$

(2) (Symmetry) If $s, t \in X$ and $s \sim t$ then $t \sim s$

(3) (Transitivity) If $s, t, u \in X$ and $s \sim t$, and $t \sim u$ then $s \sim u$

Proof (2) as an example (All proofs are similar)

let $t_1 \sim t_2$

By a HW exercise $\vdash \forall v_1 \forall v_2 (v_1 = v_2 \to v_2 = v_1)$

We can find other variables, x and y, which do not appear in t_1 or t_2

so $\vdash \forall x \forall y (x = y \to y = x)$ (2) Alphabetic variant of (1)

Remark If $\vdash \forall x \alpha$ then $\vdash \alpha^x_t$ (provided t subst for x)

since $\vdash \forall x \alpha \to \alpha^x_t$ (axiom and MP)

By 2 $\vdash (\forall y)(t_1 = y \to y = t_1)$ (3)

$\vdash t_1 = t_2 \to t_2 = t_1$

$T^* \vdash t_1 = t_2$

But $T^* \vdash t_1 = t_2 \to t_2 = t_1$

so $T^* \vdash t_2 = t_1$

so $t_2 \sim t_1$

Remark: A relation defined on X satisfying (1) – (3) is called an equivalence relation

Quotient set construction:

$\mathbb{Z}$ integers

$n \equiv m$ iff $(n-m)$ is even

$\mathbb{Z}$	
$\vdots$	
-2	-1
0	1
2	3
4	5
$\vdots$	$\vdots$

$\mathbb{Z}/\equiv$ consists of two elements: Even elts and odd elts

Defn: Let $t \in X$

Then $[t] = \{ s \in X \mid t \sim s \}$

$[t]$ is called the equivalence class of t

Prop: TFAE Let $t_1, t_2 \in X$

(1) $t_1 \sim t_2$

(2) $[t_1] = [t_2]$

Proof (2) $\to$ (1) $t_2 \sim t_2$ (reflexivity)

so $t_2 \in [t_2]$

by (2), $t_2 \in [t_1]$

Assume $t_1 \sim t_2$ To see $[t_1] = [t_2]$

enough to see $[t_1] \subseteq [t_2]$ and $[t_2] \subseteq [t_1]$

First we see $[t_1] \subseteq [t_2]$

Let $s \in [t_1]$

To see $s \in [t_2]$

$s \in [t_1] \to t_1 \sim s$

$\to s \sim t_1$

But $t_1 \sim t_2$

$\to s \sim t_2$

$\to t_2 \sim s$

so $s \in [t_2]$

On the other hand,

if $t_1 \sim t_2$,

$t_2 \sim t_1$

so (by the same argument)

$[t_2] \subseteq [t_1]$

$\therefore [t_1] = [t_2]$

So done ∎

Next Goal Define $L_T{*}$ structure $\mathfrak{A}$

$|\mathfrak{A}| = \{[t] : t \in X\}$

$\Gamma \vdash t_1 = t_2$

$\neg P \neg \quad \Gamma \vdash t_2 = t_1$

$\forall \vdash \quad \forall x_1 \forall x_2 \quad \forall y_1 \forall y_2 \alpha \ldots$

$\vdash \forall x_1 \forall x_2 \, \alpha \to \forall x_2 \forall x_1 \, \alpha$

$\vdash \forall x_1 \forall x_2 \, \alpha \to \forall x_2 \, \alpha$

$\vdash \forall x_2 \, \alpha \to \alpha$

$\vdash \forall x_1 \, \alpha \to \alpha$

$\forall x \forall y \, Pxy \vdash \forall y \forall x \, Pyx$

$\vdash \forall x \forall y \, Pxy \to \forall y \, Pxy$

$\vdash \forall x \forall y \, Pxy \to Pxy$

$\forall x \forall y \, Pxy \vdash Pxy$

$\vdash \forall x Pxy \to Pzy$

$\vdash \forall x \forall y Pxy \to \forall z \, Pzy$

$\vdash \forall x \forall y \, Pxy \to \forall z \forall w \, Pzw$

$\forall x \forall y \, Pxy \vdash \forall y \forall z \, Pyz$

Let c be a constant symbol of L_{T^*}

Defn $c_a = [c]$

Let P be an n-ary predicate of L_{T^*}

$P_a = \{ \langle [t_1], \ldots, [t_n] \rangle : T^* \vdash P t_1 \ldots t_n \}$

Let f be an n-ary function symbol of L_{T^*}

To define $f_a : |a|^n \to |a|$

Let $x_1, \ldots, x_n \in |a|$

pick $t_1, \ldots, t_n$ s.t. $x_i \in t_i$

Set $f_a(x_1, \ldots, x_n) = [f t_1 \ldots t_n]$

Major worry: Suppose $s_1, \ldots, s_n$ are other terms with $[s_i] = x_i$

will $[f t_1 \ldots t_n]$ be equal to $[f s_1 \ldots s_n]$

the goal: [illegible] equivalence [illegible] & [illegible] in L_{T^*}

$\frac{2}{3} + \frac{1}{2} = \frac{4}{6} + \frac{3}{6}$

$= \frac{4}{6} + \frac{5}{10}$

Lemma: Let $s_1 \sim t_1, \ldots, s_n \sim t_n$

Then $f s_1 \ldots s_n \sim f t_1 \ldots t_n$

Proof: the following is a theorem of formal logic.

$\forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n [(x_1 = y_1 \wedge x_2 = y_2 \wedge \ldots x_n = y_n) \to (f x_1 \ldots x_n = f y_1 \ldots y_n)]$

Rest of proof: use variant. Rest of proof is like our proof that $\sim$ is symmetric

Lemma Let $s_1 \sim t_1, \ldots, s_n \sim t_n$

Then $T^* \vdash Ps_1 \ldots s_n \Leftrightarrow Pt_1 \ldots t_n$

Proof (amounts to)

$$T^* \vdash (s_1 = t_1 \wedge \ldots s_n = t_n) \rightarrow (Ps_1 \ldots s_n \Leftrightarrow Pt_1 \ldots t_n)$$

Use equality Axioms

Prop Let $t_1, \ldots, t_n$ terms

Then $\langle [t_1], \ldots, [t_n] \rangle \in P_\mathfrak{a}$ iff

$$T^* \vdash Pt_1 \ldots t_n$$

$\Leftarrow$ · clear from defn of $P_\mathfrak{a}$

If $\langle [t_1], \ldots, [t_n] \rangle \in P_\mathfrak{a}$

then for some $s_1, \ldots, s_n$

$\langle [t_1], \ldots, [t_n] \rangle = \langle [s_1], \ldots, [s_n] \rangle$ and $T^* \vdash Ps_1 \ldots s_n$

~~But by the lemma~~

So $[s_1] = [t_1], \ldots, [s_n] = [t_n]$,

so $s_1 \sim t_1, \ldots, s_n \sim t_n$

so by lemma

$$T^* \vdash Pt_1 \ldots t_n$$

—— × ——

M125a

Oct 30'91
Wednesday

Theory T, T consistent

Goal: to construct an L_T structure $\mathfrak{a}$,

$s : V \to |\mathfrak{a}|$

so that for all $\varphi \in Ax_T$, $\mathfrak{a} \models \varphi[s]$

We construct an auxiliary theory T^*

Last time built $\mathfrak{a}$

$|\mathfrak{a}| = \{ [t] : t \text{ is a term of } L_{T^*} \}$

$c_{\mathfrak{a}} = [c]$

$f_{\mathfrak{a}}([t_1], \ldots, [t_n]) = [ft_1 \ldots t_n]$ /* checked to ensure
* well defined
*/

$\langle [t_1], \ldots, [t_n] \rangle \in P_{\mathfrak{a}} \Leftrightarrow T^* \vdash Pt_1 \ldots t_n$

We now define

$s : V \to |\mathfrak{a}|$

$s(v_i) = [v]$

Proposition Let t be a term of L_{T^*}

Then $\bar{s}(t) = [t]$

Proof Easy induction (left to you)

<u>Claim</u> Let σ be a wff of L_{T^*}

Then T^* proves σ iff $\mathfrak{a} \models \sigma[s]$

$(T^* \vdash \sigma)$ iff $\mathfrak{a} \models \sigma[s]$

<u>Proof</u>: Define rank(σ) = # of "$\neg$"s in σ + 2 · # of "$\to$"s in σ ~~...~~ + # of "$\forall$s" in σ

We'll prove by induction on rank(σ) that lemma is true for σ.

So in proving Lemma at σ, we can assume it known for all wffs of ~~...~~ rank ~~...~~ less than σ

Case 1> σ has form $t_1 = t_2$

$$\mathfrak{a} \models (t_1 = t_2)[s] \text{ iff } \bar{s}(t_1) = \bar{s}(t_2) \text{ iff}$$

$$[t_1] = [t_2] \text{ iff}$$

$$t_1 \sim t_2 \text{ iff}$$

$$T^* \vdash t_1 = t_2$$

$\blacksquare$

Case 2> σ is $P t_1 \ldots t_n$

$$\mathfrak{a} \models P t_1 \ldots t_n [s]$$

iff $\langle \bar{s}[t_1], \ldots, \bar{s}[t_n] \rangle \in P_{\mathfrak{a}}$

iff $\langle [t_1], \ldots, [t_n] \rangle \in P_{\mathfrak{a}}$

iff $T^* \vdash P t_1 \ldots t_n$

$\blacksquare$

Case 3> σ is $\neg\psi$

Notice our IH applies to ψ

Assume

$T^* \vdash \sigma$

To see $\mathfrak{A} \models \sigma[s]$

$T^* \vdash \sigma$

ie $T^* \vdash \neg\psi$

since T^* is consistent,

$T^* \nvdash \psi$

So by IH $\mathfrak{A} \not\models \psi[s]$

So $\mathfrak{A} \models \neg\psi[s]$

ie $\mathfrak{A} \models \sigma[s]$

Now assume $\mathfrak{A} \models \sigma[s]$

To see $T^* \vdash \sigma$

ie $\mathfrak{A} \models \neg\psi[s]$

so $\mathfrak{A} \not\models \psi[s]$

By IH $T^* \nvdash \psi$

But T^* is complete

$\therefore T^* \vdash \psi$ or $T^* \vdash \neg\psi$

since $T^* \nvdash \psi$, $T^* \vdash \neg\psi$

ie $T^* \vdash \sigma$

$\square$

<u>Case 4</u> $\quad \sigma$ is $\psi \to \chi$

notice $\text{rank}(\neg\psi) < \text{rank}(\sigma)$

(because of how we defined rank)

$$\begin{aligned} \text{rank}(\neg\psi) &= \text{rank}(\psi) + 1 \\ &\leq \text{rank}(\psi) + 2 \\ &\leq \text{rank}(\sigma) \end{aligned}$$

also $\text{rank}(\chi) < \text{rank}(\sigma)$

First assume

$$\mathfrak{a} \vdash \sigma[s]$$

To see $\quad T^* \vdash \sigma$

$\mathfrak{a} \models \sigma[s]$

$\Rightarrow \mathfrak{a} \models \neg\psi[s]$ or $\mathfrak{a} \models \chi[s]$

$\Rightarrow$ (IH) $\quad T^* \vdash \neg\psi$ or $T^* \vdash \chi$

$\Rightarrow$ (Taut implication) $\quad T^* \vdash \psi \to \chi$

$\Rightarrow \quad T^* \vdash \sigma$

$\blacksquare\ (\Rightarrow)$

Now assume

$\mathfrak{a} \nvDash \sigma[s]$

To see $\quad T^* \nvdash \sigma$

$\mathfrak{a} \nvDash \sigma[s]$

$\Rightarrow \mathfrak{a} \models \psi[s]$ and $\mathfrak{a} \models \neg\chi[s]$

$\Rightarrow T^* \vdash \psi$ and $T^* \vdash \neg\chi$ $\quad$ ↖ IH can apply to $\neg\chi$ too ($\because$ nots are cheaper!!)

But $\psi \to (\neg\chi \to \neg(\psi \to \chi))$ is Taut

By rule T, $\quad T^* \vdash \neg(\psi \to \chi)$; T^* consistent so

$T^* \nvdash (\neg\psi \to \chi)$ i.e. $T^* \nvdash \sigma$ ..

case 5 σ is $\forall x\, \psi$ "I cant find any room appropriate for public lectures"

Since T^* has "enough constants" for some constant symbol c,

$\neg \forall x\, \psi \to \neg \psi^x_c$ is an axiom of T^*

Remark For any alphabetic variant, ψ_1 of ψ,

rank (ψ_1) = rank (ψ)

Also for any term t,

rank (ψ^x_t) < rank $(\forall x\, \psi)$ /* If IH on length, has a new (explosion) */

So if ψ_1 is an alphabetic variant of ψ, induction hypothesis applies to $(\psi_1)^x_t$

first direction:

Suppose $\mathfrak{a} \models \sigma[s]$

To see $T^* \vdash \sigma$

$\mathfrak{a} \models \sigma[s]$

$\mathfrak{a} \models \forall x \psi[s]$

So $\mathfrak{a} \models \psi[s(x\,|\,[c])]$; $\bar{s}[c] = [c]$

So $\mathfrak{a} \models \psi^x_c\,[s]$ /* compare Evaluation, Substitution

Now by IH, $T^* \vdash \psi^x_c$

Also $T^* \vdash \neg \forall x \psi \to \neg \psi^x_c$ (Axiom)

$\therefore$ $T^* \vdash \psi^x_c \to \forall x\, \psi$

$\therefore$ by MP $T^* \vdash \forall x\, \psi$

M125a

1 Nov '91
Friday

MT 2 : Nov 18 (Mon)

on material through Nov 11
open book / open notes

Consistent theory T

Goal is : build a model $\mathfrak{a}$ and an $s : V \to |\mathfrak{a}|$

$\mathfrak{a} \models \theta[s]$ for all $\theta \in Ax_T$

what's been done:

Enlarged T to T^*

We are proving

Defined $\mathfrak{a}$, s (Equivclasses, obvious extension)
$\quad \hookrightarrow s : V \to |\mathfrak{a}|$

Goal now is to prove by induction on rank(σ) that

$T^* \vdash$ iff $\mathfrak{a} \models \sigma[s]$

If we can do this, will be done. Namely if $\theta \in Ax_T$

then $\theta \in Ax_{T^*} \to T^* \vdash \theta \to \mathfrak{a} \models \theta[s]$

Only have to finish case (v)

$\to$ σ is of form $\forall x \varphi$

so far showed: $\mathfrak{a} \models \forall x \varphi[s]$ then $T^* \vdash \forall x \varphi$

(used enuff constants!)

Remains to show: $\mathfrak{a} \not\models \forall x \varphi[s]$ then $T^* \not\vdash \forall x \varphi$

Lemma Let $d \in \mathfrak{a}$ Then $d = [c_j]$ for some $j \in \omega$

Proof: we know $d = [t]$ for some term t of L_{T^*}

Since T^* has "enough" constants for some c_j, an
some x not appearing in t

$\neg \forall x \; \neg (x = t) \rightarrow \neg\neg \; (c_j = t)$

is an axiom of T^*

Clearly, $T^* \vdash \neg \forall x \neg (x = t)$

But $T^* + (*)$

$T^* \vdash \neg\neg (c_j = t)$

$T^* \vdash c_j = t$

So $[c_j] = [t] = d$

■ QED Lemma

Result: Dont have to worry about substitutability since c_j is a constant $\Rightarrow$ no free variables

Now suppose

$\mathfrak{a} \not\models \forall x \; \varphi[s]$

To see $T^* \nvdash \forall x \varphi$

—— x ——

If $\mathfrak{a} \not\models \forall x \; \varphi[s]$

then for some $d \in |\mathfrak{a}|$

$\mathfrak{a} \not\models \varphi[s(x|d)]$; but $d = [c_j]$ by lemma (*)

By Enderton p127 top of page (*) implies

$\mathfrak{A} \nvDash \varphi^x_{c_j}[s]$ ~~#~~ The text goes crazy $\because t$ $\overset{\to \text{subs?}}{\text{in place of } c_j}$

Notice that $\text{rank}(\varphi^x_{c_j}) < \text{rank}(\forall x\, \varphi)$

Applying IH,

$$T^* \nvdash \varphi^x_{c_j}$$

But if $T^* \vdash \varphi^x_{c_j}$ then clearly we would have

$$T \vdash \varphi^x_{c_j}$$

Upshot $T^* \nvdash \forall x\, \varphi$

QED case 5

QED Gödel Completeness Theorem

Consequences of the Gödel Completeness Theorem

- If φ is logically valid then φ has a proof (using only MP) and ~~[illegible]~~ our six types of axioms (We know all laws of logic)

Suppose L has only finitely many non logical symbols:

Its believable (and we will prove)

" $\exists$ a decision procedure to check if a sequence of wffs is a proof "

<u>Claim</u> There is an algorithm, which given $i \in \omega$ generates v_i a wff of L such that

(1) Every v_i is valid

(2) Every valid wff appears in list

Let w_i be a effective listing of all finite sequences of wffs

Define v_i as follows:

If w_i is a proof set v_i equal to its last line

If w_i is not a proof, set $v_i = \exists x (x = x)$

We will show that in general there is no algorithm that given a wff will tell if its valid or not.

Nov 4 1991
Monday

Consequence of Completeness Theorem

Compactness Theorem: (Related to topology)
Let Σ be a set of sentences
Suppose every finite subset of Σ has a model
Then Σ has a model

Proof: ETS if Σ has no model, some finite subset of Σ has no model

But if Σ has no model,
$\Sigma \vdash$ "$0=1$" (G.C. Thm)

Proof is finite $\Rightarrow$ uses only finitely many sentences of Σ say those in Σ_0

so $\Sigma_0 \vdash$ "$0=1$"

so Σ_0 has no model

Application: A model of True Arithmetic (TA) not isomorphic to the standard model

$L = \{ \ldots, 0, 1, S, +, \cdot \}$ f.o.l.

Standard model $\mathfrak{N}$ has $|\mathfrak{N}| = \omega = \{0, 1, 2, \ldots\}$ non neg integers

$\mathfrak{N}$ is an L structure
(interpret $+, \cdot, S$ in usual way)

$TA = \{\sigma \mid \sigma$ is a sentence of L and σ true in $\mathfrak{N}\}$

Goal to produce a model $\mathfrak{M}$ of TA not isomorphic to $\mathfrak{N}$. Way we will make it not isomorphic is to have an elt c_M with

$c_M \neq 0_M, \; c_M \neq S^n 0_M, \ldots$

$L^+ = L \cup \{c\}$
$\uparrow$ new constant

$T^+ = \cancel{A X \neq}\ TA \cup \{c \neq 0, c \neq 1, c \neq 2, \ldots c \neq S^n 0\}$

Clear if M^+ is a model of T^+
$M^+ | L$ is a model of TA not isomorphic to $\mathfrak{N}$
since if $f : |\mathfrak{N}| \to |M|$ is an iso, nothing can map onto c_{M^+}.

Remains to show T^+ has a model
By compactness, it is enough to see every finite subset, S, of T^+ has a model
Since S is finite,
"$c \neq S^n 0$" doesn't appear in S

There is a model M_S of S
$|M_S| = |\mathfrak{N}| \qquad 1_{M_S} = 1_{\mathfrak{N}} \qquad \cdot_{M_S} = \cdot_{\mathfrak{N}}$
$0_{M_S} = 0_{\mathfrak{N}} \qquad + \ldots$

Technical Goal toward Gödel Completeness theorem:

Want to give a precise mathematical defn of "effectively comp"

Will describe an ideal class of digital computers called register machines.

Every register m/c has a program, which is a finite sequence of instructions.

$\underline{P} = \langle P_0, \ldots, P_n \rangle$ P_i is some instruction.

Each machine has an infinite number of registers $R_0, R_1, R_2, \ldots$ capable of holding an arbitrarily large non negative integer

Also an instruction register I.

State of m/c at an instant is described completely by knowing ① program ② contents of I (c(I)) ③ contents of all R_j's

R. Solovay M125a-Lecture 6 NOV '91
Wednesday

Register Machines:

- Program
- Infinite set of registers $R_0, R_1, R_2, \ldots$ which contain non negative integers
- Instruction register I tells which instruction to do next

1) ADD 1 i:

This has following effects $c(R_i) := c(R_i) + 1$
$c(I) := c(I) + 1$

So if for example,

$c(I) = 12$

$c(R_5) = 7$

and instruction # 12 of our program is ADD 1 5,

then after I_{12} is performed

$c(I) = 13$

$c(R5) = 8$

$c(R_i)$ is unchanged $i \neq 5$

2) SUB 1 i j: (subtract one; conditional transfer on zero)

If $c(R_i) > 0$, then set $c(R_i) := c(R_i) - 1$,

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

If $c(R_i) = 0$ then set $c(I) = j$

3) TRA j (Unconditional transfer)

4) HALT (The machine halts)

5) COPY i j

Set $c(R_j) := c(R_i)$

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

$c(R_i) = c(R_i)$

6) STZ i : (store zero)

$c(R_i) := 0$

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

Illegal instruction : HALTs

State of a register m/c consists of

(1) Program (finite sequence of instructions $\langle I_0, \ldots, I_n \rangle$)

(2) $c(I)$ (current instruction)

(3) $c(R_i)$ for all i

If we have a state S, define <u>next</u> state S^* as follows:

If $c(I)$ ~~to~~ in state S is $> m$ ($m = \#$ instr)

S^* is undefined, and m/c halts

If $c(I) = j$ with $j \leq m$ and I_j = "HALT",

m/c halts S^* undefined

o.w. determine the contents of R_i's for S^* and I

for S^* as just described

keep same program

Suppose s_0 is a machine state

either we can define s_n for all n by

$s_{n+1} = (s_n)^*$

(m/c never halts)

Otherwise possibility is s_i is defined for $i \le k$ via

$s_{i+1} = (s_i)^*$, and in state s_k, machine halts

[$(s_k)^*$ is undefined]

<u>Dfn</u> Let P be a program
(finite sequence of instructions) and $n \geq 1$

we will define $f_{P,n}$ a function w/ $\mathrm{Dom}(f_{P,n}) \subseteq \omega^n$

$\mathrm{Range}(f_{P,n}) \subseteq \omega$

as follows

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

Let $s_0(x_1, \ldots, x_n)$ be the following m/c state

program is P

$c(I) = 0$

$c(R_0) = 0$

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

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

start machine in this state

Case 1 M/c runs forever

Then $f_{P,n}(x_1, \ldots, x_n)$ is undefined

case 2 Machine eventually halts in some state

$S'(x_1, \ldots, x_n)$

Then $f_{P,n}(x_1, \ldots, x_n)$ is contents of R_0 in state

$S'(x_1, \ldots, x_n)$

Dfn f is partial recursive if $f = f_{P,n}$ for some P, n

Dfn f is recursive if its partial recursive and $Dom(f) = \emptyset$ [sic]

Program to compute $x+y$ i computes f where

$f(x,y) = x+y$

0. CPY 1 Ø
1. SUB 1 2 4
2. ADD 1 Ø
3. TRA 1
4. HALT

Last Time -

Defined "register machine"

If P is a program for a register m/c and $n \geq 1$

defined $f_{P,n} : A \to \omega$

$A \subseteq \omega^n$

f is partial recursive if $f = f_{P,n}$ some $n \geq 1$ and program P.

f is ~~total is~~ total recursive if f is partial recursive and everywhere defined.

Next set of goals:

1. Prove general theorems about which functions are recursive
2. Give an alternate characterization of recursive functions (μ recursive)

Develop a list of simple recursive functions "so that can emerge from the ooze"

(1) $Z(x) = 0$ for all x

Z is recursive: Program STZ 0
HALT

(2) $\pi_i^n(\langle x_1, \ldots, x_n\rangle) = x_i$

This is recursive: Program COPY i 0
HALT

(3) $S(x) = x + 1$

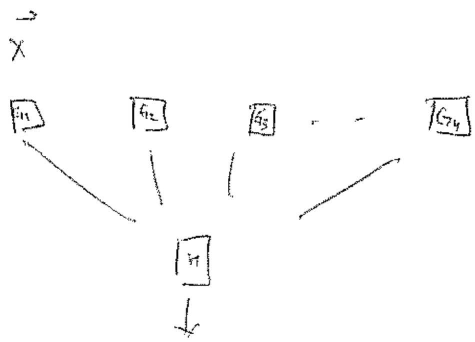
This is recursive: Program COPY 1 0
ADD 1 0
HALT

Dfn Let $m, n \geq 1$

Let $H : \omega^m \to \omega$

and for $1 \leq i \leq m$

$G_i : \omega^n \to \omega$



Then we can define $F : \omega^n \to \omega$ (the composition of H and the $\vec{G}$'s)

i.e. if $z_i = G_i(x_1, \ldots, x_n)$ for $1 \leq i \leq m$

$F(\vec{x}) = H(z_1, \ldots, z_n)$

Thm If $G_1, \ldots, G_m$ are recursive and H is recursive, then F (the composition $H \circ \vec{G}$) is recursive

<u>Lemma</u> Let $f : \omega^n \to \omega$ be recursive. Then there is a program $\underline{P}$ that computes f such that:

1. Last line of $\underline{P}$ is a HALT instruction
2. All transfer and conditional transfer addresses are to lines in program
3. No other HALT instructions in program
4. Program may use some scratch registers $R_{n+1}, \ldots, R_{n+l}$

After halt of program

$c(R_1), \ldots, c(R_n)$ will be unchanged

$c(R_{n+1}), \ldots, c(R_{n+k})$ will be zero

Say P_0 uses registers $R_0, \ldots, R_{n+k}$

Proof Say P_0 is some program that computes f

Add preamble to P_0 that contains the following

1) Add a preamble:

COPY 1 $n+k+1$
COPY 2 $n+k+2$
$\vdots$
COPY n $n+k+n$

This squirrels away input variables where they won't get hurt

2) Next follow by a translation of P_0
replace each transfer or conditional transfer l by $n+l$)

3) Add a postamble

Instruction Z: COPY $n+k+1$, 1 STZ $n+1$
COPY $n+k+2$, 2 STZ $n+2$
$\vdots$ $\vdots$
COPY $n+k+n$, n STZ $n+k+n$
HALT

4) Replace "illegal" transfer or conditional transfer out of copy of P_0, replace address by Z

5) Replace HALTs in copy of P_0 by a TRA Z

This works

Old Program

Proof converts this to

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

0	CPY 1, 3	7	CPY 3, 1
1	CPY 2, 4	8	CPY 4, 2
2	CPY 1, 0	9	STZ 3
3	SUB1 2, 6	10	STZ 9
4	ADD1 $ 0	11	HALT
5	TRA 3		
6	TRA 7		

Let $F(\vec{x}) = H(\vec{G}(\vec{x}))$

Following program for F works

$\boxed{G_1}$ computes $G_1(x_1, \ldots, x_n)$ & stores it in R_{n+1}

$\boxed{G_2}$ computes $G_2(x_1, \ldots, x_n)$ & stores it in R_{n+2}

. . . .

$\boxed{G_m}$ computes $G_m(x_1, \ldots, x_n)$ & stores it in R_{n+m}

$\boxed{H}$ computes $H(c(R_{n+1}), \ldots, c(R_{n+m}))$ and stores it in R_0

then HALT

$\boxed{G_i}$ is a modification of a nice program for computing G_i

Will have different scratch registers for all subprograms & all these different from

$F : \omega^n \to \omega$ recursive
and $H(\vec{x}) = F(G_1(\vec{x}), \dots, G_m(\vec{x}))$
then H is recursive

Notation: $F : A \to B$
means F is a function $\mathrm{Dom}(F) = A$ $\mathrm{Ran}(F) \subseteq B$

Notation: $F \vdots A \to B$
means F is a function $\mathrm{Dom}(F) \subseteq A$ $\mathrm{Ran}(F) \subseteq B$

Now suppose $F \vdots \omega^m \to \omega$
and $G_1, \dots, G_m \vdots \omega^n \to \omega$ are partial recursive

[D]efine $H \vdots \omega^n \to \omega$

$H(x_1, \dots, x_n)$ is defined $\Leftrightarrow$
1) If $1 \le i \le m$,
$G_i(x_1, \dots, x_n)$ defined
(and $= z_i$, say)
and $F(z_1, \dots, z_m)$ is defined
In this case $H(\vec{x}) = F(z_1, \dots, z_m)$

Same proof as for total functions shows $F, G_1, \dots, G_m$
are partial recursive, so is H

G_1
$\vdots$
G_m

Primitive recursion

Lemma ~~variation 1~~: Given $c \in \omega$ and $F : \omega^2 \to \omega$
then there is a unique function H
such that

① $H(0) = c$

② $H(Sn) = F(n, H(n))$

H is said to be obtained from c, F by primitive recursion.

Example: $n!$ $\quad 0! = 1$

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

So $n!$ is gotten by primitive recursion from

$c = 1$

$F(x, y) = (x+1) \cdot y$

Showing uniqueness is trivial (induction on n)

Lemma Let $n \geq 2$

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

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

"Your question sounds like hopeless mush"

Then there is a ~~variable~~ unique function

$F : \omega^n \to \omega$ such that

$F(x_1, \ldots, x_{n-1}, 0) = G(x_1, \ldots, x_{n-1})$

$F(x_1, \ldots, x_{n-1}, Sy) = H(x_1, \ldots, x_{n-1}, y, F(\vec{x}, y))$

F is said to be obtained from G, H by primitive recursion

Example 2
$$+(x,0)=x$$
$$+(x,Sy)=S(+(x,y))$$
$$x+0=x$$
$$x+Sy=S(x+y)$$

Example 3
$$\cdot(x,0)=0$$
$$\cdot(x,Sy)=x+\cdot(x,y)$$

Example 4 ~~$\cdot(x,y)$ is $x\cdot y$~~

$$Exp(x,0)=1$$
$$Exp(x,Sy)=Exp(x,y)\cdot x$$

Lemma If $G:\omega\to\omega$ and $H:\omega^3\to\omega$ are recursive, and
$$F(x,0)=G(x)$$
$$F(x,Sy)=H(x,y,F(x,y))$$
then F is recursive

Outline of Program

R_0 : Answer

$_1$: X

R_2 : Y

R_3 : Z (Z will go from 0 to y)

R_4 : F(X, Z)

R_5 : Scratch to store F(X, Z) while computing F(X, Z+1)

Initialization

```
        STZ  3
        [ c(R4) := G(c(R1)) ]

LOOP    SUB_1  2  END
        CPY  4  5
        [c(R4) := H(c(R1), c(R3), c(R5))]
        ADD_1  3
        TRA  LOOP

END     COPY  4  0
        HALT
```

This program clearly works (?!)

Variation 1 Let $n \geq 2$

If $G: \omega^{n-1} \to \omega$

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

are recursive and $F: \omega^n \to \omega$

is defined by primitive recursion,

F is recursive

<u>Same Proof</u>

Variation 2 If $c \in \omega$ and $H: \omega^2 \to \omega$ are recursive and F is defined from c, H by primitive recursion, F is recursive

Proof is essentially the same

Defn $F: \omega^{n+1} \to \omega$

is recursive

Define a partial function

$G: \omega^n \to \omega$ as follows

$G(x_1, \dots, x_n)$ is the least y: $F(x_1, \dots, x_n, y) = 0$

$G(\vec{x})$ is defined iff there is a z such that

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

If so, its least such z

$G(\vec{x}) = \mu z [F(\vec{x}, z) = 0]$

μz is read "least such z"

G is said to be obtained from F by μ recursion

lect#31
13-nov-91

prog: If $F: \omega^{n+1} \longrightarrow \omega$
and I define $g: \omega^n \longrightarrow \omega$ by
$g(x_1, \dots, x_n) = (\mu z)[f(x_1, \dots, x_n, z) = 0]$.
Then g is partial recursive.

R_0 : Answer
R_1 : x_1
⋮
$R_n = x_n$
$R_{n+1} = z$
$R_{n+2} = f(\vec{x}, z)$.

Following prog. works:

```
Init  STZ  n+1
Loop  [c(R_{n+2}) :=
          f(c(R_1), ..., c(R_{n+1}))]
      SUB1  n+2   End
      ADD1  n+1
      TRA   Loop
End   copy  n+1  0
      HALT.
```

Def: The class of primitive recursive functions is the smallest collection $\mathcal{C}$ of functions :

1) Z, S and the projection functions π_i^n are in $\mathcal{C}$.
2) $\mathcal{C}$ is closed under composition. I.E. if $f: \omega^m \to \omega$ is in $\mathcal{C}$ and $g_1, \dots, g_m : \omega^n \to \omega$ are all in $\mathcal{C}$. and $h: \omega^n \to \omega$ is defined by $h(\vec{x}) = f(g_1(\vec{x}), \dots, g_m(\vec{x}))$ then $h \in \mathcal{C}$.
3) $\mathcal{C}$ is closed under primitive recursion.

ex: on ω $J(n) = 2^{2^{\cdot^{\cdot^{2}}}}$ n 2's.

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

$J(4) = 2^{16}$
$J(5) = 2^{60,000}$?

$J(0) = 1$ $J(1) = 2^1$ $J(2) = 2^2$ $J(3) = 2^{(2^2)}$ $J(4) = 2^{(2^{2^2})}$

Defn: μ-recursive functi's are the smallest class $\mathcal{C}$: Clauses 1) – 3) as before apply, and

4) If $g: \omega^{n+1} \to \omega$ is in $\mathcal{C}$ and $f(\vec{x}) = \mu z\,[g(\vec{x}, z) = 0]$ is total, Then $f \in \mathcal{C}$.

$\mathcal{C}$ is closed under μ-recursion.

Clear: Every primitive recursive fn is μ-recursive.
Every μ-recursive fn is recursive.
(By induction on "proof" something is μ-recursive.).

Gossip: 1) Not every recursive fn is primitive recursive.
2) It is true that every recursive funct is μ-recursive.
3) Prob. 3 shows not all recursive functions are "potentially computable".

Careful proof that: $h(x, y) = x + y$
$h: \omega^2 \to \omega$ is a primitive recursive fn.

pf: The following fns are prim. recursive.

① $f_1(x, y, z) = z$ is ——— (π^3_3)

② $f_2(x) = x + 1$ ——— (S)

③ $f_3(x, y, z) = Sz$ composition of f_2 & f_1

④ $f_4(x) = x$ (π^1_1)

If $f_5(x, 0) = f_4(x) = x$
$f_5(x, Sy) = f_3(x, y, f_5(x, y))$

Then $f(x, y) = x + y$.

proofs require projection & composition functions.

$(\exists x, y, z, n)\ \underbrace{(x^n + y^n = z^n \ \& \ n \geq 3)}_{\text{computable predicate}}$.

Defn: A relation $R \subseteq \omega^n$ is (prim.) recurs. if the function $\chi_R : \omega^n \to \omega$ is ~~[illegible]~~.

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

$\chi_R(\vec{x}) = 0$ otherwise

prop.: If $R \& S \subseteq \omega^n$ are prim. recur.

so is $\neg R$ $\quad (\omega^n - R)$

$R \vee S$ $\quad (R \cup S)$

$R \wedge S$.

Defn: $sg : \omega \to \omega$ as follows:

$sg(0) = 0$

$sg(St) = 1$

$\left(\text{so } sg(x) \in \{0, 1\} \atop = 1 \text{ iff } x > 0 \right)$

$\overline{sg}(0) = 1$

$\overline{sg}(St) = 0.$

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

~~Pf of prop.~~:

$\chi_{\neg R} = \overline{sg}(\chi_R)$

so R prim. rec. $\to \neg R$ is prim rec.

$\chi_{R \vee S}(\vec{x}) = sg[\chi_R(\vec{x}) + \chi_S(\vec{x})]$.

$R \wedge S = \neg(\neg R \cup \neg S)$

so R, S prim. rec. $\longrightarrow R \wedge S$ is prim. recur.

There are examples where $R(x, \vec{y})$ is prim. rec. and $(\exists x) R(x, \vec{y})$ is not.

Introduce following abbreviation.

$(\exists x < y) R(x, y_1, \ldots, y_n)$

means: $(\exists x)(x < y \ \& \ R(x, y_1, \ldots, y_n))$.

Variation:

$(\exists x \leq y) R(x, y_1, \ldots, y_n)$. means

$(\exists x < Sy) R(x, y_1, \ldots, y_n)$.

$(\forall x < y) R(x, y_1, \ldots, y_n)$ means $(\forall x)(x < y \rightarrow R(x, y_1, \ldots, y_n))$

Easy fact:

$(\forall x < y) R(x, y_1, \ldots, y_n)$ iff $\neg(\exists x < y) \neg R(x, y_1, \ldots, y_n)$

r.e. closed under bdd quantification

prop: If $R(x, y_1, \ldots, y_n)$ is prim. rec. and $S(z, y_1, \ldots, y_n)$ is $(\exists x < z) R(x, y_1, \ldots, y_n)$ Then S prim. rec.

Notice $S(0, y_1, \ldots, y_n) = F$

$S(t+1, y_1, \ldots, y_n) \Leftrightarrow S(t, y_1, \ldots, y_n) \vee R(t, y_1, \ldots, y_n)$

M Solovay

M125a

Nov 15 1991
Friday

$R(x_1, \ldots, x_n, y)$ is primitive recursive
so is $(\exists y < z)\, R(x_1, \ldots, x_n, y)$

or if $R(\vec{x}, y)$ is primitive recursive,
so is $(\forall y < z)\, R(\vec{x}, y)$

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

rop The predicate $x = y$ is primitive recursive

roof consider the following primitive recursive function

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

$x \dot{-} 0 = x$
~~Sx/9tt~~ $x \dot{-} Sy = pd(x \dot{-} y)$

(so $x \dot{-} y = x - y$ if $x \geq y$
$= 0$ if $x < y$)

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

$x \geq y$ iff $Sx > y$
$x = y$ iff $x \geq y$ and $y \geq x$

We've already remarked

$$\cdot(x,y) = x\cdot y$$

$$\exp(x,y) = x^y \quad \text{are prim recursive}$$

$$x\cdot 0 = 0$$

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

$$x^0 = 1$$

$$x^{Sy} = x^y \cdot x$$

can now prove scads of things -

x divides y is prim recursive

x divides y iff

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

Prob 2A "x is prime" is primitive recursive

<u>Next goal</u> function: $i \to p_i$ is primitive recursion

$p_0 = 2,\ p_1 = 3, \ldots$

<u>Next</u> Defn by cases

suppose $R_1, \ldots, R_n$ relations $\subseteq \omega^n$

Let $f_1, \ldots, f_n : \omega^m \to \omega$

Key assumption: For any $\vec{x} \in \omega^m$,

exactly one of

$\vec{x} \in R_1, \ldots, \vec{x} \in R_n$ is true

Define a function

$f : \omega^m \to \omega$,

by $f(\vec{x}) = f_i(\vec{x})$ if $\vec{x} \in R_i$

Then if $R_1, \ldots, R_n, f_1, \ldots, f_n$ are primitive recursive, so is f. ∃ (characteristic) is P.R.

<u>Proof</u>: $f(\vec{x}) = \chi_{R_1}(\vec{x}) \cdot f_1(\vec{x}) + \cdots \chi_{R_n}(\vec{x}) \cdot f_n(\vec{x})$

This shows

Let $P(x_1, \ldots, x_n, y)$ be a predicate of $n+1$ variables.

$(\mu y < z) P(\vec{x}, y)$ is defined as follows

Case 1 $(\exists y < z) P(\vec{x}, y)$

Then $(\mu y < z) P(\vec{x}, y)$ is the least z s.t. $P(\vec{x}, z)$

Case 2 o.w. $(\mu z < z) P(\vec{x}, y) = z$ /* Boring case */

<u>Prop</u> If P is primitive recursive and ~~$h(x_1, \ldots, x_n, z) =$~~

$h(x_1, \ldots, x_n, z) = (\mu y < z) P(\vec{x}, y)$

then h is primitive recursive

$h(\vec{x}, 0) = 0$ $(\because \nexists y < 0)$

$h(\vec{x}, St)$ is defined thus:

<u>Case 1</u> $(\exists y < t)$ $P(\vec{x}, t)$

$h(\vec{x}, St) = h(\vec{x}, t)$

<u>Case 2</u> $(\forall y < t) \neg P(\vec{x}, y)$

but $P(\vec{x}, t)$

then $\ldots (t) = t$

Now using this problem 2B is easy

__Prop__ The function $\{i \to p_i\}$ is prim recursive

__Next topic:__ Sequence Numbers

Lemma : $\vec{x} \in \omega^n$ Let $f(\vec{x}, y)$ be prim recursive

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

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

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

Then g, h are prim recursive

$$g(\vec{x}, 0) = 0$$
$$g(\vec{x}, St) = g(\vec{x}, t) + f(\vec{x}, t)$$

$$h(\vec{x}, 0) = 1$$
$$h(\vec{x}, St) = h(\vec{x}, t) \cdot f(\vec{x}, t)$$

Fundamental Theorem of Arithmetic

Let $n \geq 2$ be an integer

Then we can write $n = y_1 \cdot y_2 \cdots y_s$

with $y_1 \leq y_2 \leq y_3 \cdots \leq y_s$ and all y_i's prime

If also $n = z_1 \cdot z_2 \cdots z_t$ with all z_i's prime

then $s = t$

and $y_i = z_i$ for $1 \leq i \leq s$

If $s = \langle a_\emptyset, \ldots, a_n \rangle$

is a finite sequence of non negative integers, we associate to s the sequence number

$\langle\langle a_\emptyset, \ldots, a_n \rangle\rangle$

$\parallel$

$p_0^{a_0+1} p_1^{a_1+1} \cdots p_n^{a_n+1}$

$\emptyset \to 1$

$\langle 2 \rangle \to 8$

$\langle 0 \rangle \to 2$

$\langle 0, 0 \rangle \to 6$

"I feel I should beat the mums about this"

clear that the map from sequences of integers to integers is 1-1

Prop "x is a sequence number" is primitive recursive

Proof x is a sequence number if
$(\forall y \le x)$ (y prime & y divides x
$\to (\forall z \le y)$ (z prime $\to$ z divides x))

Prop Let $n \geq 1$

Then the map $x_0, \ldots, x_{n-1} \to \langle\langle x_0, \ldots, x_{n-1} \rangle\rangle$ is prim recursive

$\langle\langle x_0, \ldots, x_{n-1} \rangle\rangle = p_0^{x_0+1} p_1^{x_1+1} \cdots p_{n-1}^{x_{n-1}+1}$

"Visibly" primitive recursive

MSolovay | M125a | Nov 18 '91 Monday

[C]oding for sequences of integers

$\langle a_0, \ldots, a_{n-1} \rangle$ was coded by $\prod_{i<n} P_i^{a_i+1}$

$$\prod_{i<n} P_i^{a_i+1} = \langle\langle a_0, \ldots, a_{n-1} \rangle\rangle$$

$\langle 1, 1 \rangle$ is coded by 36

Prop There is a primitive recursive function $lh(z)$

$$lh(\langle\langle a_0, \ldots, a_{n-1} \rangle\rangle) = n$$

Following works:

if $z = 0$ $lh(z) = 0$

if $z > 0$, $lh(z) = (\mu i \le z)(P_i \text{ doesn't divide } z)$

Prop There is a primitive recursive function of two variables $(x)_i$

such that if $x = \langle\langle a_0, \ldots, a_{n-1} \rangle\rangle$

and $i < lh(x)$

$(x)_i = a_i$

$$(x)_i = (\mu z \le x)(P_i^z \text{ doesn't divide } x) \dot{-} 2$$

Prop There is a primitive recursive fn of two variables $*$ such that if $a = \langle\langle x_0, \ldots, x_{n-1} \rangle\rangle$ and $b = \langle\langle y_0, \ldots, y_{m-1} \rangle\rangle$

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

$(b)_i + 1$

<u>Dfn</u> If $f : \omega \to \omega$, define $\bar{f} : \omega \to \omega$
by $\bar{f}(n) = \langle\langle f(0), \ldots, f(n-1) \rangle\rangle$

<u>Prop</u> f is prim recursive $\Leftrightarrow$ $\bar{f}$ is

<u>Proof</u> $f(n) = (\bar{f}(n+1))_n$

Conversely if f is prim recursive

$$\bar{f}(0) = 1$$

$$\bar{f}(n+1) = \bar{f}(n) * \langle\langle f(n) \rangle\rangle$$

so $\bar{f}$ is prim recursive

We are going to introduce code numbers (Godel #'s) for various things related to register machines

mistake in notes $\Rightarrow$ "proving Thm not just asleep when I teach"

(1) Coding Instructions

Instruction	Code Number
ADD 1 i	$\langle\langle 0, i \rangle\rangle$
SUB 1 i j	$\langle\langle 1, i, j \rangle\rangle$
HALT	$\langle\langle 2 \rangle\rangle$
TRA i	$\langle\langle 3, i \rangle\rangle$
CPY i j	$\langle\langle 4, i, j \rangle\rangle$
STZ i	$\langle\langle 5, i \rangle\rangle$

② coding Programs

A program is a sequence $\langle I_0, \ldots, I_{n-1} \rangle$ of instructions. Its code # is

$$\langle\langle \#I_0, \ldots, \#I_{n-1} \rangle\rangle$$

where $\# I_j$ is the code no. for I_j

Next we want to introduce numbers that code the instantaneous state of a register machine

This code for state is

$$\langle\langle \# P, c(I), \tilde{c}(R) \rangle\rangle$$

where:

(1) $\# P$ is the code number for the program

(2) $c(I)$ is the contents of the instruction register

(3) $\tilde{c}(R) = \prod_{i \in \omega} p_i^{c(R_i)}$

(Note: This is a finite product since $c(R_i) = 0$ for all but finitely many i)

"there are inf. many so it looks as though it may be infin."

and $p_i^0 = 1$

If all $c(R_i) = 0$ $\tilde{c}(R) = 1$

Let $n \geq 1$

<u>Lemma</u> There is a primitive recursive fn $In(P, x_1, \ldots, x_n)$ such that if P is a code for a program P, $In(P, x_1, \ldots, x_n)$ is the code for the initial register m/c state with program P and inputs $x_1, \ldots, x_n$

<u>Proof</u> $In(P, x_1, \ldots x_n) = \langle\langle P, 0, \prod_{0 < i \leq n} p_i^{x_i} \rangle\rangle$

<u>Prop</u> There is a prim recursive fn $H: \omega \to \omega$ such that $H(s) = 1$ if s is a halted ~~sequence~~ instantaneous description

$H(s) = 0$ if s is a non halted instaneous desc.

<u>Proof</u> s codes a halted machine iff either (a) current instruction is "halt" or (b) construction register does not point to an instruction

<u>Prop</u> There is a prim recursive function $N: \omega \to \omega$ such that if s is a code number for a non halted instantaneous description $N(s)$ is the code number for the "next" register state.

if ~~Ξ~~ s is halted, $N(s) = s$.

<u>Lemma</u> For each $n \geq 1$, there is a primitive recursive function

$S_n(P, x_1, \ldots, x_n, t)$ such that

$$S_n(P, x_1, \ldots, x_n, 0) = I_n(P, x_1, \ldots x_n)$$

$$S_n(P, x_1, \ldots, x_n, t+1) = N(S_n(P, x_1, \ldots x_n, t))$$

Notice that if m/c halts at time t, then for all $s \geq t$,

$S_n(P, x_1, \ldots, x_n, s)$ will be this halted state

<u>Lemma</u> There is a fn U if s is a halted instantaneous state

$U(s) = c(R_0)$ is this state

$U(x) = ((x)_2 \cdot 2)_0$ ← so have exp of $\prod p_i^{c(R_i)+1}$

This moves, or easy to prove yourself

<u>Theorem</u> "Enumeration Theorem" There is a partial recursive function $g : \omega^{n+1} \to \omega$ so that if P is a Gödel number for a program, and $f_{P,n}$ is a partial recursive function of n variables computed by program coded by P, then

$g(P, x_1, \ldots, x_n)$ is defined $\Leftrightarrow$ $f_P(x_1, \ldots, x_n)$ is defined and if either is

and $g(P, x_1, \ldots, x_n) = f_P(x_1, \ldots, x_n)$

RMSolovay MI25a Nov 20 1991 Wednesday

Notation: $s \simeq t$

This means s is defined if and only if t is defined, and if defined they have same value

I'll say "p is a program" rather that "p is a Gödel number for a program"

Enumeration Theorem Let $n \geq 1$. Then there is a partial recursive function

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

such that for any program p, and inputs $x_1, \ldots, x_n$

$E_n(p, x_1, \ldots, x_n) \simeq f_p(x_1, \ldots, x_n)$

We had p.r. functions

$T_n(p, x_1, \ldots, x_n, t)$

(State of machine at time t if started with inputs $x_1, \ldots, x_n$ and program p)

$U(s)$ was contents of R_0 in state s.

$H(s) = 1$ if s is in a halted state

$= 0$ o.w.

$HT(p, x_1, \ldots, x_n) = \mu t[H(T(p, x_1, \ldots, x_n, t))]$

$E_n(p, x_1, \ldots, x_n) \simeq U(T(p, x_1, \ldots x_n, HT(p, x_1, \ldots, x_n)))$

Cor: Every recursive function is μ recursive

Proof: Let $f : \omega^n \to \omega$ be recursive and let k be Gödel number of a fn. that computes f.

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

$$g_1(x_1, \dots, x_n) = \mu T(k, x_1, \dots, x_n)$$

g_1 is ~~total~~ and defined by μ recursion from a p.r. function

So g_1 is μ-recursive

But

$$f(x_1, \dots, x_n) = U(T(k, x_1, \dots, x_n), \mu T(x_1, \dots, x_n)))$$

So f is μ-recursive

$99 \frac{44}{100} \%$

This proof works for almost all styles of machines.

I'll write $\varphi_i(x)$ for $E_1(i, x)$

$$\varphi_i : \omega \to \omega$$

Consider the following set: $K = \{ i \mid \varphi_i(i) \text{ is defined} \}$

Dfn A set of integers A is recursively enumerable if for some recursive predicate $R(x,y)$,

$A = \{ x \mid (\exists y) R(x,y) \}$

Prop K is r.e. (recursively enumerable)

Proof $x \in K \leftrightarrow (\exists y)\ (H(T(x,x,y)))$

Prop K is not recursive

wrong again
youre right

We Suppose χ_K is recursive and derive a contradiction.

Define $g: \omega \to \omega$

$$g(i) = (\mu y)[\,(\chi_K(i) = 1 \wedge (y = y+1))$$
$$\text{or } (\chi_K(i) = 0 \wedge (y = y))\,]$$

If χ_K is recursive, so is g is partial recursive

Say g has a Gödel number
consider g(e)

Case 1 $e \in K$

So $g(e) = (\mu y)(y = y+1)$

So g(e) is undefined. But $g = \varphi_e$

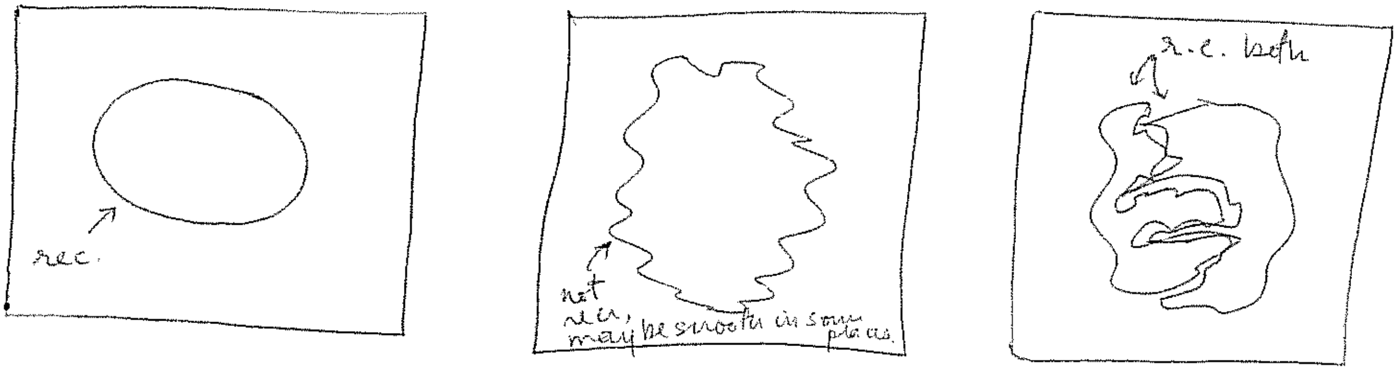
~~But~~ So $\varphi_e(e)$ undefined so $e \notin K$

Case 2 $e \notin K$

So $\chi_K(e) = 0$

$g(e) = \mu y\,(y = y) = 0$

$\therefore$ g(e) is defined



disjoint but horribly intertwined. in particular ∄ recursive set cont one but not other

<u>Theorem</u> There are r.e. sets A, B such that

(1) $A \cap B = \emptyset$

(2) For no recursive set C, is $A \subseteq C$, $B \cap C = \emptyset$

<u>Proof</u> Let $A = \{ i \mid \varphi_i(i) \simeq 0 \}$

$B = \{ i \mid \varphi_i(i) \simeq 1 \}$

A and B are r.e.

$i \in A \Leftrightarrow (\exists y)(H(T_1(i,i,y)) \,\&\, U(T_1(i,i,y)) = 0))$

$i \in B \Leftrightarrow$ - - - - - $= 1 \}$

Suppose toward a contradiction, there is C recursive

with $A \subseteq C$,

$B \cap C = \emptyset$

Let $\chi_C = \varphi i$ for some i

Case 1 $i \in C$ so $\varphi_i(i) = \chi_C(i) = 1$

Case 2 $i \notin C$. So $\chi_C(i) = \varphi_i(i) = 0$

so $i \in A$ But $A \subseteq C$, so $i \in C$; contradiction

$\blacksquare$ QED

RMSolovay | M125a-Lecture | Nov 22 '91 Friday

Gödel Incompleteness Thm

Will prove results for a specific theory P_E (Peano arithmetic, E exponentiation)

1) We will show that there is a sentence Φ such that $P_E \nvdash \Phi$ $P_E \nvdash \neg\Phi$

2) If time, $P_E \nvdash$ "P_E is consistent"

Language of P_E (cf Enderton §3.3)

Predicates: $=, <$

Constant: 0

Unary Fn: S

Binary operations: $+, *, E$

Axioms for P_E (Can't even prove $\forall x\ (\neg x < x)$

A_E: S1-2, L1-3, A1-2, M1-2, E1-2; 11 (conj ⇒ 1)

S1 $\forall x\ (\neg Sx = 0)$

S2 $\forall x \forall y\ (Sx = Sy \rightarrow x = y)$

L1 $\forall x \forall y\ (x < Sy \leftrightarrow (x < y \vee x = y))$

L2 $(\forall x)\ (\neg\, x < 0)$

L3 $\forall x \forall y\ (x < y \vee x = y \vee y < x)$

A1 $\forall x\ (x + 0 = x)$

A2 $\forall x \forall y\ (x + Sy = S(x + y))$

* usual eqns that
* define add from S
* using P.R.

M1 $\forall x\ (x \cdot 0 = 0)$

M2 $\forall x \forall y ((x \cdot Sy) = (x \cdot y) + x)$

E1 $\forall x\ (x E 0 = S0)$

P_E consists of axioms of A_E plus an infinite list of induction axioms

ecall : The closure of a formula φ,

is $\forall x_1 \ldots \forall x_k \varphi$
where $x_1, \ldots, x_k$ are variables free in φ

Induction axioms are all axioms gotten by taking a formula Θ, a variable x free in Θ, and taking closure of

$$[\Theta(0) \wedge \forall x(\Theta(x) \to \Theta(Sx))] \to \forall x\, \Theta(x)$$

..is is quite a powerful theory.
- define all p.r. functions and prove their defining equations

P_E proves there are infinitely many primes.
infact P_E shows "prime number thm" (w/o complex, cauchy, etc)

"out in the wild there yonder"

Mendelson : Ch 3 "Arithmetization of Syntax"

"I owe my soul to Borcherds video"

Going to assign Gödel numbers to symbols of L_{P_E}, sequences of symbols, sequences of seq. of symbols.

$\forall$	3		0	17
$\neg$	5		S	19
$\to$	7		+	21
(	9		$\cdot$	23
)	11		E	25
=	13		v_0	27
<	15		v_1	31

$\boxed{v_i = 27 + 4i}$

(enough spaces left to add stuff)

We will write $\#\forall$ rather than 3

This is arranged so symbols get Gödel #s that are odd and bigger than 1.

Notice every sequence # is 1 or even

so no conflict.

To a sequence of symbols

$\langle s_1, \ldots, s_n \rangle$

assign the Gödel numbers

$\langle \# s_1, \ldots, \# s_n \rangle$

To a sequence of sequences

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

assign the Gödel #,

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

"it takes the spirit of a salesman to hype this up"

"push the liar paradox to arithmetic"

Prop The property "is a variable" is p.r. (the Gödel number of "is a variable")

(Do I mean the symbol v_i or the length 1 seq $\langle v_i \rangle$

↳ Ans: The length 1 sequence)

x is the Gödel # of a variable iff

$$(\exists y < x)\,(x = \langle\langle 27+4y \rangle\rangle)$$

Prop The property "x is the Gödel number of a term" is p.r.

observation $s * t > s, t$ $\qquad (2^2) * (2^3) = 2^2 \cdot 3^2$

Recall if $f : \omega \to \omega$

defined $\bar{f} : \omega \to \omega$

$$\bar{f}(n) = \langle\langle f(0), \dots, f(n-1) \rangle\rangle$$

$$\bar{f}(0) = 1$$

Lemma Suppose $g : \omega^2 \to \omega$ is primitive recursive and $f(n) = g(\bar{f}(n), n)$

then f is primitive recursive

Proof $\bar{f}(0) = 1$

$$\bar{f}(n+1) = \bar{f}(n) * \langle\langle f(n) \rangle\rangle$$

$$= \bar{f}(n) * \langle\langle g(\bar{f}(n), n) \rangle\rangle$$

So $\bar{f}$ is p.r. $\qquad$ so since $\bar{f}$ is p.r.

But $f(n) = (\bar{f}(n+1))_n$

"See one diff between preds & their char fns"

So x is a term $\to$ the Godel # of a

$Tm(x)$ iff

(1) x is a variable

or (2) $x = \langle\langle \# 0 \rangle\rangle$

or (3) $(\exists y < x)\,(Tm(y) \wedge x = \langle\langle \# S \rangle\rangle * y$

or (4) $(\exists y < x)(\exists z < x)\,[Tm(y) \wedge Tm(z) \wedge x = \langle\langle \# + \rangle\rangle * (y * z)$

or (5) similar clause for $\cdot$

or (6) " " E

Thm Let χ_{Tm} be the fn $\chi_{Tm}(x) = 1$ if x is G. #

of a term

$= 0$ otherwise

You can recast this defn as

$\chi_{Tm}(x) = g(\bar{\chi}_{Tm}(x), x)$ for some p.r. g

Now apply preceding lemma

RMSolovay | M125a-Lecture | Adnan Aziz

Mon 9 1:00-X

§2.4 > like what we are doing now

"realizing w/ horror that theres no chalk"

Introduced theory P_E:

1. Introduced Gödel numberings of
   (a) symbols of L_{P_E}
   (b) sequences of symbols
   (c) sequences of sequences of symbols (ie Proofs!!)
2. Proved "x is the Gödel number of a term" is primitive recursive.

Proof of used key idea -
Enough to see $Tm(x)$ can be expressed primitive recursively in terms of $\bar{x}_{Tm}(x)$

Gödels device:
TERM stands for "is the Gödel number of a term"

"Hi I liked your incomp. Thm. keep up the good work"

The following are primitive recursive

"x is an ATOMIC FORMULA"

"x is a WFF" (By induction on x)

Next goal: "x is a SENTENCE"
We are going to reduce this:
We ~~will~~ will first define a substitution function

Proposition There is a primitive recursive function $Sb(a,b,c)$ such that if α is a TERM or WFF and x is a VARIABLE and t is a TERM then $Sb(\alpha, x, t)$ is the Gödel number of result of substituting term coded by t for free occurrances of variables coded by x in the wff or term coded by α.

Basic Idea: define $Sb(a,b,c)$

by induction on a (same trick as for Tm moves

$Sb(a,b,c) = q(\overline{S}b(a,b,c))$

$\overline{S}b(a,b,c) = \underset{i<a}{*} \langle\langle Sb(i,b,c)\rangle\rangle$

<u>Definition</u> (is by cases)

Case 1. a is a variable

$a = b$

$Sb(a,b,c) = c$

case 2. (compound terms or atomic formula)

~~① i is a sequence~~

$(\exists i<a)(\exists k<a)$:

① i is a sequence number

② k is the Gödel number of a predicate or fn symbol

③ $(\forall j < lh(i))((i)_j$ is a TERM) ← not nec $= i$!

④ $a = \langle\langle k\rangle\rangle * \underset{j<lh(i)}{*} (i)_j$ ← j^{th} component!

Length of i

Then $Sb(a,b,c) = \langle\langle k\rangle\rangle * \underset{j<lh(i)}{*} Sb((i)_j, b, c)$

$+ + 0\ 0 \quad 0$

$i_0 = *(+\ 00)$

$i_1 = *\ 0$

$k = *+$

$\langle\langle k\rangle\rangle * i$

$((abc)(def))$

$\neq (abcdef)$

calc LISP

/* Details < p.r. rec. ; official defn of rec. fn : Req M(e*/

Case 3. $(\neg)$

$(\exists i<a)$ (i is a WFF and $a = \langle\langle *(, *\neg\rangle\rangle * i * \cancel{i} \langle\langle *)\rangle\rangle$

Then $Sb(a,b,c) =$
$\langle\langle *(, *\neg\rangle\rangle * Sb(i,b,c) * \langle\langle *)\rangle\rangle$

Case 4. $(\to)$: a isa WFF and

$(\exists i < a)(\exists j < a)$ (i and j are WFFs)

and $a = \langle\langle \# (\rangle\rangle * i *$

$\langle\langle \# \to \rangle\rangle * j *$

$\langle\langle \#) \rangle\rangle$

$Sb(a,b,c) = \langle\langle \# (\rangle\rangle * Sb(i^*, b, c)$

$* \langle\langle \# \to \rangle\rangle * Sb(j^*, b, c)$

$* \langle\langle \#) \rangle\rangle$

Here $i^* = \mu i < a (\exists j < a) \cdot \cdot \cdot \cdot$

$j^* = \mu j < a (\exists i < a)$

where $\cdot \cdot \cdot \cdot$ is

"i & j are WFFs and

$a = \langle\langle \# (\rangle\rangle * i * \langle\langle \# \to \rangle\rangle *$

$j * \langle\langle \#) \rangle\rangle$"

Case 5. $(\exists i < a)(\exists j < a)$

i is a ~~XXXX~~ WFF and

j is a VARIABLE and

$j \neq b$ and

$a =$ ~~XXXX~~ $\langle\langle \# \forall \rangle\rangle * j * i$

$Sb(a,b,c) = \langle\langle \# \forall \rangle\rangle * j * Sb(i,b,c)$

Case 6. o.w.

$Sb(a,b,c) = a$ 　 Covers case $\forall x (\ ) \Leftarrow x$ rules

case constant symbols

RM Solovay M125a Nov 27, 1991 Wednesday

Starts with a BANG!

"Chicago rule - vote early & often"

<u>goal</u> "x is a PROOF" is p.r.

"Puddles out in a pile of trivialities"

x is a PROOF iff

1) x is a sequence number (pf of length 0 not legal)

2) $\ln(x) > 0$

3) $(\forall i < \ln(x))$ $(x)_i$ is a WFF

4) $(\forall i < \ln(x))$ ($(x)_i$ is a LOGICAL AXIOM

or $(x)_i$ is AN AXIOM OF P_E

or $(\exists j < i)(\exists k < i)$ AND

$(x)_i$ FOLLOWS FROM $(x)_j$ $^{\wedge}$ BY MODUS PONENS

(**) x FOLLOWS FROM y AND z BY MODUS PONENS

just means "z" is "y → x")

i.e. $z = \langle\langle ※ (\rangle\rangle * y * \langle\langle ※ \rightarrow \rangle\rangle * x * \langle\langle ※) \rangle\rangle$

is a LOGICAL AXIOM if $(\exists y \leq x)$ [

x is a GENERALIZATION of y and

[y is of TYPE 1 or y is of TYPE 2 .... TYPE 6]]

x is a GENERALIZATION of y (Defined by Ind on x)

if ① x is a WFF or

and ② y is a WFF VARIABLE and $\langle\langle ※ \forall \rangle\rangle * v * z$

Type 2 Axioms

$\forall x\, \alpha \to \alpha^x_t$ (where t is substitutable for x in α).

"quasi subtitable"

t is subs for x in $\beta \to \gamma$ iff it's

subs for x in β and x in γ

t is subs for x in $\forall y\, \beta$

$\Leftrightarrow$ (i) $y = x$

or (ii) y doesn't occur in t & t is subs for x in β

---

x OCCURS IN t

iff ① x is a VARIABLE

② t is a TERM

and ③ $Sb(t, x, \langle\langle \not{x}\, 0 \rangle\rangle) \neq t$

We need the following is primitive recursive

t is SUBSTITUTABLE for x in α

This is routine defn by ind on α, provided that "x OCCURS t" is p.r.

b is an AXIOM OF TYPE 2

if $(\exists \alpha < b)(\exists x < b)(\exists t < b)$

① α is a WFF

② x is a VARIABLE

③ t is a TERM

④ t is SUBSTITUTABLE for x in α

and (5) $b = $ "$\forall x\ \alpha \to \alpha^x_t$" ← idea

formally: $(\exists u < b)\ (u = Sb(\alpha, x, t)$

$\&\ b = \langle\langle \circledast (\rangle\rangle * \langle\langle \forall \rangle\rangle * \ldots$

<u>Lemma</u> If s is a sequence number

and $lh(s) \leq a$

and $(\forall i < lh(s))\ (s)_i \leq a$,

then $s \leq p_a^{a(a+1)}$ ✓

t is a TAUTOLOGY if ~~$\forall s \leq p_{t+1}^{(t+1)(t+2)}$~~ (Extravagant Bound)

(1) t is a WFF

(2) $\left(\forall s \leq p_{t+1}^{(t+1)(t+2)}\right)$ [if s is a Sequence no. and $lh(s) = t+1$ and $(\forall i \leq lh(s))\ ((s)_i \leq 1)$ & s "respects prop. connectives"] $\Rightarrow S_t = 1$]

Three last details

v OCCURS FREE in α

iff v is a VARIABLE

α is a WFF

and $Sb(\alpha, v, \langle\langle \circledast\, 0 \rangle\rangle) \neq \alpha$

α is a SENTENCE iff

α is a WFF &

$(\forall v < \alpha)(v$ is a VARIABLE $\to$ v DOES NOT OCCUR FREE in $\alpha)$

α is a CLOSURE of β iff

α is a WFF and

β is a WFF and

α is a GENERALIZATION of β

and α is a SENTENCE

Rest of details are routine

Ends with a ~~[illegible]~~ whimper!

RM Solovay | M125a Lecture | Dec 2 '91 Monday

O Evans
Mon Dec 9 ← Review
→ rm Session

"running out of later"

Gödel Numbered wffs, proofs

The following are prim recursive

Prf(π, φ)

π is a ~~proof~~ PROOF (in P_E)

and φ is a WFF

and φ is last line of π ← $\varphi = (\pi)_{lh(\pi)-1}$

"principle of dissent first"

Recall we introduced a theory P_E and a finitely axiomatizable subtheory A_E.

Key fact which I will state it precisely now & sketch proof of later

Every recursive function can be "represented" in A_E

By a numeral I mean a term of P_E containing only symbols from $\{S, 0\}$

0 S0 SS0 SSS0 . . . .

For each $x \in \omega$ let $\underline{x}$ be the corresponding numeral

Let be a relation, $R \subseteq \omega^n$

Let $\theta(v_0, \dots, v_{n-1})$ (θ is a wff having as free vars $v_0, \dots v_{n-1}$)

Then θ numeralwise represents R if

$\dots A_E \vdash \neg\theta(\underline{x}_1, \dots \underline{x}_n$

Key fact: R is recursive $\Leftrightarrow$ there is a Θ that numeralwise represents it.

Gödel's First Incompleteness Theorem:

- P_E is not complete

(In fact there is a true sentence saying some machine does not halt which P_E doesn't prove)

Recall φ_i for partial recursive function of one variable with Gödel number i.

$K = \{ i \mid \varphi_i(i) \text{ is defined} \}$

We showed K isn't recursive

Plan: Assuming P_E is complete, we will get an algorithm (recursive procedure) that decides "Is $i \in K$?"

But K is not recursive

So upshot P_E is incomplete

$i \in K \Leftrightarrow (\exists y)\, S(i, y)$

$S(i, y)$: Program i on i/p i is halted at time y.

So there is a formula $\Theta(v_0, v_1)$ that numeralwise represents S.

<u>Lemma</u> $i \in K \Leftrightarrow A_E \vdash (\exists v_1)\, \theta(\underline{i}, v_1)$

<u>Proof</u> ($\Rightarrow$) If $i \in K$, there is a $y \in \omega$,

$S(i, y)$

So $A_E \vdash \theta(\underline{i}, \underline{y})$

So $A_E \vdash (\exists v_1)\, \theta(\underline{i}, v_1)$

PF

$A_E \vdash \theta(\underline{i}, \underline{y}) \to \exists x\, \theta(\underline{i}, x)$

$A_E, \theta(\underline{i}, \underline{y}) \vdash \exists x\, \theta(\underline{i}, x)$

$A_E, \forall x \neg\theta(\underline{i}, x) \vdash \neg\theta(\underline{i}, \underline{y})$

$A_E \vdash \forall x \neg\theta(\underline{i}, x) \to \neg\theta(\underline{i}, \underline{y})$

$\phi \vdash \forall x \alpha \to \alpha^x_t$

$t = \underline{y}$ over here

Let η be the standard model of P_E

$|\eta| = \{0, 1, 2, 3, \dots\}$

non negative integers

Obvious dfns of $\cdot$, $+$, E etc

So $\eta \models P_E$

So $\eta \models A_E$

Notice for each $x \in \eta$ there is a numeral $\underline{x}$ such that $\underline{x}_\eta = x$. (Defining charac of std model!!)

So if $A_E \vdash (\exists v_1)\, \theta(\underline{i}, v_1)$

then $\eta \models (\exists v_1)\, \theta(\underline{i}, v_1)$

So for some $y \in \omega$,

$\eta \models \theta(\underline{i}, \underline{y})$

(Since every elt of η is denoted by a numeral)

Since θ numeral-wise represents S,

Because θ is numeral-wise representing a recursive relation!! Nothing to do w/ A_E being complete

either $A_E \vdash \theta(\underline{i}, \underline{y})$

or $A_E \vdash \neg\theta(\underline{i}, \underline{y})$ ← But this can't happen,

Now assume P_E is complete.

We produce an algorithm that decides "Is $i \in K$?"

Given i

1) First compute the Gödel number of the sentence

$(\exists v_1)\,\theta(\underline{i}, v_1)$

2) Search for the least n such that

(a) n is the Gödel number of a proof in P_E

(b) last line of the proof is "$(\exists v_1)\,\theta(\underline{i}, v_1)$" or "$\neg(\exists v_1)\,\theta(\underline{i}, v_1)$"

Remark: search must converge since P_E is complete

· Output 1 if the last line is $(\exists v_1)\,\theta(\underline{i}, v_1)$

Output 0 if the last line is $\neg(\exists v_1)\,\theta(\underline{i}, v_1)$

Remains to see algorithm gives right answer

If $i \in K$, $A_E \vdash (\exists v_1)\,\theta(\underline{i}, v_1)$

(since $P_E \supseteq A_E$, so $P_E \vdash (\exists v_1)\,\theta(\underline{i}, v_1)$

so since P_E is consistent (it has a model $\mathcal{N}$),

$P_E \nvdash \neg(\exists v_1)\,\theta(\underline{i}, v_1)$)

Very impt: ① can embed P_E in itself. ② set of Axioms was relevant ③ P_E consistent

If $i \notin K$ we have to see

$P_E \nvdash (\exists v_1)\,\theta(\underline{i}, v_1)$

But if $P_E \vdash (\exists v_1)\,\theta(\underline{i}, v_1)$

then $\mathcal{N} \models (\exists v_1)\,\theta(\underline{i}, v_1)$

$A_E \vdash (\exists v_1)\,\theta(\underline{i}, v_1)$

RMSolovay

M125a-Lecture

Adnan Aziz
Dec 4 '91
Wednesday

Thm P_E is incomplete i.e.

There is a sentence σ such that $P_E \nvdash \sigma$ and $P_E \nvdash \neg\sigma$
(If it was complete, could solve halting problem)

Next Result $L = L_{T_E} = \{\ldots, +, \cdot, E, S, 0, <\}$

No decision procedure for telling whether a sentence is logically valid.

Thm (Church) There is no decision procedure for telling whether or not a sentence τ of L is logically valid.

Proof Let $\sigma_1, \ldots, \sigma_{11}$ be the closures of the Axioms of A_E.

Let $\sigma = \sigma_1 \wedge \sigma_2 \wedge \cdots \wedge \sigma_{11}$

By a result of last time,

$i \in K \iff A_E \vdash (\exists v_1)\, \theta(\underline{i}, v_1)$

iff $\sigma \vdash (\exists v_1)\, \theta(\underline{i}, v_1)$

iff $\vdash \sigma \rightarrow (\exists v_1)\, \theta(\underline{i}, v_1)$

iff "$\sigma \rightarrow (\exists v_1)\, \theta(\underline{i}, v_1)$" is logically valid.

So we've reduced question of membership in K to deciding logical validity.

But there is no decision procedure for settling "$i \in K$"?

So none for logical validity

QED

Question For which dialects of f.o.l. is validity decidable

(1) Undecidable if for some $n \geq 2$ there is an n-ary predicate symbol.

(2) Undecidable if for some $n \geq 2$ there is an n-ary function symbol.

3) If there are at least two unary fn symbols, validity is undecidable.

(4) All others are decidable! (only unary Preds → easy. 1 unary fn-harder

Key idea of Gödel's original proof - construct a sentence σ which says "I am unprovable"

Saint Paul's Epistles - "A Cretan says all Cretans are liars"
"I am untrue"
"I am unprovable"

Know - "X is ~~the Gödel number~~ PROVABLE" $(\exists y) Prf(y, x)$

"X is NOT PROVABLE" $\neg(\exists y) Prf(y, x)$

First Try - Try to produce a sentence σ w/ Gödel number e so that σ is $\neg(\exists y) Prf(y, \underline{e})$

Can't be done: because Gödel number will be a lot bigger

if $\underline{e}$ appears in σ, then G# of $\sigma \gg e$.

Next Try Find σ such that

$A_E \vdash \sigma \Leftrightarrow \neg(\exists y) Prf(y, \underline{e})$ where e is the

" "

__Thm__ Let $\Psi(v_0)$ be a formula of L having at most v_0 free. Then there is a sentence σ:

$$A_E \vdash \sigma \Leftrightarrow \Psi(\underline{e}) \quad \text{where } e \text{ is Gödel \# of } \sigma.$$

__Ansatz:__ σ will have the form $\theta(\underline{K})$ where K is the Gödel number of $\theta(v_0)$

"this is exactly the same way Biological Reprod' works"

__Dfn__ Define $h : \omega \to \omega$

(1) If K is the Gödel # of a well formed formula $\theta(v_0)$ having at most v_0 free, $h(K)$ is G# of $\theta(\underline{K})$

(2) o.w. $h(K)$ is G# of $\underline{0=0}$

__Lemma__: h is p.r. (Its defn by cases, wff → pr, subs → pr etc)

Proof is routine & won't be given.

__Lemma__: There is a formula $H(v_0, v_1)$ such that if $i \in \omega$, and $j = h(i)$, then $A_E \vdash (\forall v_1)\, H(\underline{i}, v_1) \Leftrightarrow v_1 = \underline{j}$

σ is going to be of form $\chi(\underline{K})$ for some carefully chosen $\chi(v_0)$ with G.# $\chi = K$

$$\chi(v_0) = (\exists y)\big(H(v_0, y) \wedge \Psi(y)\big)$$

and let $\sigma = \chi(\underline{K})$

To see $A_E \vdash \sigma \Leftrightarrow \Psi(\underline{e})$ where e is G# of σ

σ is $(\exists y)[H(\underline{k}, y) \wedge \Psi(y)]$

But $A_E \vdash (\forall y)\, H(\underline{k}, y) \Leftrightarrow y = \underline{e}$

$A_E \vdash \sigma \Leftrightarrow \Psi(\underline{e})$

Done

"DNA - specially? numbers"

σ "I am provable" (is in fact provable)

Let σ be a sentence s.t. $A_E \vdash \sigma \Leftrightarrow \neg(\exists v_1) Prf(v_1, \underline{e})$

$\boxed{e = G\#\ \sigma}$

Claim If $P_E \vdash \sigma$ then $P_E \vdash$ "$0=1$"

"P_E dare not prove σ on pain of being inconsistent"

Proof If $P_E \vdash \sigma$, let the proof have G# K $\quad P_E \vdash Prf(\underline{K}, \underline{e})$

$P_E \vdash (\exists v_1)\, Prf(v_1, \underline{e})$

But $P_E \vdash \sigma \Leftrightarrow \neg(\exists v_1)\, Prf(v_1, \underline{e})$

so $P_E \vdash \neg\sigma$

so P_E is inconsistent.

Thm Con_{P_E} is

$(\forall v_1)\, \neg Prf(v_1, \#\, 0=1)$

$P_E \vdash$ "If σ is not provable" then "P_E is inconsisti

$P_E \vdash (\forall v_1) \neg Prf(v_1, \underline{e}) \rightarrow \neg Con\, P_E$

RMSolovay

M125a Lecture

Dec 6 '91
Friday

Final: Tuesday Dec 10
12:30 - 3:30
open book, open notes
3106 Etcheverry

Mon Dec 9
1-3 pm
70 Evans

Last lecture on which final will be held:

"Prf (y, x)" is primitive recursive

solovay@math.berkeley.edu 654-3047

We constructed a sentence σ of L_{P_E}

such that $P_E \vdash \sigma \Leftrightarrow \neg(\exists v_0)\, PRF(v_0, \underline{e})$

where e is G.# of σ

Intuitively, σ "says" "I am unprovable"

Far from obvious one can do this

<u>Claim 1</u>: If P_E is consistent, then $P_E \nvdash \sigma$

<u>Proof</u>: ETS $P_E \vdash \sigma$ then P_E is inconsistent

If $P_E \vdash \sigma$ then there is some proof, say with Gödel # y

Let $PRF(v_0, v_1)$ be a formula of L_E that numeralwise expresses $\cancel{PRF}$ Prf (x, y)

If Prf (x, y) $(x, y \in \omega)$ then $P_E \vdash PRF(\underline{x}, \underline{y})$

$P_E \vdash PRF(y, \underline{e})$

So $P_E \vdash (\exists v_0)\, PRF(v_0, \underline{e})$

But $P_E \vdash \sigma \Leftrightarrow \neg(\exists v_0)\, PRF(v_0, \underline{e})$

So $P_E \vdash \neg\sigma$

But $P_E \vdash \sigma$

so P_E is inconsistent.

<u>Claim</u> $P_E \nvdash \neg\sigma$

<u>Proof</u> we know: if P_E proves σ, P_E is inconsistent

But P_E has a model $(\langle \omega, \ldots \rangle)$

so P_E is consistent

so $P_E \nvdash \sigma$

i.e. $\eta \models \neg(\exists v_0)\, PRF(v_0, \underline{e})$

ie $\eta \models \sigma$ (since $P_E \vdash \sigma \Leftrightarrow \neg(\exists v_0)\, PRF(v_0, \underline{e})$ and $\eta \models P_E$)

so if $P_E \vdash \neg\sigma$, $\eta \models \neg\sigma$

upshot: $P_E \nvdash \neg\sigma$

/* Note: $PRF(v_0, \underline{e})$ is
* the wff which
* numeralwise
* represents ...
*/

Want sentences of P_E that express
"P_E is consistent"

$\neg(\exists v_0)\, PRF(v_0, \underline{\# 0=1})$

↑

Con P_E

σ is not provable is expressed by

$\neg(\exists v_0)\, PRF(v_0, \underline{e})$

(where e is G# of σ)

But $P_E \vdash \sigma \Leftrightarrow \neg \exists v_0\, PRF(v_0, \underline{e})$

So in effect, σ says "σ is unprovable"

By formalizing in P_E the proof of Claim 1,

$P_E \vdash \mathrm{Con}\, P_E \to \sigma$

"Every saw the Punchline"
"Went off to a corner for a week or two and convinced myself it was true"

Thm (Gödel's 2nd incompleteness Thm)

If $P_E \vdash \mathrm{Con}\, P_E$, then P_E is inconsistent

(Applies to recursively axiomatizable theories that "contain arithmetic")

"From the paradise of Cantor, no one will drive us out" Dave Hilbert

Set theory + unreachable cardinal
$\to$ Set thy is consistent!!

$\epsilon_0 > \omega^{\omega^{\omega^{\cdots}}}$

1 $P_E \vdash \sigma \Rightarrow P_E$ is inconsistent

2 $P_E \vdash \mathrm{Con} P_E \to \sigma$

↳ done by formalizing in P_E 1

Now suppose $P_E \vdash \mathrm{Con}\, P_E$

Then by 2 $P_E \vdash \sigma$

But then by 1 P_E is inconsistent

QED (G I Thm 2)

A natural example of a sentence which is true, but not provable.

$\mathrm{card}(\mathbb{R}) = \aleph_1$??

$\aleph_0 < \mathrm{card}(\mathbb{R})$ Known

Representability - § 3.3

Adnan
Aziz
10679156

Ans 1 (a) $P \Rightarrow (Q \Rightarrow P)$ ($= \alpha$ wff)

(b) $P \Rightarrow (P \Rightarrow Q)$ ($= \beta$ wff)

(a) is tautolagy

P	Q	$(Q \to P)$	$P \to (Q \to P)$
T	T	T	T
T	F	T	T
F	T	F	T
F	F	T	T

QED (method of truth tables)

(b) is not tautology

consider v: $P \to T$
$Q \to F$

then $\bar{v}(\beta) = T \to (T \to F)$

$= T \to F$

$= F$

so $\{\} \not\models \beta$

QED

1. 25
2. 25
3. 25
4. 25

100

$$\begin{array}{r} 6858 \\ 12880 \\ \hline 19738 \end{array}$$

page

... range over $\{0,1,2,\ldots\}$ const symbols 0 and 1
and binary operations + and ×
we devise formulae to express –

(a) "x is less than y"
Roughly: iff $\exists z$ such that $y = x+z$ and $z \neq 0$
More formally $\exists z((=y+xz) \wedge (\neg(=0z)))$ (Polish used for functions & predicates)

(b) "x is a divisor of y"
Roughly: iff $\exists z$ such that $y = x*z$ (* is × mult symbol)

Formally $\exists z(=y*xz)$

c) x is a prime (Recall x prime $\Leftrightarrow x>1$ and only divisors are itself and 1.)

~~Roughly: iff for all z ~~s.t. ∃y~~ for all y s.t. $x = z*y \Rightarrow ((z=x) \wedge (y=1) \vee (z=1) \wedge (y=x))$~~

~~Formally: $\forall z \forall y (=x*zy \Rightarrow (((=xz) \wedge (=1y)) \vee ((=1z) \wedge (=xy)))$~~
~~(could relax it a bit viz $(z=1) \vee (y=1)$ would also work~~

Roughly: iff $(x \neq 1)$ and (for all z for all y $x = zy \Rightarrow y = x$ or z
and $(x \neq 0)$

Formally
$\underbrace{(\neg = x0)}_{x\neq 0} \wedge \left(\underbrace{(\neg = x1)}_{x \neq 1} \wedge \left(\forall z \forall y \left(\underbrace{x = zy \Rightarrow ((z=x) \vee (y=x))}_{(x=zy \Rightarrow z=x \text{ or } y=x)}\right)\right)\right)$
$\Rightarrow x>1$

A. A212 Page

not logically valid

(a) $\forall x \forall y\, Pxy \Rightarrow \forall y \forall x\, Pxy$

(b) $\exists x \exists y\, Pxy \Rightarrow \exists y \exists x\, Pxy$

(c) $\exists x \forall y\, Pxy \Rightarrow \forall y \exists x\, Pxy$

(d) $\forall x \exists y\, Pxy \Rightarrow \exists y \forall x\, Pxy$

~~(c) is false~~

~~informally if x, y range over {0, 1, 2, ...}~~

~~and P is '≤' (less than or equal to)~~

~~then clearly~~

~~$\exists x \forall y (x \leq y)$~~ (pick $x = 0$)

~~but it is not the case~~ that

~~$\forall y \exists x\, Pxy$~~

(d) is false. (not logically valid) ~~x, y range over~~

informally, if x, y range over $\{0, 1, 2, \ldots\}$ (FOL for number theory)

and P is '<' (less than)

then clearly

$\forall x \exists y (x < y)$ (pick $y = Sx$ (i.e. $x+1$)) is TRUE

but $\exists y \forall x (x < y)$ is False (for any y there is an x which is larger e.g. Sy.

$\therefore$ the FOL formulae is $T \to F$ which is

<u>FALSE</u>

A. Aziz

(Reg)

Ans 4. Let t be a term of the FO language L.
s is a symbol occurring in t.
TPT s begins a subterm of t.

We use induction on the length of the term

$IH(K)$: for all terms t of length $\leq K$, s is a symbol occurring in $t \Rightarrow s$ begins a subterm of t

Base case $K=1$ term is either variable or const.
$\therefore s$ is variable or constant (and hence a term)
$\Rightarrow$ starts a subterm
Base case verified

~~KH~~ Let the IH hold for all terms t of length $\leq K$.

Take any term t of length $K+1$.

Then $t = P t_1 \ldots t_n$

either the symbol s chosen is P itself; in which case the claim holds ($\because$ can take the subterm to be $P t_1 \ldots t_n$ i.e. t itself)

or s is chosen from $t_1 \ldots t_n$.
~~But~~ suppose it is chosen from t_l $1 \leq l \leq n$

Then length of $(t_l) < (K+1)$ ($\because$ length $t = K+1$, and t_l doesn't include P)

Applying IH to t_l, we see s starts a subterm of t_l i.e a finite sequence of consecutive symbols ~~of~~ in t_l → ...ich are ...term

Thus s starts a finite sequence of consecutive symbols in t ~~itself~~ which is a term, i.e. s starts a subterm of t in this case too.

A-A217
(page 4)

Adnan Az

Math. 225
Take-home final

Robert M. Solovay
April 15 1992

This final is due on or before 5 PM May 19, 1992. The best way to hand it in is to deliver it to me personally. You run the risk if you slip it under the door or deliver it to my mail-box that it will not get to me, though I do intend to look in my mailbox before leaving Evans Hall on Tuesday May 19th. I will be in my office (717 Evans) that day from 4–5 PM.

If you absolutely can't contact me, an alternative resource for returning the final is to leave it with my secretary, Catalina Carpenter, 731 Evans, 642-2065. You should make sure that she places it on the chair in my office.

Your work on the final should be individual work. Although I don't think they will be particularly helpful, you may consult the texts of Shoenfield and Enderton on mathematical logic, as well as the book of Tarski, Mostowski and Robinson on Undecidable Theories. If you copy a large portion of a proof from one of these books, you should explicitly note this. You are welcome to consult your course notes as well. Consulting any other book (especially copying a proof verbatim from some other book) will be considered *cheating* and will be dealt with accordingly. It was clear during the fall final that a few people were "solving" problems by the device of wholesale copying. I let it slide then since I had not explicitly forbidden this; I will not let this slide this spring. It is certainly fine to ask me questions concerning the problems on the final either in person or via electronic mail. My email address is solovay@math.berkeley.edu or simply solovay@math if you are mailing from a Berkeley machine.

Mathematics is done with a combination of the conscious and unconscious minds; therefore, it is imperative for you not to leave working on the final to the last minute, since this will allow the unconscious mind no time to contribute. The following point is *very* important. Your answer should not be a faithful rendition of your thought processes (with all the associated false paths and misstarts) but a well-organized presentation of the answer when you finally understand it.

There is another point which I mentioned orally last term, but which was not taken seriously—this caused me considerable discomfort. Finals in pencil or written in a very tiny hand are *unacceptable.* If you are unable to TEX your final, then it is best to write it on every other line of lined paper. Miniscule unreadable handwriting *must* be avoided. Violation of this rule may cause your final to be returned for rewriting or a lowering of your grade.

These caveats are caused by the misbehaviour of a very few people. I hope my complaints about this misbehaviour will not put the rest of you in a sour mood.

Understanding "x" ⇒ can do small variations of it

1. **Variations on the self-referential lemma.**

(a) Show that there is a recursive function of two variables $f(m,n)$ such that:

1. For each fixed m the function $f(m,\cdot)$ is one-to-one;
2. If m is the Gödel number of a formula $\phi(v_0)$ having at most the variable v_0 free, then $f(m,n)$ is the Gödel number of a sentence Φ of L_P such that

$$Q \vdash \Phi \iff \phi(\mathbf{f}(\mathbf{m},\mathbf{n})).$$

Remark: In the preceding formula, we are using the following convention. If e is a non-negative integer, then $\mathbf{e}$ denotes the corresponding numeral, i. e., the term S^e0 of L_P.

hardest (b) The following asks you to prove a two formula version of the self-referential lemma. exactly same as proof of self ref lemma

Let $\phi(v_0,v_1)$ and $\psi(v_0,v_1)$ be formulas of L_P having at most the variables v_0 and v_1 occuring freely in them. Show that there are sentences Φ and Ψ of L_P having Gödel numbers p and q respectively, such that the following equivalences are provable in Q:

$\Phi \iff \phi(\mathbf{p},\mathbf{q})$; $\Psi \iff \psi(\mathbf{p},\mathbf{q})$.

2. **Rosser's Theorem.**

measured ω consist to consis (from Gödel pf)

Let T be a theory in the language of L_P such that the following hold:

1. T is consistent.
2. Every axiom of Q is a theorem of T.
3. T is recursively axiomitizable.

Let R be a sentence of L_P such that the following is provable in Q:

R holds iff the following obtains:

$T \vdash (\exists x\,(Prf_T(x, \#\underline{R}) \to \exists x\, Prf_T(x, \#\underline{\neg R})))$

For any integer q, if there is a proof of R with Gödel number q, then there is a proof of the negation of R with Gödel number less than q. (R is easily constructed by means of the self-referential lemma. This should be clear to you, but you need not argue this point for this final.)

Show that neither R nor its negation is a theorem of T.

Remark: This result of Rosser provides an explicit failure of completeness for T, and improves Gödel's original result since it does not require that T be ω-consistent.

3. **ω-consistency and truth.**

Let P be Peano arithmetic. If Φ is a sentence of L_P, then the theory $P+\Phi$ is the theory whose axioms are those of P together with the sentence Φ.

(a) Show that if Φ is any sentence of L_P, then at least one of the theories $P+\Phi$ and $P+\neg\Phi$ is ω-consistent.

R holds iff "For any integer q if there is a proof of R w/ G # q then there is a proof of the negation of R w/ G # less than q" $(\forall q\in\omega)(\exists x[Prf_T(x, \#\underline{R}) \to \exists y < y<x \wedge Prf_T($

& ¬R holds iff "There is an integer such that

$x^l \ni (1,1,1,\ldots)$

$$F_{n+}^{x+2}(0) = \overbrace{F_{n+}(F_n - - (0)))}^{x+2} > x$$

~~$F_n(0) >$~~

(b) Let $W = \{\#\Phi \mid \Phi$ is a sentence of L_P and $P + \Phi$ is ω-consistent.$\}$. Show that W is definable in the standard model of P. I. e., there is a formula $\phi(v_0)$ of L_P having only v_0 free such that (letting $\mathcal{N}$ be the standard model of P)

$$\mathcal{N} \models \phi(\mathbf{e}) \iff e \in W.$$

(c) Conclude from (b) that there is a sentence Φ such that both $P + \Phi$ and $P + \neg\Phi$ are ω-consistent. Hence, there is a theory T extending P which is ω-consistent, but does not hold in the standard model of P.

Hint for (c): This follows "in one line" from part (b).

4. **Primitive recursive functions.**

Show total?

Consider the following sequence of functions F_n:

"cofinal" in the p.r. fn.

$$F_0(x) = x + 2; \quad F_{n+1}(x) = F_n{}^{x+2}(0).$$

(Here if $G : \omega \to \omega$, then $G^j(x)$ is defined by induction as follows:

every p.r. fn has growth bdd by this expression. By ord ind

$$G^0(x) = x; G^{j+1}(x) = G(G^j(x)).)$$

(a) Show that each of the functions F_j is primitive recursive.

(b) Prove the following facts about the F_j's.

1. $F_j(x) > x$.
2. If $x > y$, then $F_j(x) > F_j(y)$.
3. If $j > k$, then $F_j(x) > F_k(x)$.

(c) Let $H(x_1, \ldots, x_k)$ be primitive recursive. Show that there is an integer j such that

$$F_j(\max(x_1, \ldots, x_k)) \geq H(x_1, \ldots, x_k).$$

Discussion: Define a function F_ω by $F_\omega(n) = F_n(n)$. Then F_ω is a variant of "Ackerman's function". One can show that F_ω is recursive but not primitive recursive. (The fact that F_ω is not primitive recursive follows easily from the preceding exercise.)

The function F_2 has approximately the growth rate of 2^x. The function F_3 grows very rapidly. For example,

$$F_3(1) > 10^{10^{100}}.$$

Perfect mongrel ⇒ 1, 2, 3 short length
4 medium length