$\mathcal{U}^{k+2}(e, y, x_1, \ldots, x_k)$

that is universal for $k+1$-ary Π^0_n relations.
So let:

$\mathcal{W}^{k+1}(e, x_1, \ldots, x_k)$ be $\exists y\, \mathcal{U}^{k+2}(e, y, x_1, \ldots, x_k)$

Then $\mathcal{W}^{k+1} \in \Sigma^0_{n+1}$ (by its defn)
and is universal since if $R \in \Sigma^0_{n+1}$ is k-ary then

$R(\vec{x})$ iff $\exists y\, Q(y, \vec{x})$
(for some $Q \in \Pi^0_n$)
iff $\exists y\, \mathcal{U}^{k+2}(e, y, \vec{x})$
(for some e) ✓

Follows for Π^0_{n+1} ✓ ✓

doesn't contradict Kleene's enum. theorem? why?

<u>Prop'n</u> Δ^0_n does not have enum. prop.

Pf: exercise (diag arg + fact that Δ^0_n closed under complements)

<u>Corollary</u>

- Σ^0_n, Π^0_n <u>not</u> closed under complements
- Σ^0_n <u>not</u> closed under $\forall$
- Π^0_n " " under $\exists$
- $\Sigma^0_n \cup \Pi^0_n \subsetneq \Delta^0_{n+1} \subsetneq$ both Σ^0_{n+1} and Π^0_{n+1}

32

Pf: exercise

recall S^m_n thm

Example Let $\{\varphi_e\}$ be an effective, acceptable enumeration of the (unary) recursive functions

Let $C = \{e \mid \varphi_e \text{ is total}\}$

What is complexity of C?

Can define C by:

$$C = \{e \mid \forall x \exists y\ \varphi(e,x) = y\}$$

"$\forall\exists$ recursive" so Π_2^0 (at most)

i.e. $C \in \Pi^0_2$.

why is this enough

does $C \in$ any lower complexity class?

No: in fact C is Π^0_2-complete

Why: Fix $A \in \Pi^0_2$, $A \subseteq \mathbb{N}$.

WTS $A \leq_R C$

$x \in A \Leftrightarrow \forall y\, B(x,y)$ for some Σ^0_1 B

$\Leftrightarrow \forall y\ f(x,y)\downarrow$ for some partial recursive f

117a

So there is e s.t.

S from S^m_n theorem → is p.r.

$x \in A \Leftrightarrow \forall y\ (\varphi(e,x,y)\downarrow) \Leftrightarrow \forall y\ (\varphi(S(e,x),y)\downarrow)$

so $x \to S(e,x)$ gives reduction from A to C

Post's Theorem

↳ gives a connection between the arithmetic hierarchy and Turing jump.

Theorem (Post): Sps C_n is a complete Σ^0_n set. Then:

(1) $A \subseteq \mathbb{N}^k$ is r.e. in C_n iff $A \in \Sigma^0_{n+1}$

(2) $A \subseteq \mathbb{N}^k$ is recursive in C_n iff $A \in \Delta^0_{n+1}$

View C_n as subset of $\mathbb{N}$.

pf (2) follows from (1) by relativization of fact: A is rec. iff A and A^c both r.e.

if $A \subseteq \mathbb{N}^k$, $A \leq_R C_n$ means we view A as subset of $\mathbb{N}$ via coding.

Pf of (1) ($\Leftarrow$) Sps $A \in \Sigma^0_{n+1}$

– then $A(\vec{x})$ iff $\exists y\, P(y, \vec{x})$ for some $P \in \Pi^0_n$

$\Rightarrow P^c \in \Sigma^0_n$

$\Rightarrow P^c \leq_R C_n$ (Since C_n is Σ^0_n-complete)

$\Rightarrow P^c \leq_T C_n$

$\Rightarrow P \leq_T C_n$ ($P^c \equiv_T P$ always)

So P is rec. in C_n

So A is r.e. in C_n ✓

($\Rightarrow$) Sps $A \in RE(C_n)$ let α denote char. function of C_n

$C_n \in \Sigma^0_n$ means assertion "$\alpha(i) = 1$" is Σ^0_n
hence "$\neg(\alpha(i) = 1)$" is Π^0_n
i.e. "$\alpha(i) = 0$"

Since A is r.e. in C_n we knew from before there is a p.r. T and $e \in \mathbb{N}$ s.t.

$$A(\vec{x}) \text{ iff } \exists m\, T(e, \langle \vec{x} \rangle, \bar{\alpha}(m))$$

equiv to:

$$\exists n \left(Seq(n) \wedge \forall i < length(n) \left((n)_i = \alpha(i) \right) \wedge T(e, \langle \vec{x} \rangle, n) \right)$$

everything after "$\exists n$" is p.r. except for ~~[illegible]~~ clause "$(n)_i = \alpha(i)$"

How complex is this?
Equiv to: $(\alpha(i) = 1 \Rightarrow (n)_i = 1) \wedge (\alpha(i) = 0 \Rightarrow (n)_i = 0)$
equiv: $(\underbrace{\alpha(i) = 0}_{\Sigma^0_n} \vee \underbrace{(n)_i = 1}_{p.r.}) \wedge (\underbrace{\alpha(i) = 1}_{\Pi^0_n} \vee \underbrace{(n)_i = 0}_{p.r.})$

overall $\Sigma^0_n \wedge \Pi^0_n$ which is Σ^0_{n+1}

Hence, " " is "$\exists n\, \Sigma^0_{n+1}$" and since Σ^0_{n+1} closed under $\exists$, is Σ^0_{n+1}
$\Rightarrow A$ is Σ^0_{n+1} ✓

Let $\underset{\sim}{0}^n$ denote $\underset{\sim}{0}^{\overbrace{''\cdots'}^{n\text{-times}}}$.

<u>Corollary</u>: Again letting C_n be any fixed Σ^0_n-complete set, we have
$$[C_n]_T = \underset{\sim}{0}^{(n)}$$
for every $n \in \mathbb{N}$.

<u>Pf.</u> ez induction using prev. theorem.

<u>First-order logic (FOL)</u>

- want to study <u>mathematical structures</u> (sets equipped w/ relations, functions, and constants)

- e.g. structure of arithmetic:
  - $\mathcal{N} = \langle \mathbb{N}, <, S, +, \cdot, 0 \rangle$
    (where S is successor function $S(n) = n+1$)
  - <u>or</u> our favorite group $\mathcal{G} = (G, e, \cdot)$
  - <u>or</u> our fav partial order $\mathcal{P} = (P, \leq)$

aim to formalize:

① how to write statements about a structure (syntax)
② how to evaluate when such a statement is true in a given structure (semantics)
③ when such a statement can be formally proved.

Syntax: first-order languages

Def'n a formal language L consists of

(i) the logical symbols (these belong to every L)

- variable symbols $x_1, x_2, \ldots, y, z$ etc. (ctbly many)
- connectives $\wedge$ ("and") and $\neg$ ("not")
- exists quantifier $\exists$
- equality symbol $=$
- parentheses ( ) and comma ,

(ii) its non-logical symbols (specific to L)

- set of relation symbols $\{R_i\}$ where R_i has arity n_i

- " " function symbols $\{f_j\}$
  where given f_j arity m_j
- " " constant symbols $\{c_k\}$

↳ to specify L enough to specify $\{R_i\}$, $\{f_j\}$, $\{c_k\}$
- write: $L = \{R_i\} \cup \{f_j\} \cup \{c_k\}$
- e.g. lang. of arithmetic is
  $L_{ar} = \{<\} \cup \{S, +, \cdot\} \cup \{0\}$
  $= \{<, S, +, \cdot, 0\}$ where

$<$ is binary relation symbol, $S, +, \cdot$ are binary function symbols, 0 is constant symbol.

The terms of L are defined inductively
(i) variables x_i and constants c_k are terms
(ii) if $t_1, \ldots, t_n$ are terms and f is a n-ary function symbol, then $f(t_1, \ldots, t_n)$ is a term

terms are syntactic, but think: represent outputs of functions that can be defined by iteratively composing the basic functions (represented by) the f_j.

<u>Formulas</u> also defined inductively:

"atomic" formulas
(i) if $t_1, \dots, t_n$ are terms and R is n-ary relation symbol, $R(t_1, \dots, t_n)$ is a term
(ii) if s, t ~~[illegible]~~ terms, $s = t$ is a term

(iii) if φ, χ formulas, so are $\varphi \wedge \chi$, $\neg\varphi$, $\exists x_i \varphi$

Formulas also syntactic - but represent assertions about terms

We'll also use connectives $\vee$ "or"
$\Rightarrow$ "implies"
$\Leftrightarrow$ "iff"

and quantifier $\forall$ "for all"
but these are technically abbreviations:

$$\varphi \vee \chi := \neg(\neg\varphi \wedge \neg\psi)$$
$$\varphi \Rightarrow \chi := \neg\varphi \vee \chi$$
$$\varphi \Leftrightarrow \chi := (\varphi \Rightarrow \chi) \wedge (\chi \Rightarrow \varphi)$$
$$\forall x_i \chi := \neg\exists\neg\chi$$

<u>Ex:</u> Consider $L = \{<, S, +, \cdot, 0\}$
0, x_1, x_5, $S(0)$, $S(S(0))$, $+(x_1, x_5)$ are all terms

We'll abbreviate $S(0)$ as $\underline{1}$
$S(S(0))$ as $\underline{2}$ etc.

Write $x_1 + x_5$ for $+(x_1, x_5)$
$x_1 \cdot x_5$ for $\cdot(x_1, x_5)$ etc.

other similar abbreviations, $x^3 = x \cdot (x \cdot x)$

A more complicated term: $\underline{2}x^3 + \underline{5}x + \underline{1}$
(terms in this lang rep multivar poly's w/ integer coeff's)

$0 = x_3$; $0 < S(0)$; $x^3 + z = y^2$
are all atomic formulas

$\forall x (0 < x \Rightarrow \exists y (S(y) = x))$ is a non-atomic fmla.

- want to define when a given fmla φ is T/F in a given structure.
- not all fmlas intuitively have a truth value, e.g. $0 < x$ depends on value of x
- leads to concept of free and bound variables.

Def'n if fmla $\exists x\, \varphi$, the scope of $\exists x$ is φ

Def'n an occurrence of a variable x is a formula φ is bound if it occurs in a quantifier $\exists x$ or in the scope of a quantifier in a subformula $\exists x\, \psi$. Otherwise it is free

e.g. in $\forall y \exists z\, (y < x \wedge x < z)$
y, z are bound
x is free

(intuition: free vars are those we need to specify before fmla becomes T or F)

Def'n φ is a sentence if it has no free variables.

We write $\varphi(x_1, \dots, x_n)$ to indicate free variables of φ are among $x_1, \dots, x_n$
(some might be bound or not occurring)