<u>Composition</u> $R \circ S$ (from the quizzes)

$= \{ u \circ v : u \in R \text{ and } v \in S \}$

$R = \{ \text{cat}, \text{dog} \}$, $S = \{ \text{purr}, \text{growl} \}$

$R \circ S = \{ \text{catpurr}, \text{dogpurr}, \text{catgrowl}, \text{doggrowl} \}$.

<u>Kleene star</u> $R^* = \{ w_1 w_2 \cdots w_\ell : w_i \in R, \forall i \ \& \ \ell \geq 0 \}$

<u>Ex</u> $R^* = \{ \epsilon, \text{cat}, \text{dog}, \text{catcat}, \text{catdog}, \ldots \}$

<u>Thm</u> If R is c.e., then R^* is c.e.

(Exercise)

Thm: Let Γ be a c.e. set of sentences

Let $T = Cn\,\Gamma$ = consequences of Γ

$= \{A : \Gamma \models A,\ A \text{ is a sentence}\}$

Then if T is complete, T is decidable.

Proof: $\boxed{\text{Since } \Gamma \text{ is c.e., } T \text{ is c.e.}}$

Suppose T is consistent. So $A \in T$ iff $\neg A \notin T$ since T is complete.

Input A

Algorithm If A is not a sentence, reject.

Enumerate T (with our algorithm that enumerates T)

If A appears, accept.

If $\neg A$ appears, reject.

This algorithm decides T.

Now suppose T is inconsistent. - Then $T = \{\text{sentences } A\}$ is certainly decidable.

Examples

$Cn\{AtLeast_k : k \geq 2\}$ is complete.

And $\{AtLeast_k : k \geq 2\}$ is decidable (hence c.e.)

So $Cn\{AtLeast_k : k \geq 2\}$ is decidable.

Example Dense Linear Order without endpoints.

is complete + has a finite set of axioms.

So it is decidable.

(Non)Example $Th\,\mathfrak{N}$ is certainly complete. $\mathfrak{N} = (\mathbb{N}, 0, +, \cdot, S)$

Later: $Th\,\mathfrak{N}$ is not decidable.

Hence $Th\,\mathfrak{N}$ is not axiomatizable. (!!)

Claim: For any structure $\mathfrak{A}$, $Th\,\mathfrak{A}$ is complete.

~~But $Th\,\mathfrak{A}$ is not decidable, so $Th\,\mathfrak{A}$ is not axiomatizable~~

or decidable

Def'n A theory T is axiomatizable if there is a c.e. set T' such that $Cn\,T' = T$.

Proving undecidability

- Need a formal definition of algorithm
- Church-Turing thesis has consequences:

(1) Uniform representation of algorithms as strings, members of Γ^*, $\Gamma = \{0,1\}$ w.l.o.g

ulcorner M urcorner

$\ulcorner M \urcorner$ - Gödel number is the string ~~a~~ describing the finite set of an ambiguous instruction for algorithm M.

(2) Universal algorithm. $U(\ulcorner M \urcorner, w)$ $\quad w \in \Gamma^*$, $\ulcorner M \urcorner \in \Gamma^*$

$U(\ulcorner M \urcorner, w)$ - simulates M on input w

$U(\ulcorner M \urcorner, w)$ - halts/accepts/rejects/outputs the same that $M(w)$ does.

"Compiler" "interpreter".

(3) Malleability of algorithms

Given a $\ulcorner M \urcorner$ for algorithm M,

can modify $\ulcorner M \urcorner$ to form $\ulcorner N \urcorner$ for a related algorithm N.

For instance, N might accept iff M rejects

and reject iff M accept.

Another example compose 2 algorithms.

<u>Here</u> M_1, M_2 compute unary functions f_1, f_2

Let $f = f_2 \circ f_1$

Let M compute f.

$\langle \ulcorner M_1 \urcorner, \ulcorner M_2 \urcorner \rangle \mapsto \ulcorner M \urcorner$ is a computable function

Theorem (a) There is a $X \subseteq \mathbb{N}$ which is not decidable

(b) For any Σ, there is an $X \subseteq \Sigma^*$ which is not decidable

Σ-nonempty!

Proof: There are countable many members of T^*, so countably many algorithms (uniform representability)

But there are uncountably many $X \subseteq \mathbb{N}$.

So "most" $X \subseteq \mathbb{N}$ are undecidable.

This proves (a).

(b) is similar.

qed

Theorem The set $\{X : X \subseteq \mathbb{N}\}$ is uncountable.

Pf Cantor's Diagonal Argument:
Proof by Contradiction. Suppose $X_0, X_1, X_2, \ldots$ enumerates the power set $\{X : X \subseteq \mathbb{N}\}$ of $\mathbb{N}$.

	0	1	2	3	4	...
X_0	1	0	1	0	1	...
X_1	1	1	0	1	0	...
X_2	0	1	0	0	1	...
X_3	1	0	0	1	0	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
Y	0	0	1	0		...

A "0" in row X_i column j means $j \notin X_i$. A "1" means $j \in X_i$.

$i \mapsto X_i$ as a set in a nonstd model of ZFC, may not exist.

$Y = \{i : i \notin X_i\}$ $\qquad i \in Y \iff i \notin X_i$

Hence $Y \notin \{X_0, X_1, X_2, \ldots\}$ since $i \in Y \iff i \notin Y$ is impossible.

<u>Next</u>: The Halting Problem is not decidable.