Composition: \( R \circ S \) (from the quizzies)

\[
R = \{ \text{cat, dog} \}, \quad S = \{ \text{purr, grow} \}
\]

\( R \circ S = \{ \text{catpurr, dogpurr, catgrow, doggrow} \} \)

Kleene star: \( R^* = \{ w_1 w_2 \cdots w_k : w_i \in R, t_i \text{ & } k \geq 0 \} \)

Example: \( R^* = \{ \epsilon, \text{cat}, \text{dog}, \text{catcat}, \text{catdog}, \ldots \} \)

Theorem: If \( R \) is c.e., then \( R^* \) is c.e.

(Exercise)
**Theorem:** Let $T$ be a c.e. set of sentences

Let $T = Cn T = \text{consequences of } T$

$= \{ A : T \vdash A, \ A \text{ is a sentence} \}$

Then if $T$ is complete, $T$ is decidable.

**Proof:** Since $T$ is c.e., $T$ 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 $T = \{ \text{sentence} A \}$

is certainly decidable.
Examples

$\text{Cu } \exists \text{AtLeast}_k : k \geq 2$ is complete.

And $\exists \text{AtLeast}_k : k \geq 2$ is decidable (hence c.e.)

So $\text{Cu } \exists \text{AtLeast}_k : k \geq 2$ is decidable.

Example Dense Linear Order without endpoints.

It is complete and has a finite set of axioms.

So it is decidable.

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

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

Hence Th $\mathbb{N}$ is not axiomatizable.

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

But Th $\mathcal{A}$ is not decidable, so Th $\mathcal{A}$ is not axiomatizable.

A theory $T$ is axiomatizable if there is a c.e. set $T'$ such that $\text{Cu } T' = T$. 
Proving undecidability:

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

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

2. Gödel number is the string describing the finite set of unambiguous instructions for algorithm \( M \).

2. Universal algorithm. \( U(\overline{M^7}, w) \) \( w \in \mathbb{T}^* \), \( \overline{M^7} \in \mathbb{T}^* \)

\( U(\overline{M^7}, w) \) - simulates \( M \) on input \( w \)

\( U(\overline{M^7}, w) \) - halts/accepts/rejects/outputs the same that \( M(w) \) does.

"Compiler"/"interpreter".
Given a $\Gamma_M$ for algorithm $M$, can modify $\Gamma_M$ to form $\Gamma_N$ for a related algorithm $N$.

For instance, $N$ might accept if $M$ rejects and reject if $M$ accepts.

Another example concerns 2 algorithms.

Here $M_1$, $M_2$ compute any functions $f_1$, $f_2$.

Let $f = f_2 \circ f_1$.

Let $M$ compute $f$.

$(\langle M_1, M_2 \rangle \implies \Gamma_M)$ is a computable function.
There is a $X \leq \mathbb{N}$ which is not decidable.

For any $\Sigma$, there is an $X \leq \Sigma^*$ which is not decidable.

Proof: There are countably many members of $\mathbb{T}^*$, so countably many algorithms (uniform representability).

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

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

This proves (c).

(b) is similar.

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

Proof: (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 $\mathbb{N}$.

Let $\mathcal{Y} = \{X_i : i \notin X_i\}$.

Hence $\mathcal{Y} \notin \{X_0, X_1, X_2, \ldots\}$, since $i \notin \mathcal{Y}$ if and only if $i \notin X_i$.

A "0" is row $X_i$.
A "1" is column $j$.

Let $\mathcal{Y}$ be a set in a non-standard model of ZFC, meaning it exists.
Next: The Halting Problem or not decidable.