Encode formulas or sets of formulas over a finite alphabet:

\[ \Sigma_{\text{prop}} = \{ \neg, \wedge, \vee, \rightarrow, \leftrightarrow, (, ), \text{p, } 0, 1, \ldots, 9 \} \]

Example: \( p \lor \neg p \) is encoded by \((p \lor \neg p) \in (\Sigma_{\text{prop}})^*\)

For sets of formulas:

\[ \Sigma_{\text{prop}^+} = \Sigma_{\text{prop}} \cup \{ , \} \]

\( \{ p_1, p_2, p_3 \} \) encoded by \( p_1, (p_2 \lor p_3) \in (\Sigma_{\text{prop}^+})^* \)

Theorem: The set of syntactically valid propositional formulas is decidable.

Here we are conflating formulas and the strings in \((\Sigma_{\text{prop}})^*\) that represent them.
Then the set of tautologies is decidable.

Pf: the method of truth tables.

Thus the set of finite sets \( T \) of syntactically correct formulas is decidable.

Thus the binary relation \( \langle TT, A \rangle \) of

\( TT \) is a finite set of formulas, \( A \) is a formula and \( TT \vdash A \) is decidable.

The set of satisfiable set \( TT \) of sentences is decidable.

\( \vdash \) contradictory sets.

Pf: Method of truth tables.
Thus: Suppose $T$ is a c.e. set of formulas then, the $\forall A: T \vdash A \equiv$ c.e.

Pf: Algorithm that semi-decides this set is

**Assumption:** $M$ enumerates $T$

**Input:** $A \ (w \in (Σ^pr)^*)$

**Algorithm**

If $w$ does not correctly encode a formula, reject.

For $i = 0, 1, 2, 3, \ldots$

- Run $M$ until it enumerates $i$ member of $T$.
- Let $T_i$ be these first $i^*$ formulas.
- If $\exists M_i \vdash A$, accept (and halt).

End-for.

Algorithm semi-decides $\forall A: T \vdash A \equiv$ by the

Completeness Theorem.
Above theorem cannot be strengthened for decidable $TT$ to conclude that $\forall A: TT \vdash A$ is decidable.

Example Let $A$ be $P_1$.

Suppose $TT$ might contain one of $P_1 \lor P_1 \land P_1 \land P_1$ or

$\forall_{j=1}^{k} P_i$.

Exercise If $TT$ is c.e., there is a decidable $T$ such $T \vdash \# TT$ (Craig’s Theorem)
Algorithm for first-order logic

Encode first-order formulas on alphabet \( \Sigma \) (for L-attribute language)

\[
\Sigma_{\text{fo-2}} = \Sigma \cup \{ \neg, \rightarrow, \forall, (, ), x, 0, \ldots, n \} = \Sigma \cup L
\]

\[
\forall x_i (x_i = f(x_i)) \quad \forall x_1 x_1 = f(x_1)
\]

Encode finite sets of first-order formulas using

\[
\Sigma_{\text{fo-2}^+} = \Sigma_{\text{fo-2}} \cup \{ \exists \}
\]

The set of syntactically correct \( L \)-formulas is decidable

Encode proofs in \( FO \), by a sequence of formulas separated by commas as \((\Sigma_{\text{fo-2}^+})^*\) - string
Thus: The binary relation \( \langle T_I, P \rangle \) of pairs \( T_I \) and \( P \) such \( P \) is a valid FO proof from the hypotheses \( T_I \) is decidable.

Thus: the set of valid L-formula is c.e.

**Pf**: A is valid iff A has a proof.

**Idea**: Search for a proof\( A \). Accept if a proof is found.

**Input**: A

**Algorithm**

\[
\text{For } \ i = 1, 2, 3, \ldots \\
\text{Let } v_i \text{ be the } i^{th} \text{ member of } (\Sigma^*)^* \\
\text{If } v_i \text{ encodes a valid proof } P \text{ of } A, \\
\text{ accept (and halt)}
\]

End-for
The set of pairs \( \langle T, A \rangle \) s.t. \( T \) is a finite set of formulas, \( A \) is a formula, and \( T \vdash A \), is computably enumerable.

**Pf:** Algorithm

Enumerate all pairs \( \langle T', P \rangle \) s.t. \( P \) is a valid proof from hypotheses \( T' \). If \( T' = T \) and \( A \) is the final formula in \( P \), accept.
Theorem Let $T$ be a c.e. set of formulas.
Then $\exists A : T \vdash A$, $A$ is a formula) is c.e.

Proof Algorithm idea: Start enumerating $T$ and all possible proofs (above tainting) and watch for a proof of $A$ from a finite subset of $T$.

Assumption: $M$ enumerates $T$

Input: $A$

Algorithm

For $i=1,2,3,\ldots$

Run $M$ for $i$ steps, let $T_i$ be the subset of $T$ enumerated by $M$ within $i$ steps.

For $j=1,2,3,\ldots i$

Let $y_j$ be the $j$-th member of $(\Sigma^* \cup)^*$

If $y_j$ runs codes a valid proof of $A$ from $T_i$, accept

End for

End for
Theorem: If $T$ is a complete, c.e. theory, then $T$ is decidable.

(In the PDF.)