\[ A_{TM} = \{ <M, w> | M \text{ is a TM that accepts } w \} \]

\( A_{TM} \) is not decidable.

Proof by contradiction

Suppose \( A_{TM} \) is decidable and \( H \) is the TM that decides it.

\[ H(<M, w>) = \begin{cases} 
\text{accept when } M \text{ accepts } w \\
\text{reject when } M \text{ does not accept } w
\end{cases} \]

Construct \( D \), a TM, that uses TM \( H \).

\( D \): on input \(<M>\) where \( M \) is a TM

1. run \( H \) on input \(<M, <M>>\)
2. output the opposite of \( H \).
   - if \( H \) accepts \( \Rightarrow D \) reject
   - if \( H \) rejects \( \Rightarrow D \) accept
run $D$ on input $\langle D \rangle$

$$D(\langle D \rangle) = \begin{cases} 
\text{accepts} & \text{if } D \text{ accepts } \langle D \rangle \\
\text{rejects} & \text{if } D \text{ does not accept } \langle D \rangle 
\end{cases}$$

a contradiction.

So $H$ cannot exist.

$\therefore A_{TM}$ is not decidable.
co-Turing recognizable
if its complement is
Turing recognizable.

$L$ is recognizable
$T \text{ rec. } L$

$L$ is co-Turing rec., $R \text{ TM}$

A language is decidable iff
it is Turing rec. and
co-Turing rec.

$L$ is recog.
and $\overline{L}$ is recog.

$A_{\text{Tm}}$ is not decidable
$A_{\text{Tm}}$ is Turing recognizable.

$A_{\text{Tm}}$ is not co-Turing recog.
so $\overline{A_{\text{Tm}}}$ is not Turing recog.
Rules

\[ S \to aSa \]

CFG:

LHS: 1 variable
RHS: string of vars and term.

\[ \alpha A \beta \to \alpha X \beta \]

**See:** \( X \neq \varepsilon \)

\( \alpha, \beta, X \) are strings of vars and terms

\( A \): variable

\[ L = \{ a^n b^n c^n | n \geq 1 \} \]

\[ S \to S_a BC S / S_a BC \]

\[ S_a \to ABC S / ABC \]

\[ BA \to AB \quad aA \to aa \]

\[ CA \to AC \quad aB \to ab \]

\[ CB \to BC \quad bB \to bb \]

\[ S_a \to a \quad bC \to bc \]

\[ C \to cc \]

---

\[ S \]

\[ S_a BC S / S_a BC \]

\[ aBC S / aBC ABC \]

\[ aBAC BC \]

\[ aABC BC \]

\[ aaBC BC \]

\[ aaBB CC \]

\[ aabBC \]

\[ aaabbccc \]