Thm 4.1! (Section 4.2): \( A_T = \{ \langle M, w \rangle \mid w \text{ is accepted by } M \} \) is recognizable but not decidable.

\[
\text{Thm 4.1! (Section 4.2): } A_T = \{ \langle M, w \rangle \mid w \text{ is accepted by } M \} \\
\text{is recognizable but not decidable.}
\]

(1) \( L = \{ \langle 0^n \rangle \mid n \in \mathbb{N} \} \) is Turing decidable

Decidable: A language is Turing-decidable if there is a T.M., \( M \), that on input \( \omega \in \Sigma^* \), \( M \) accepts \( \omega \) if \( \omega \in L \)

\( M \) rejects \( \omega \) if \( \omega \notin L \)

Remark: A Turing machine on a given input can

1. accept the input \( \omega \) if it reaches \( q_{\text{accept}} \) (finite number of steps)
2. reject \( \omega \) if it reaches \( q_{\text{reject}} \) (infinite number of steps)
3. loops or does not halt, i.e., runs forever, never reaching \( q_{\text{accept}} \) or \( q_{\text{reject}} \)

(2) \( \text{PALINDROME} \) is Turing decidable

To describe a T.M., you usually:

- give a high level description (describes how many tapes, various phases, etc.,
- give an "implementation"

Specify \( \delta : \)

\[
\delta(q_{\text{init}}, s) = (q_{\text{halt}}, 2, \lambda)
\]

- 2\( \rightarrow 2\), \( \rightarrow \)
- \( \lambda \rightarrow 0 \), \( \rightarrow \)
- \( \rightarrow \), etc.

Say 2-tape

- \( \text{Palindromes} \)
\[ \delta(q_{\text{init}}, 1, 2) = (q_1, 3, 1, R, S) \]

\[ 3, L \rightarrow 3, 1, R, S \rightarrow q_1 \]

\[ q_{\text{init}} \]

\[ \delta : Q \times \Gamma^2 \rightarrow Q \times \Gamma^2 \times \{L, R, S\}^2 \]

\[ \delta(q_{\text{init}}, x, y) = (q_{\text{init}}, x, y, R, S) \text{ if } x \neq \# \text{ and } y \text{ any} \]

In 4.2, the textbook describes a universal Turing machine in one sentence 😞

Warm up: Graph \((V, E)\), \(\langle \text{graph} \rangle\) — give \# vertices, a list of edges

\[
\begin{align*}
&4 \neq 1 \# 3 \\
&1 \# 4 \neq 3 \# 4 \\
&1 \neq 4 \neq 2 \# 3 \neq 2 \# 4 \neq 3 \# 4 \\
&\neq 1 \# 4
\end{align*}
\]

(lexicographical order to get a unique representation)
(1) Let \( L = \{ \langle G \rangle \mid G \text{ is a graph that can be 3-coloured, i.e.} \) 
\[ \text{there exist } V \models \{1, 2, 3\} \text{ such that each edge has different colours} \] 

started to give an algorithm to find bleh bleh

(2) Let \( L = \{ \langle D, w \rangle \mid D \text{ is a DFA}, w \text{ is an input to the DFA} \) 
\[ \text{that is accepted by the DFA} \] 

\( \Delta_{DFA} \) 

(\( q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \rightarrow q_5 \rightarrow q_6 \)) 
\[ q_{init} = 1, \quad q_1 = 2, \quad q_3 = 3, \quad q_{init}, \] 
\[ \text{describe \( \delta \) \( f \) \( 1 \)} \] 
\[ \text{describe input,} \] 
\[ \text{Section 4.1 of Sipser} \]

(3) Last Friday: \( \langle M, w \rangle \) description of a T.m. and \( w \) input

(\( Q, \Sigma, \delta, \text{acc}, \text{ rej}, \text{ init} \)) 
\[ \text{27 # 3 # 5 # long description} \] 
\[ \text{of} \] 
\[ \# 2 \# 1 \# 3 \]
Claim: There is a "universal Turing machine," $U$, such that 

on input $\langle M, w \rangle$, $U$ stops in $q_{acx}$ if $M$ accepts $w$

$U$ stops in $q_{rej}$... reject $w$

$\text{time...}$ $U$ doesn't halt if $M$ doesn't halt

($C_1 + \text{poly time}$) on input $w$

debugger

Describe input describe $M$ describe $w$

$\text{tape 1 } \rightarrow$

$\text{tape 2 } \rightarrow$ copy input

(simulation of the $\text{tape 2 } \rightarrow$)

where the tape head is

$\text{tape 2 } \rightarrow$ copy input

$\text{future } \rightarrow$

look here look here