Today's learning goals

- Trace the computation of a Turing machine on given input
- Describe the language recognized by a Turing machine
- Give an implementation-level description of a Turing machine
- Determine if a Turing machine is a decider

(Group HW4: Ch2 - due Sat)
Turing machine computation

• Read/write head starts at leftmost position on tape
• Input string written on leftmost squares of tape, rest is blank
• Computation proceeds according to transition function:
  • Given current state of machine, and current symbol being read
  • the machine
    • transitions to new state
    • writes a symbol to its current position (overwriting existing symbol)
    • moves the tape head L or R
• Computation ends if and when machine enters either the accept or the reject state.
Language of a Turing machine

$L(M) = \{ w \mid \text{computation of } M \text{ on } w \text{ halts after entering the accept state} \}$

i.e. $L(M) = \{ w \mid w \text{ is accepted by } M \}$
Language of a Turing machine

\[ L(M) = \{ w \mid M \text{ accepts } w \} \]

Which of the following is not always true?

A. If \( w \) is in \( L(M) \) then the computation of \( M \) on \( w \) halts and accepts.

B. If the computation of \( M \) on \( w \) halts and rejects, then \( w \) is not in \( L(M) \).

C. If \( w \) is not in \( L(M) \) then the computation of \( M \) on \( w \) halts and rejects.

Different from other models!
Computation of a Turing machine

To trace DFAs: enough to list states.
To trace NFAs: tree of possible current states (incl. spontaneous moves)
To trace PDAs: tree of possible computations incl. state + stack

- Current state ✔
- Current tape contents up to (finite) point after which all blank
- Current location of read/write head

current state is q
current tape contents are uv (and then all blanks)
current head location is first symbol of v
Special configurations

For input string $w$

- Starting configuration $q_0 \ w$
- Accepting configuration $u \ q_{acc} \ v$
- Rejecting configuration $u \ q_{rej} \ v$

Current state is $q$
Current tape contents are $uv$ (and then all blanks)
Current head location is first symbol of $v$
Language of a TM

$L(M) = \{ \text{w} \mid \text{M accepts w} \}$

= $\{ \text{w} \mid \text{there is a sequence of configurations of M where C}_1 \text{ is start configuration of M on input w, each C}_i \text{ yields C}_{i+1} \text{ and C}_k \text{ is accepting configuration} \}$

"The language of M"

"The language recognized by M"
Language of a TM

- L is **recognized** by M if $L = L(M)$
- L is **decided** by M if $L = L(M)$ and each computation of M halts, i.e. enters a halting configuration in finite time.

**Challenge**: Convert the TM on the handout to one that recognizes the same language but does not decide it.
An example

$L = \{ w#w \mid w \text{ is in } \{0,1\}^* \}$

We already know that $L$ is

- not regular
- not context-free

We will prove that $L$ is

the language of some Turing machine

(and even is decided by Turing machine)
Implementation-level description

\[ L = \{ w#w \mid w \text{ is in } \{0,1\}^* \} \]

Idea for Turing machine

- Zig-zag across tape to corresponding positions on either side of '#' to check whether these positions agree. If they do not, or if there is no '#', reject. If they do, cross them off.
- Once all symbols to the left of the '#' are crossed off, check for any un-crossed-off symbols to the right of '#': if there are any, reject; if there aren't, accept.
**Implementation-level description**

Zig-zag across tape to corresponding positions on either side of '#' to check whether these positions agree. If they do not, or if there is no '#', *reject*. If they do, cross them off.

Once all symbols to the left of the '#' are crossed off, check for any un-crossed-off symbols to the right of '#': if there are any, *reject*; if there aren't, *accept*.
\[ Tm (Q, \Sigma, \Gamma, \delta, q_1, q_{accept}, q_{reject}) \]

Formal definition

\[ Q = \{ q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_{accept}, q_{reject} \} \]

\[ \Sigma = \{ 0, 1, \# \} \]

\[ \Gamma = \{ 0, 1, \#, x, _ \} \]

All missing transitions have output \((q_{reject}, _, R)\)

Fig 3.10 in Sipser
Computation on input 0#0?
Computation on input 0#?
For next time

**GroupHW4 due** Saturday, February 17

For Friday, pre-class reading: pp. 184-185