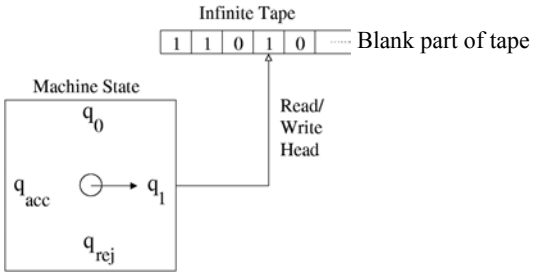
Turing Machines Review

- Turing Machine Basics
- Decidable versus Recognizable Languages
- An example of a decidable language that is not a CFL
  - Implementation-level description of a TM
  - State diagram of TM

Turing Machines



Just like a DFA except:

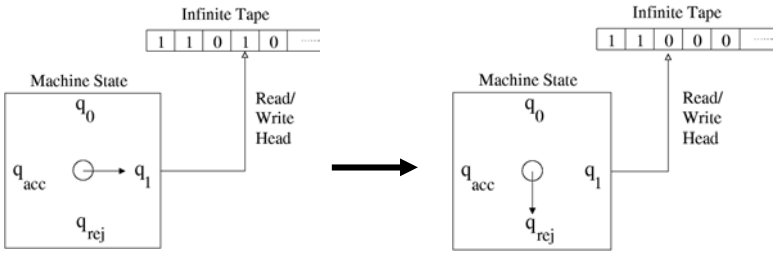
- You have an infinite "tape" memory (or scratchpad) on which you receive your input and on which you can do your calculations
- You can read one symbol at a time from a cell on the tape, write one symbol, then move the read/write pointer (head) left (L) or right (R)

Who was Turing?

- Alan Turing (1912-1954): one of the most brilliant mathematicians of the 20th century (one of the "founding fathers" of computing)
- Click on "Theory Hall of Fame" link on class web under "Lectures"
- Introduced the Turing machine as a formal model of what it means to compute and solve a problem (i.e. an "algorithm")
  - Paper: `On computable numbers, with an application to the Entscheidungsproblem, Proc. London Math. Soc. 42 (1936).`

How do Turing Machines compute?

- δ(current state, symbol under the head) = (next state, symbol to write over current symbol, direction of head movement)



- Diagram shows: $\delta(q_1,1) = (q_{rej}, 0, L)$ (R = right, L = left)
- In terms of "Configurations": $110q_1\underline{1}0 \Rightarrow 11q_{rej}\underline{0}00$

On-Board Discussion of Turing-Recognizable versus Decidable Languages

How does a TM accept a string?

How can a TM reject a string?

What is a decider TM?

Solving Problems with Turing Machines

- We know L = $\{0^n1^n0^n \mid n \geq 0\}$ is not a CFL (pumping lemma)
- Show L is decidable
  - Construct a decider M such that L(M) = L
  - A decider is a TM that always halts (in q_{acc} or q_{rej}) and is *guaranteed not to go into an infinite loop for any input*

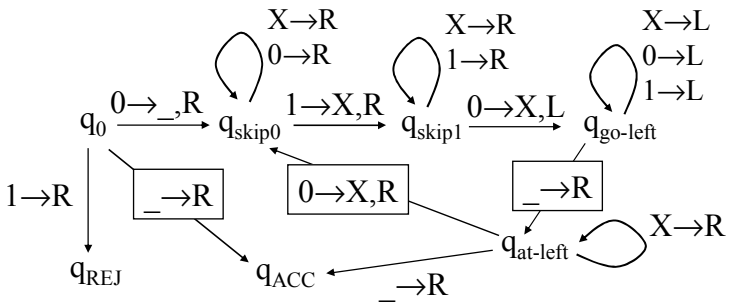
Idea for a Decider for $\{0^n1^n0^n \mid n \geq 0\}$

- **General Idea**: Match each 0 with a 1 and a 0 following the 1.
- Implementation Level Description of a Decider for L:

  On input w:

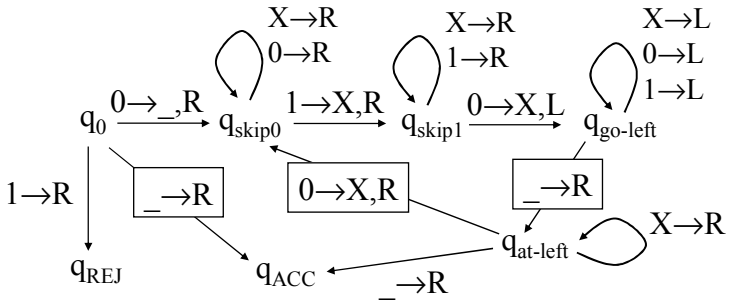
  1. If first symbol = blank, ACCEPT
  2. If first symbol = 1, REJECT
  3. If first symbol = 0, Write a blank to mark left end of tape
     a. If current symbol is 0 or X, skip until it is 1. REJECT if blank.
     b. Write X over 1. Skip 1's/X's until you see 0. REJECT if blank.
     c. Write X over 0. Move back to left end of tape.
  4. At left end: Skip X's until:
     a. You see 0: Write X over 0 and GOTO 3a
     b. You see 1: REJECT
     c. You see a blank space: ACCEPT

State Diagram



- Try running the decider on:
  - 010, 001100, … → ACCEPT
  - 0, 000, 0100, … → REJECT
  - What about 010010?

What's the problem?



- The decider accepts incorrect strings:
  - 010010, 010001100 → ACCEPT!!!
  - Accepts $(0^n1^n0^n)^*$

Need to fix it…How??

A Simple Fix (to the Decider)

- Scan initially to make sure string is of the form 0*1*0*
- On input w:
  1. If first symbol = blank, ACCEPT
  2. If first symbol = 1, REJECT
  3. If first symbol = 0: if w is not in 00*11*00*, REJECT; else,
     Write a blank to mark left end of tape
     a. If current symbol is 0 or X, skip until it is 1. REJECT if blank.
     b. Write X over 1. Skip 1's/X's until you see 0. REJECT if blank.
     c. Write X over 0. Move back to left end of tape.
  4. At left end: Skip X's until:
     a. You see 0: Write X over 0 and GOTO 3a
     b. You see 1: REJECT
     c. You see a blank space: ACCEPT

Add this

The Decider TM for L in all its glory