Agenda

• Last week
  - Turing machines

• This week
  - Sections 3.2 & 3.3
    • Variants of Turing machines and the definition of algorithm
Announcements

• Homework due next Tuesday
  - 3.3, 3.8 a (low-level description), 3.9, 3.12, 3.13

• Tutorial sessions are back
  - Office hours return to normal
    • Tuesday 3:00 - 4:00
    • Wednesday 3:00 - 4:00
Group project 1

• Design a Turing machine to accept any string in \( \{a,b\}^* \) after making a copy of it on the tape
  - The tape will start with \( w \)
  - After TM processes the string, the tape should read \( ww \)
TM design

• **Read first symbol**
  - If a, replace with x, move to end and write x
  - If b, replace with y, move to end and write y

• **Move left to first x or y**

• **Move left to next x or y**
  - If x, replace with a and move right
  - If y, replace with b and move right
**TM design (cont.)**

- **Read symbol at tape head**
  - If x, replace with a and accept
  - If y, replace with b and accept
  - If neither, loop to beginning
States

• **Start state** $q_1$

• **Move right**
  - Write a at end, write b at end, write x at end, write y at end

• **Move left to first x or y**

• **Move left to second x or y & replace with a or b**

• **Replace a or b with x or y**
Copy machine

q_{accept}

q_{reject}

October 12, 2004
Variants of Turing machines

- **Robustness of model**
  - Varying the model does not change the power
  - Example, making finite automata nondeterministic

- **Simple variant of TM model**
  - Add “stay put” direction

- **Other variants**
  - More tapes
  - Nondeterministic
Multitape Turing machines

• Same as standard Turing machine, but have several tapes
• TM definition changes only in definition of $\delta$

$$\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L,R\}^k$$
Equivalence of machines

Theorem: Every multitape Turing machine has an equivalent single tape Turing machine

Proof method: construction
Equivalent machines

M

0 1 ~ ~ ~ ~ ~ ~

a a a ~ ~ ~ ~ ~ ~

a b ~ ~ ~ ~ ~ ~

S

# 0 1 # a a a # a b # ~ ~
Simulating k-tape behavior

• Single tape start string is
  \[\#w\#_\#\ldots\#_\#\]

• Each move proceeds as follows:
  - Start at leftmost slot
  - Scan right to \((k+1)^{st}\) \# to find symbol at each virtual tape head
  - Make second pass making updates indicated by k-tape transition function
  - When a virtual head moves onto a \#, shift string to right
Corollary

Corollary: A language is Turing-recognizable if and only if some multitape Turing machine recognizes it.
Example

• Using 2-tape Turing machine, write a copy machine
• Copy tape 1 to tape 2
• Move tape 1 to beginning
• Copy tape 1 to tape 2
• Accept
Nondeterministic Turing machines

- Same as standard Turing machines, but may have one of several choices at any point

\[ \delta : Q \times \Gamma \rightarrow P(Q \times \Gamma \times \{L,R\}) \]
Equivalence of machines

Theorem: Every nondeterministic Turing machine has an equivalent deterministic Turing machine

Proof method: construction

Proof idea: Use a 3-tape Turing machine to deterministically simulate the nondeterministic TM. First tape keeps copy of input, second tape is computation tape, third tape keeps track of choices.