Agenda

• Yesterday
  - Variants of Turing machines
    • Allow “stay put” state
    • Multiple tapes
    • Nondeterministic

• Today
  - Prove equivalence of deterministic and nondeterministic Turing machines
  - Enumerators
Announcements

• **Quiz tomorrow**
  - High-level description of TM
  - Trace through a TM’s operation
    • Tape notation & configuration notation
  - High-level description of equivalences

• **Reminder: tutorial sessions are back**
  - Office hours return to normal
    • Tuesday 3:00 – 4:00
    • Wednesday 3:00 – 4:00
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.
Proof idea

Try new decision paths until the string is accepted
**Intuition**

- Consider the nondeterministic calculations as a tree
  - Each node represents a configuration
  - A node for configuration $C_1$ has one child for each configuration $C_2$ such that $C_1$ yields $C_2$
  - Root of tree is configuration $q_1w$
  - A configuration may appear more than once in the tree
Example

• Nondeterministic TM that accepts \{ww \mid w \in \{a,b\}^*\}
• Use 2 tapes
• Copy input to tape 2
• Position heads at beginning of tapes
• Move both heads right simultaneously
Nondeterministic solution

• Nondeterministically choose the midpoint
  - Mark this point on tape 2 and return tape 2’s head to beginning

• Compare strings
  - If tape head points to ~ on tape 1 and midpoint marker on tape 2 then accept
  - Otherwise, if all possible midpoints have been tried then reject
  - Otherwise, try a new midpoint
Nondeterministic TM

(qaccept)

(qreject)

(a,~) → {a,a},{R,R}
{b,~} → {b,b},{R,R}
(a,x) → {a,a},{R,R}
{b,x} → {b,b},{R,R}

(a,~) → (a,a),(R,R)
(b,~) → (b,b),(R,R)

(a,a) → (L,L)
(b,b) → (L,L)

(a,a) → (R,R)
(b,b) → (R,R)

(a,b), (b,a), (a,x), (b,x), (~,a), (~,b) →
(Res,Res)

(~,x) → (S,S)
Tree representation
Deterministic equivalent

- Assume midpoint is at beginning
  - If so accept
  - If not ...
- Assume midpoint is after first symbol
  - If so accept
  - If not ...
- Assume midpoint is after second symbol
  - If so accept
  - If not ...
- etc. ...
How should it search the tree?

• **Breadth first search**
  - Search all possibilities involving k steps before searching any possibilities involving (k+1) steps

• **What’s wrong with depth first search?**
  - If some sequence of choices results in no halting, we will never get to the accept state
When does it halt?

• When it reaches an accept state or
  - Return accept
Will it halt on strings in the language?

• Yes if the TM accepts the input string
  - Let b be the largest number of children of any node
    • Can we be sure b is finite?
  - Let k be the minimum number of steps it takes to get to the accept state
  - This method will take at most $b^k$ steps to get to the accept state
What about strings not in the language?

• Won’t halt
  – That’s okay
Equivalence of approaches

Corollary: A language is Turing-recognizable if and only if some nondeterministic Turing machine recognizes it.
Equivalence of approaches

Corollary: A language is Turing-decidable if and only if some nondeterministic Turing machine decides it.

Proof: Homework

- Modify proof in the book