Lecture 7

Thursday, April 06, 2023 11:11 AM

- Turing Machines
  - General model of computation
  - Turing machines as lang acceptors
  - Turing machines that compute
- General Model of Computation
  - Both finite automata and pda are models of computation
    - Each gets input string and executes an algorithm to get an answer, following a set of rules specific to the machine type
  - Easy to find examples of langs that cannot be accepted bc of the machine's
  -
  -
  -
  - PDA-like machine w/ 2 stacks can accept AnBnCn
  - An FA w/ a queue instead of a stack can accept L
  - In both cases, might seem like machine is specifically deved to handle 1 lang
  -
  - Abstract model called Turing machine
    - Not got by adding data structures onto a finite automaton
    - It predates the FA and PDA models
  - Turing machine not just next step beyond PDA
    - Thesis/theorem, general model of computation
  - Turing objective was to demonstrate the inherent limitations of algorithmic methods
    - That's why wanted his device to be able to execute any algorithm that a human computer could
  - Formulate computational model, think of human w/ pencil
  - Steps include
    - Examine individual symbols on paper
    - Erase symbol/replace it by another
    - Transfer attention from one symbol to nearby one
  - Simplicity, specified a linear tape which has left end and potentially infinite to right
    -
    -
  - In our version, turing machine, single move is determined by the current state, current tape symbol has 3 pts:
    - Changing from the current state to another
    - Replacing the symbol in then square by another
    - Leaving the tape
  - Tape gives mem needed and serves as output device
  - Diff btwn Turing and FA/PDA is that Turing machine not restricted to single pass thru the input, so it can backtrack, pointer can go left/right
  - So finite automata cannot go back and re-read previous symbol
  - Focus on 2 main objectives of turing machine
    - Accepting a lang
    - Computing a fn
  - Turing machine will have 2 halt states, one acceptance, other rejection
    - More than 2 necessary
    - Unlike FA, complete input string is on the tape initially, and a separate answer for each prefix isnt req
  - Unlike FA's and PDA's (or at least pDA's w/out lambda transitions), Turing machines
  - Def
  - Delta function
    - H is halt
    - D is direction
    - In diagram, one state (p) is where we at, other state (q) is where we go
    - Input start at square 1 (literally), square 0 is blank, and ones following input string is blank
    - Nonblank squares on tape must be finite
    - Describe the current config of a Turing machine by single string xqy, where q is current state, x is string of symbols to left of current square, and y is
- Turing Machine Model
  - Sometimes, the blank symbol – is denoted as delta
  -
- Turing Machines as Lang Acceptors
  - If lang not reg, the Turing Machine (TM) couldn't move its tape head to the right on every move
    -
    -
    -
    -
    -
- Turing Machines that Compute Partial Fns
  - Make output string for every legal input string said
  - Def
  - For our purposes just consider partial fn natural numbs
  - Use unary notation for numbs

    - Official def like the def, except input alphabet is {1} and initial config look long
- Ex of TM that computes a Partial fn
    - String over {a, b} it switches a's and b's w/ opp
- Combo Turing Machine
    -
    -