Computability and Computational Complexity

Week 1

Course Notes

Definitions

- Course is about limits of computational logic. What type of problems can't be reasonably computed by algorithms, what can and how efficiently.
  - ***Computability***: what questions can logic solve.
  - ***Complexity***: what questions can logic solve efficiently.
- ***Decision problem***: problem whose answer is Yes or No.
- ***Instance***: inputs to a problem (analogous to inputs to a program).
  - ***Yes-instances***: inputs to a decision problem that should give a Yes answer.
  - ***No-Instances***: inputs to a decision problem that should give a No answer.
- ***Exercise***: write the sets that encode the yes-instances of the problem.
  - Q: Given integers x, y, z, is $z = x^2 + y^3$?
    - A: $S_1 = \{(x, y, z) \in \mathbb{Z}^3 | z = x^2 + y^3\}$.
  - Q: Given integers x, z, is there integer y such that $z = x^2 + y^3$?
    - A: $S_2 = \{(x, z) \in \mathbb{Z}^2 | \exists y \in \mathbb{Z}, z = x^2 + y^3\}$
- ***Encoded format***: machine-readable format of an object. E.g., given graph G an encoded graph $\langle G \rangle$ is e.g., an adjacency matrix.
- ***Alphabet Σ***: finite set of characters e.g., $\Sigma = \{0,1\}$. Σ^* is the set of all possible strings composed of elements from the alphabet, including empty string ϵ.
- ***Languages over Σ^****: sets of finite strings using characters from Σ.
- Link between language and decision problems.
  - Set of yes-instances is the language describing the problem.
  - Decision problem for a language: is this input part of the language?
- ***Algorithm***: logical sequence of steps which will, for any problem instance, terminate in a finite amount of time and give the solution for that instance.
  - In this model of computation an algorithm must halt so we can be certain of the computation result.

Algorithms as Mathematical Objects

- DFSA: memoryless automata that can determine membership in a language. But can't count.
- PDA: machine that recognizes languages requiring counting by using stack memory. But because uses stack, cannot interleave patterns, e.g., cannot recognize $L = \{w\#w \mid w \in \{0,1\}^*\}$.

Fig 1. DFSA. $L = \{x \in \{0,1\}^* \mid x \text{ begins and ends in same char}\}$.

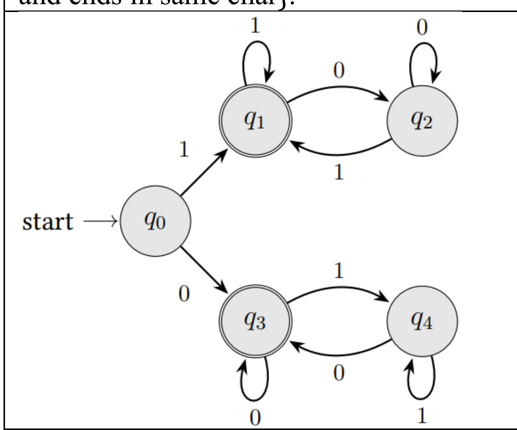
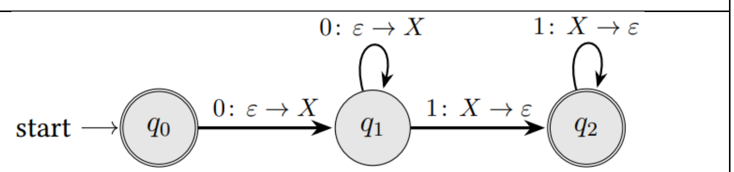


Fig 2. PDA. $L = \{0^n 1^n \mid n \in \mathbb{N}\}$.



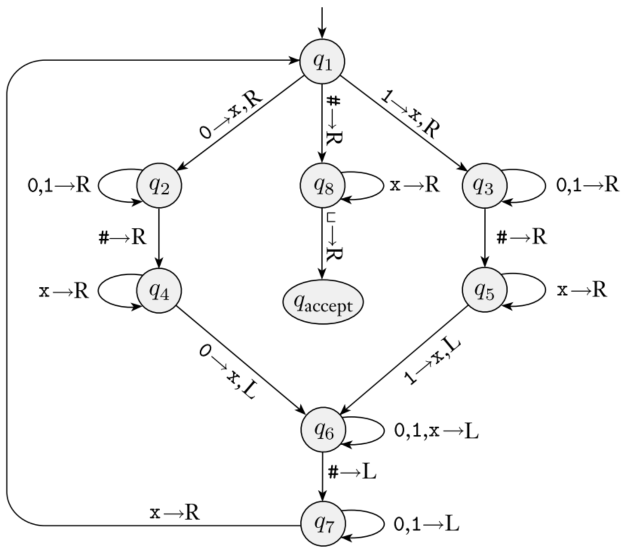


Fig 3. TM that recognizes $L = \{w\#w \mid w \in \{0,1\}^*$.

- ***Turing Machine (TM)***: A TM M is a machine described by a tuple $\langle M \rangle = (Q, \Sigma, \Gamma, \delta, s, q_A, q_R)$.
  - Q: set of states, nodes in a diagram.
  - Σ: input alphabet, set of available input characters, e.g., $\Sigma = \{0,1,\#\}$.
    - $\sqcup \notin \Sigma$.
  - Γ: tape alphabet, any characters used in tape which includes input characters and space, e.g., $\Gamma = \{0,1,\#, x, \sqcup\}$.
    - $\Sigma \subset \Gamma$, $\sqcup \in \Gamma$
  - δ: transition function, arrows in diagram. $\delta: Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$.
  - s: start state
  - q_A: accept state, halt and accept.
  - q_R: reject state, halt and reject.
    - Implicitly halt if no arrows out of a state.
- TM Notes
  - Alan Turing's formalism for describing algorithms.
  - TM operates with a tape instead of a stack.
  - TM allows arbitrarily large computing time by allowing the head of the tape to move left or right upon reading a character.
  - TM allows unrestricted memory by allowing unlimited blank spaces $\sqcup$ to the right of the string on the tape.
  - Remain in place when try to move left at left endpoint of tape.
  - ***$L(M)$***: is the language of strings accepted by a TM M.
  - ***Configuration (of a TM)***: a setting of the current state, current tape content, and current head location, e.g., $q_1 010\#011$ is the initial configuration in Fig 3.
- Advantages of TMs:
  - We can program them like regular computers.
  - Configurations are well-behaved so we can do math on them. This allows us to relate TMs (and computability results on TMs) to non-TM objects.
- ***Exercise***: Build a TM to recognize $L = \{1^{2^n} \mid n \in \mathbb{N}\}$.

Conventions

- $\mathbb{N}$ includes 0. $\mathbb{Z}_+$ is pos integers. $\mathbb{Z}_-$ is negative integers.
- If $a, b, c \in \mathbb{Z}$, then $a \equiv b \ (mod\ c)$ iff c divides $b - a$.
- $A \setminus B$ is set difference between A and B, i.e., the set $\{x \in A | x \notin B\}$.
- $A \oplus B$ is symmetric difference between A and B, i.e., $A \oplus B = (A \setminus B) \cup (B \setminus A)$.