Summary of Finite Automata

Using Pumping Lemma

- States are the memory of FSMs. There are only finite number of states. Thus only a finite amount of information can be stored. (At most n, where n is the $\#$ of states.)

  If L contains infinitely long strings and it seems necessary to store an amount of information proportional to the length of the string, likely L is not regular.

- Converse of Pumping Lemma doesn't hold

  There are non-regular languages that satisfy P.L.

  Thus P.L. is only used to show languages not regular.

  You cannot say "L satisfies P.L., thus L is regular."

Summary

- Showing regular
  - construct DFA, NFA
  - construct regular expression
  - show L is the union, concatenation, intersection, $\cdots$ (regular operations) of regular languages.
- Showing non-regular
  - pumping lemma
  - assume regular, apply closure properties of regular languages and obtain a known non-regular language.

    Examples:

    Show $L = \{(0 \cup 1)^n b (0 \cup 1)^n | n \geq 1\}$ non-regular.

    Show $L = \{0^m 1^m 0^n 1^n | n \geq 0, m \geq 0\}$ non-regular.

Context Free Languages (CFL)

- Proper superset of regular languages.
  - human languages
  - programming languages
  - recursive structure
- Context-free grammar (CFG)
  - describing CFL
  - automatic generation of parser from CFG
    compiler-compiler
- Pushdown automata
  - recognizing CFL

Context Free Grammar (CFG)

- $G = (V, \Sigma, R, S)$
  - V: finite set of variables (nonterminals),
  - Σ: finite set of terminals. $V \cap \Sigma = \emptyset$,
  - $S \in V$: a special start symbol,
  - R: finite set of productions of form $A \to \alpha$, where $A \in V, \alpha \in (V \cup \Sigma)^*$.
- Example