CSCI 383: Languages and Automata

Exam 2 Review

Stephen Checkoway

Exam format

- 3 multi-part questions
  - True+explanation or false+counterexample
  - Two constructions
  - One of two possible proofs
- 50 minutes
- Notes/printed material is fine to use, no phones or computers

Exam topics

- Everything through NPDAs, focus on material since last exam (so context-free languages)
- CFGs, both the mathematical definition as a 4-tuple $G = (N, \Sigma, P, S)$ and as lists of rules
- NPDAs, both the mathematical definition as a $M = (Q, \Sigma, \Gamma, \delta, s, \bot, F)$ with acceptance by final state and acceptance by empty stack and as a diagram
- The class of CFLs is closed under union, concatenation, and Kleene star. It is also closed under intersection with a **regular** language
- Chomsky Normal Form
  - The form of the rules
  - How to convert to CNF
- Pumping lemma for context-free languages

Topic *not* on the exam

- I won’t ask you to convert a NPDA to a CFG nor vice versa

CFG

G = (N, Σ, P, S)

- N is a finite set of nonterminals, Σ is a finite set of terminals
- P is a set of production rules of the form A → u where u is a finite sequence of terminals and nonterminals
- S is the start nonterminal
- Derivations of strings in the language proceed by starting with the start nonterminal and then iteratively replacing nonterminals with the right-hand side of matching rules until only terminal symbols are left

Derivations

- S → aaSb | T
  T → bTa | ε

- The string aaaabbaabb is in the language; a derivation
  S ⇒ aaSb ⇒ aaaaSbb ⇒ aaaaTbb ⇒ aaaabTabb ⇒ aaaabbTaabb ⇒ aaaabbaabb

- Any sequence of terminals and nonterminals derivable from the start nonterminal is called a **sentinel form**

- A **left-most derivation** is one in which the left-most nonterminal is replaced in each step of the derivation

The language of a grammar

G = (N, Σ, P, S)

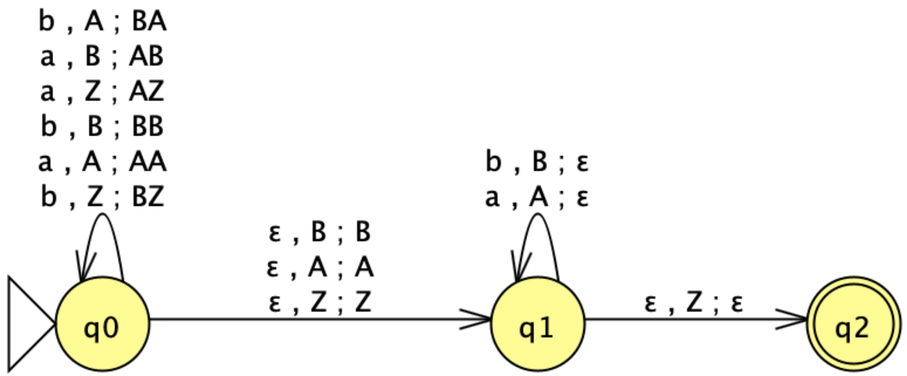
- $L(G) = \{ x \mid S \Rightarrow^* x \text{ and } x \in \Sigma^* \}$

Nondeterministic Pushdown Automata (NPDA)

$M = (Q, \Sigma, \Gamma, \delta, s, \bot, F)$

- Q is the finite set of states
- Σ is the finite input alphabet
- Γ is the finite stack alphabet
- δ is the finite transition relation on $(Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma) \times (Q \times \Gamma^*)$
  - That is, for every triple (state, input_symbol_or_epsilon, stack_symbol), there are 0 or more pairs (state, string_of_stack_symbols) that are the transitions
  - ((q, c, A), (r, BCD)) is the transition indicating that starting in state q, if the next input symbol is c and the top of the stack is A, then transition to state r, pop A off the stack and push BCD on the stack in its place

Drawing transition diagrams



- Transitions need to be c, A → BCD… where c is an input symbol or the empty string, and A, B, C, D, … are stack symbols
- (JFLAP uses Z for the initial stack symbol and ; rather than → in its transitions)

Configurations of an NPDA

M = (Q, Σ, Γ, δ, s, ⊥, F)

- A configuration is a triple $Q \times \Sigma^* \times \Gamma^*$
  - It consists of the current state, remaining input symbols to read, and the contents of the stack
- There's a "next-configuration" relation
  $(q, cx, Au) \rightarrow (r, x, vu)$
  if δ contains
  $((q, c, A), (r, v))$
  meaning there's a transition from state q to state r on input c in $\Sigma \cup \{\varepsilon\}$ which pops A off the stack and pushes v in Γ^* onto the stack

Acceptance of an NPDA

M = (Q, Σ, Γ, δ, s, ⊥, F)

- An NPDA accepts a string x by final state if
  $(s, x, \perp) \rightarrow^* (f, \varepsilon, u)$ for some f in F
- An NPDA accepts a string x by empty stack if
  $(s, x, \perp) \rightarrow^* (q, \varepsilon, \varepsilon)$
- Any NPDA can be converted to one that accepts the same language by either final state or empty stack (the idea is to add a single new accept state that empties its stack and never transitions away; details in Kozen)
- L(M) = {x | M accepts x}
  [Just plug in the definition for acceptance that applies to the NPDA M]

Closure properties of context-free languages

- In groups, prove that the set of context-free languages is closed under the regular operations: union, concatenation, and Kleene star
- Specifically, given CFGs $G_1 = (N_1, \Sigma, P_1, S_1)$ and $G_2 = (N_2, \Sigma, P_2, S_2)$, construct grammars for (you can assume that N_1 and N_2 are disjoint)
  - $L(G_1) \cup L(G_2)$
  - $L(G_1)L(G_2)$, and
  - $L(G_1)^*$

Pumping lemma (contrapositive, “Demon game” form)

1. Demon picks k (the pumping length)
2. You pick a string z of length at least k
3. Demon picks a partition of z into uvwxy = z such that $|vwx| \leq k$ and $|vx| > 0$
4. Based on the Demon’s pick in step 3, you pick some $i \geq 0$ such that uv^iwx^iy is not an element of the language

In groups, prove that the language $A = \{a^ib^jc^k \mid i < j < k\}$ is not context-free

Intersection of CFLs

- Give a CFG for the language B = $\{a^i b^j c^k \mid i < j\}$
- Give a CFG for the language C = $\{a^i b^j c^k \mid j < k\}$
- Prove that the class of context-free languages is **not** closed under intersection (i.e., find two CFLs whose intersection is not CF)