- Start as early as possible, and contact the instructor if you get stuck.
- See the course outline for details about the course's marking policy and rules on collaboration.
- Submit your completed solutions to **Crowdmark**.

1. A pushdown automaton

[6] Give a PDA that accepts the language of even-length words that are not palindromes over the alphabet $\Sigma = \{a, b\}$. Your machine may accept by final state or by empty stack, whichever is more convenient. State explicitly whether your machine accepts by final state or by empty stack. Explain why your PDA is correct.

2. Building a context-free grammar from a PDA
   Define a pushdown automaton, $P = (Q, \Sigma, \Gamma, \delta, q, Z_0)$ (which accepts by empty stack), with
   - $Q = \{q, p\}$
   - $\Sigma = \{0, 1\}$
   - $\Gamma = \{Z_0, X\}$
   - q = start state for machine
   - Z_0 = stack start letter (bottom of stack character)

   and transition function
   1. $\delta(q, 0, Z_0) = \{(q, XZ_0)\}$
   2. $\delta(q, 0, X) = \{(q, XX)\}$
   3. $\delta(q, 1, Z_0) = \{(p, Z_0)\}$
   4. $\delta(q, 1, X) = \{(p, X)\}$
   5. $\delta(p, 0, X) = \{(p, \varepsilon)\}$
   6. $\delta(p, \varepsilon, Z_0) = \{(p, \varepsilon)\}$

[2] (a) Draw a diagram for P.

[8] (b) Use the technique described on slides 51-54 of Module 6 to construct a context-free grammar, G, such that $L(G) = N(P)$. In your final grammar, replace the non-terminals from the construction with single capital letters $A, B, C, \ldots$. Simplify your grammar as much as possible after you have completed the construction. You do **not** have to prove that $L(G) = N(P)$. But you should convince yourself that the equality holds once you have completed the construction and simplification of G.

[4] (c) The construction in part 2b can introduce some productions which will never resolve all of their variables. Simplify the grammar you found in part 2b by eliminating all such productions. If this leaves some variables without any productions, then eliminate those variables as well.

[5] (d) Give a “nice” description of $L(G)$, and prove that your description is correct. (You do not need to include this parenthesized part in your solution, but you should also confirm that your description of $L(G)$ makes sense with respect to the definition of the original PDA, P.)

3. The analog of Kleene's Theorem for CFLs and PDAs

Let Σ be an alphabet.

[6] (a) Let D be an arbitrary DFA, with alphabet Σ. Describe a construction for an ε-NFA, E, which, given any input word $w \in \Sigma^*$,

i. accepts no non-empty words, and

ii. accepts the empty word, ε, if any only if D accepts w.

Briefly explain why your construction of E is correct.

[8] (b) Let L_1, L_2 be languages over Σ. Define

$$L_1/L_2 = \{w \mid wx \in L_1, \text{ for some } x \in L_2\}.$$

Suppose that L_1 is context-free, and L_2 is regular. Prove that L_1/L_2 is context-free. (Hint: use part 3a.)

4. Deterministic PDAs

[4] Suppose that D is a DPDA whose set of final states is F, and that accepts $L(D)$ by final state. Let D' be the same DPDA, but with final states $Q \setminus F$ (that is, the accept states of D' are those states that are not accept states of D. Is it always the case that $L(D')$ equals the complement of $L(D)$? Prove your answer.

5. Recall from Slides 13–17 of Module 5, a grammar for the language $\{0^i 1^j \mid 0 \leq i \leq j\}$ is:

$$G : S \to \varepsilon \mid 0S1 \mid S1.$$

[8] (a) Use the technique on slides 6-17 of Module 7 to produce a grammar, G', in Chomsky Normal Form, such that $L(G) = L(G') \cup \{\varepsilon\}$. You do **not** need to prove this equality of languages: following the algorithm correctly will guarantee this fact.

[4] (b) Give an explicit derivation, in G', for the word: 00111.