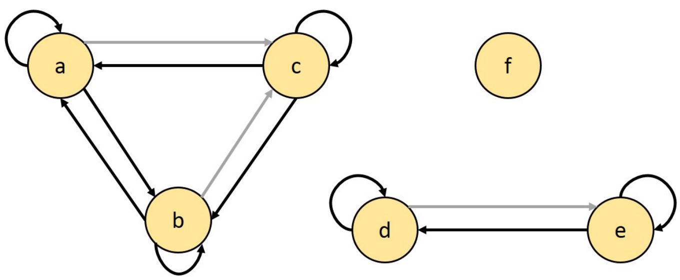
Suggested Solutions to Midterm Problems

1. Let $A = \{a, b, c, d, e, f\}$ and $R = \{(a, c), (b, c), (d, e)\}$ (which is a binary relation on A).

   (a) Give a symmetric and transitive but not reflexive binary relation on A that includes R. Please present the relation using a directed graph.

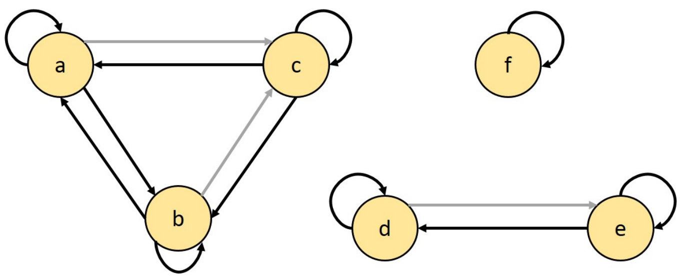
   Solution. (Jui-Shun Lai)



□

   (b) Find the smallest equivalence relation on A that includes R. Please present the relation using a directed graph.

   Solution. (Jui-Shun Lai)

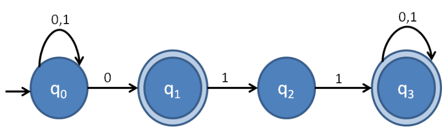


□

2. Let $L = \{w \in \{0, 1\}^* \mid w$ contains 011 as a substring or ends with a $0\}$.

   (a) Draw the state diagram of an NFA, with as few states as possible, that recognizes L. The fewer states your NFA has, the more points you will be credited for this problem.

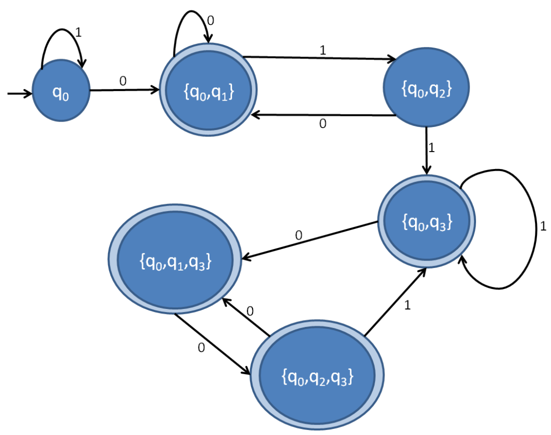
   Solution. (Wei-Hsien Chang)



□

(b) Convert the preceding NFA systematically into an equivalent DFA (using the procedure discussed in class). Do not attempt to optimize the number of states, though you may omit the unreachable states.

Solution. (Wei-Hsien Chang)

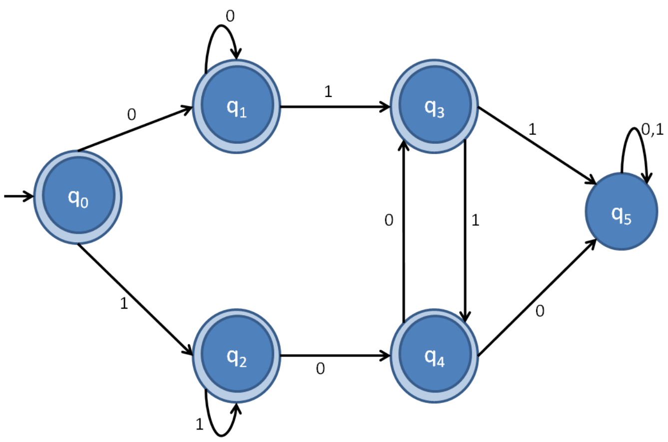


□

3. Let $L = \{w \in \{0,1\}^* \mid w$ does not contain 011 or 100 as a substring$\}$.

(a) Draw the state diagram of a DFA, with as few states as possible, that recognizes L. The fewer states your DFA has, the more points you will be credited for this problem.

Solution. (Wei-Hsien Chang)



□

(b) Translate the DFA in (a) systematically to an equivalent context-free grammar (using the procedure discussed in class).

Solution. (Wei-Hsien Chang)
$R_0 \to \varepsilon \mid 0R_1 \mid 1R_2$
$R_1 \to \varepsilon \mid 0R_1 \mid 1R_3$
$R_2 \to \varepsilon \mid 0R_4 \mid 1R_2$

$R_3 \rightarrow \varepsilon \mid 1R_4$
$R_4 \rightarrow \varepsilon \mid 0R_3$
(Note: useless production rules have been removed.) □

4. Let $L = \{1^p \mid p \text{ is a prime number less than } 2^{2^{10}}\}$. Is L a regular language? Why or why not?

Solution. Any finite set of strings is a regular language, as we can easily construct an NFA for each of the strings and take the union of all such NFAs to obtain the final NFA that recognizes the language. L is finite and hence regular. □

5. Let $C_n = \{x \mid x \text{ is a binary number that is a multiple of } n\}$. Show that for each $n \geq 1$, the language C_n is regular.

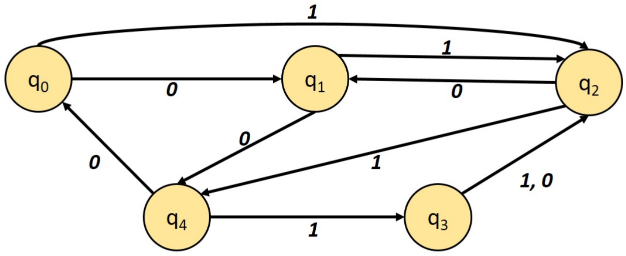
Solution. (Wei-Hsien Chang)
To show that C_n is regular, we construct a DFA $M = (Q, \Sigma, \delta, q_0, \{q_0\})$ which can recognize C_n.

- $Q = \{q_i \mid 0 \leq i < n\}$
- $\Sigma = \{0, 1\}$
- For $0 \leq i < n$,
  $\delta(q_i, 0) = q_{(2*i) \bmod n}$
  $\delta(q_i, 1) = q_{(2*i+1) \bmod n}$

□

6. A *synchronizing sequence* for a DFA $M = (Q, \Sigma, \delta, q_0, F)$ and some "home" state $h \in Q$ is a string $s \in \Sigma^*$ such that, for every $q \in Q$, $\delta(q, s) = h$. A DFA is said to be *synchronizable* if it has a synchronizing sequence for some state. Try to find a 5-state synchronizable DFA over the alphabet $\{0, 1\}$ with a synchronizing sequence as long as possible. What is the longest synchronizing sequence for the DFA and which state is the corresponding home state? The longer the synchronizing sequence is, the more points you will be credited for this problem. (Note: the synchronizing sequence we seek should be minimal in the sense that none of its proper substrings is also a synchronizing sequence.)

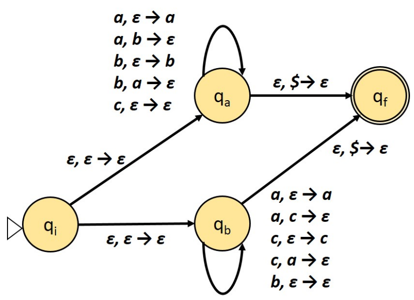
Solution. (Jui-Shun Lai)
Below is a 5-state synchronizable DFA with a minimal synchronizing sequence of length 10, namely 0100110110. (Note: the initial state and final states are not marked, as they are irrelevant.) Other 5-state synchronizable DFAs with a longer minimal synchronizing sequence might exist.



□

7. Draw the state diagram of a pushdown automaton (PDA) that recognizes the following language: $\{w \in \{a, b, c\}^* \mid$ the number of a's in w equals that of b's or c's$\}$. Please make the PDA as simple as possible and explain the intuition behind the PDA.

Solution. (Jui-Shun Lai)
In the initial state q_i, the PDA first pushes symbol \$ on the stack, and guesses the number of a's in w equals that of b's or c's. If it is the case of b's, it goes to state q_b; otherwise, it goes to state q_c. In state q_b, the PDA ignores symbol c, and tries to match the occurrences of a's with those of b's, by pushing symbol a or b to the stack when reading one of the them, or popping the top of stack if it is different from the input symbol. The PDA accepts the input when the stack is empty, so the w accepted through state q_b will have the same number of a's and b's. State q_c is analogous to state q_b.



□

8. Let $A/B = \{w \mid wx \in A \text{ for some } x \in B\}$. Show that, if A is context free and B is regular, then A/B is context free.

Solution. Suppose M_A is a PDA recognizing A and M_B a DFA recognizing B. To show that A/B is context free, we construct a PDA $M_{A/B}$ that recognizes A/B. $M_{A/B}$ proceeds in two stages. In the first stage, $M_{A/B}$ simulates M_A on the input w, and additionally in each step nondeterministically guessing that the end of w has been reached, branches to the second stage. When in the second stage, $M_{A/B}$ simulates both M_A and M_B on an imaginary input x, which is "self-supplied", rather than coming from the actual input. $M_{A/B}$ is defined such that every possible x is attempted.

Let $M_A = (Q_A, \Sigma, \Gamma_A, \delta_A, q_A^0, F_A)$ and $M_B = (Q_B, \Sigma, \delta_B, q_B^0, F_B)$. The DFA $M_{A/B} = (Q_{A/B}, \Sigma, \Gamma_{A/B}, \delta_{A/B}, q_{A/B}^0, F_{A/B})$ is defined as follows:

- $Q_{A/B} = Q_A \cup Q_A \times Q_B$.
- $\Gamma_{A/B} = \Gamma_A$.
- $\delta_{A/B}$ is defined according to the two stages:
  (a) Simulation of M_A on actual input and nondeterministic branching to the second stage. For $q_A \in Q_A$, $a \in \Sigma_\varepsilon$, and $b \in \Gamma_\varepsilon$,
  $$\delta_{A/B}(q_A, a, b) = \begin{cases} \delta_A(q_A, a, b) \cup \{((q_A, q_B^0), b)\} & \text{if } a = \varepsilon \\ \delta_A(q_A, a, b) \cup \{((q'_A, q_B^0), c) \mid (q'_A, c) \in \delta_A(q_A, a, b)\} & \text{if } a \neq \varepsilon \end{cases}$$
  (b) Simulation of M_A and M_B on imaginary input. For $(q_A, q_B) \in Q_A \times Q_B$, $a \in \Sigma_\varepsilon$, and $b \in \Gamma_\varepsilon$,
  $$\delta_{A/B}((q_A, q_B), a, b) = \begin{cases} \{((q'_A, q'_B), c) \mid (q'_A, c) \in \delta_A(q_A, a', b) & \text{if } a = \varepsilon \\ \quad \text{for some } a' \in \Sigma \text{ s.t. } q'_B = \delta_B(q_B, a')\} & \\ \emptyset & \text{if } a \neq \varepsilon \end{cases}$$

- $q^0_{A/B} = q^0_A$.
  So, $\delta_{A/B}$ includes an ε-transition going from $q^0_{A/B}$ to (q^0_A, q^0_B) with no update on the stack.
- $F_{A/B} = \{(q_A, q_B) \mid q_A \in F_A \text{ and } q_B \in F_B\}$.

□

9. Show that, if G is a CFG in Chomsky normal form, then any string $w \in L(G)$ of length $n \geq 1$, exactly $2n - 1$ steps are required for any derivation of w.

   Solution. Given a CFG G in Chomsky normal form and an arbitrary string $w \in L(G)$ of length $n \geq 1$, we observe that any parse tree T for w has the following two properties:

   - T has exactly n leaves, since no leaves may be the empty string (given that $w \neq \varepsilon$).
   - Each of the n leaves is the only child of some internal node.
   - All internal nodes, except those that are parent of a leaf, have exactly two other internal nodes as children.

   For any CFG, the number of internal nodes of a parse tree equals the number of steps in the corresponding derivation. In particular, the number of T's internal nodes equals the number of steps in the corresponding derivation of w. To count T's internal nodes, we remove all the n "single-child" leaves of T to obtain a tree T'. T' will be a binary tree with n leaves and all its internal nodes will have two children. Any tree like T' can be shown to have exactly $n - 1$ internal nodes and hence $2n - 1$ nodes in total. This implies that T has $2n - 1$ internal nodes. As T represents any parse tree for w, it follows that any derivation of w takes $2n - 1$ steps. □

10. Prove that the language over $\{a, b, c\}$ with equal numbers of a's, b's, and c's is not context free.

    Solution. The proof for the non-context-freeness of $\{a^n b^n c^n \mid n \geq 0\}$ (discussed in class) using the pumping lemma may be reused here.

    Alternatively, we can also prove the result by utilizing closure properties of regular and context-free languages. Let A denote the language as defined in the problem statement. Let $B = \{a^i b^j c^k \mid i, j, k \geq 0\}$, which is apparently regular. As the intersection of a context-free language and a regular language is context free, if A were context free, then $A \cap B$ would also be context free. However, $A \cap B$ equals $\{a^n b^n c^n \mid n \geq 0\}$, which is not context free (asserted in the appendix), and it follows that A is not context free. □

Appendix

- Properties of a binary relation R on A:
  - R is *reflexive* if for every $x \in A$, xRx.
  - R is *symmetric* if for every $x, y \in A$, xRy if and only if yRx.
  - R is *transitive* if for every $x, y, z \in A$, xRy and yRz implies xRz.
- A context-free grammar is in **Chomsky normal form** if every rule is of the form

$$\begin{array}{lll} A & \rightarrow & BC \quad \text{or} \\ A & \rightarrow & a \end{array}$$

where a is any terminal and A, B, and C are any variables—except that B and C may not be the start variable. In addition,

$$S \to \varepsilon$$

is permitted if S is the start variable.

- (Pumping Lemma for Context-Free Languages) If A is a context-free language, then there is a number p such that, if s is a string in A and $|s| \geq p$, then s may be divided into five pieces, $s = uvxyz$, satisfying the conditions: (1) for each $i \geq 0$, $uv^ixy^iz \in A$, (2) $|vy| > 0$, and (3) $|vxy| \leq p$.
- The language $\{a^nb^nc^n \mid n \geq 0\}$ is not context free.