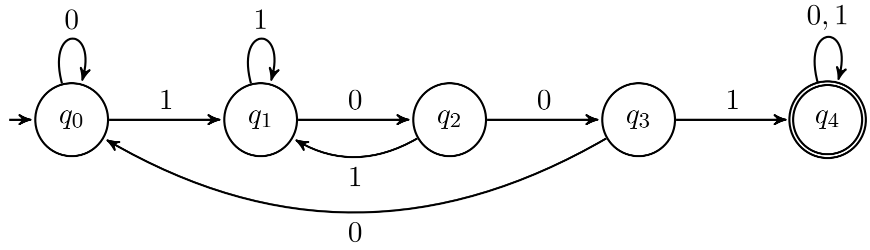
Suggested Solutions to Midterm Problems

1. Give the state diagrams of DFAs recognizing the following languages. In all parts, the alphabet is $\{\texttt{0}, \texttt{1}\}$.

   (a) $\{w \mid w$ contains the substring $\texttt{1001}$, i.e., $w = x\texttt{1001}y$ for some x and $y\}$.

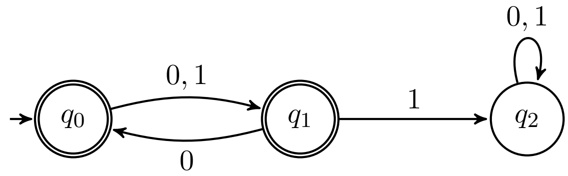
   Solution.



$\square$

   (b) $\{w \mid$ every even position of w is a $\texttt{0}\}$ (Note: see w as $w_1 w_2 \cdots w_n$, where $w_i \in \{\texttt{0}, \texttt{1}\}$).

   Solution.

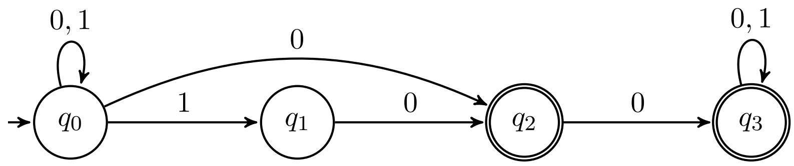


$\square$

2. Let $L = \{w \in \{0, 1\}^* \mid w$ contains 100 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.

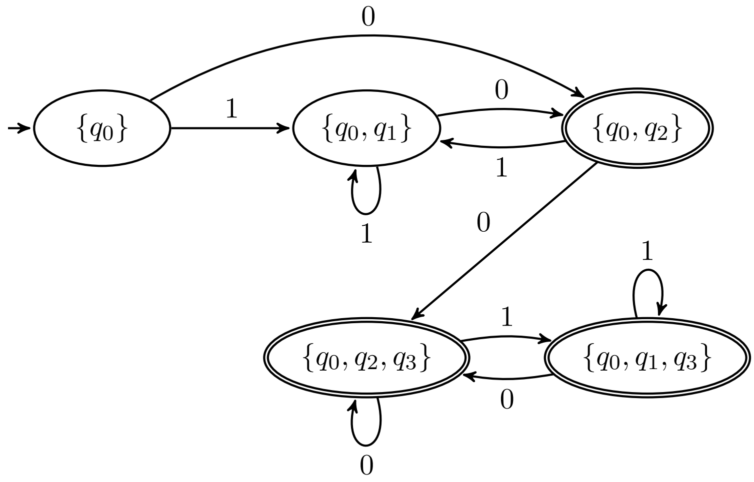
   Solution.



$\square$

   (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.



$\square$

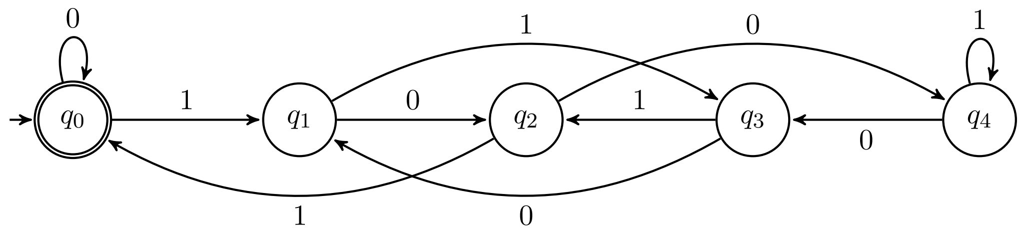
3. Let $L = \{1^p \mid p \text{ is a prime number less than } 2^{2^{10}}\}$. Is L a regular language? Why or why not?

   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. $\square$

4. Give the state diagram of a DFA that recognizes the following language:

$$C_5 = \{x \mid x \text{ is a binary number that is a multiple of } 5\}.$$

   Solution.



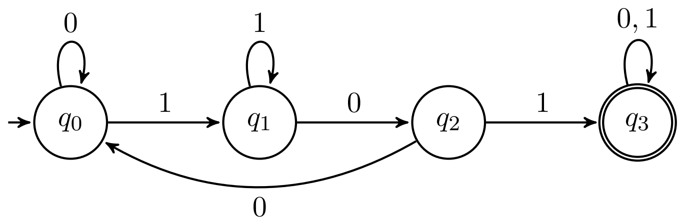
   Note: the automaton accepts the empty string, treating it as 0. $\square$

5. Given a language $L \subseteq \Sigma^*$, an equivalence relation R_L over Σ^* is defined follows:

$$xR_Ly \ \text{ iff } \ \forall z \in \Sigma^*(xz \in L \leftrightarrow yz \in L)$$

   Suppose $L = \{w \mid w \text{ contains the substring } \texttt{101}\text{, i.e., } w = x\texttt{101}y \text{ for some } x \text{ and } y\}$. What are the equivalence classes determined by R_L? Please give an intuitive verbal description for each of the equivalence classes.

   Solution. Applying the Myhill-Nerode theorem, we may discover the equivalence classes by examining a minimal DFA that recognizes L as below.



So, there are four equivalence classes corresponding to the four states:

(a) The subset of $\{0, 1\}^*$ containing ε, 0, and all strings ending with 00 but without 101 as a substring.

(b) The subset containing all strings ending with 1 but without 101 as a substring.

(c) The subset containing all strings ending with 10 but without 101 as a substring.

(d) The subset containing all strings with 101 as a substring.

$\square$

6. Consider the following CFG discussed in class, where for convenience the variables have been renamed with single letters.

$$\begin{array}{rcl} E & \rightarrow & E + T \mid T \\ T & \rightarrow & T \times F \mid F \\ F & \rightarrow & (E) \mid a \end{array}$$

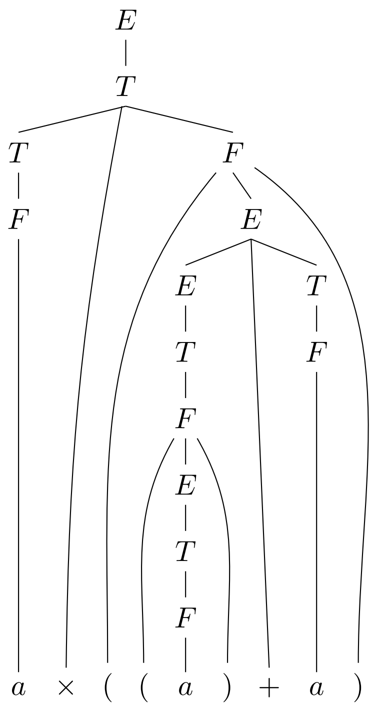
Give the (leftmost) derivation and parse tree for the string $a \times ((a) + a)$.

Solution.

The leftmost derivation

$$\begin{array}{rcl} E & \Rightarrow & T \\ & \Rightarrow & T \times F \\ & \Rightarrow & F \times F \\ & \Rightarrow & a \times F \\ & \Rightarrow & a \times (E) \\ & \Rightarrow & a \times (E + T) \\ & \Rightarrow & a \times (T + T) \\ & \Rightarrow & a \times (F + T) \\ & \Rightarrow & a \times ((E) + T) \\ & \Rightarrow & a \times ((T) + T) \\ & \Rightarrow & a \times ((F) + T) \\ & \Rightarrow & a \times ((a) + T) \\ & \Rightarrow & a \times ((a) + F) \\ & \Rightarrow & a \times ((a) + a) \end{array}$$

The parse tree



□

7. Prove that, if C is a context-free language and R a regular language, then $C \cap R$ is context-free. (Hint: combine the finite control part of a PDA and that of an NFA.)

   Solution. Suppose P is a PDA that recognizes C and A an NFA that recognizes R. A string w of $C \cap R$ will be accepted if we run P "in parallel" with A. The "parallel execution" of P and A can be simulated by another PDA whose transition function combines those of P and A.

   Formally, let

$$P = (Q_P, \sum, \Gamma, \delta_P, q_P^0, F_P)$$

   be a PDA that recognizes C and

$$A = (Q_A, \sum, \delta_A, q_A^0, F_A)$$

   be an NFA that recognizes R. Construct a PDA

$$P' = (Q_P \times Q_A, \sum, \Gamma, \delta, (q_P^0, q_A^0), F_P \times F_A)$$

   where $\delta((q_P, q_A), a, x)$, for every $(q_P, q_A) \in Q_P \times Q_A$, $a \in \Sigma$, and $x \in \Gamma$, is defined to be the set of all pairs $((q'_P, q'_A), y)$ such that:

   (a) $(q'_P, y) \in \delta_P(q_P, a, x)$ and
   (b) $q'_A \in \delta_A(q_A, a)$.

□

8. For two given languages A and B, define $A \diamond B = \{xy \mid x \in A \text{ and } y \in B \text{ and } |x| = |y|\}$. Prove that, if A and B are regular, then $A \diamond B$ is context-free. (Hint: construct a PDA where the stack is used to ensure that x and y are of equal length.)

   Solution. Given finite-state automata N_A and N_B respectively for A and B, the basic idea is to construct a PDA for recognizing $A \diamond B$ that first simulates N_A and then non-deterministically switchs to simulate N_B. The PDA counts the number of symbols while simulating N_A by pushing a marker onto the stack whenever it reads an input symbol and it later cancels out the markers with the input symbols while simulating N_B.

   Suppose $N_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ and $N_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$, assuming A and B have the same alphabet. We construct the PDA $M = (Q, \Sigma, \Gamma, \delta, q_{\text{start}}, \{q_{\text{accept}}\})$ for $A \diamond B$ as follows:

   - $Q = \{q_{\text{start}}, q_{\text{accept}}\} \cup Q_A \cup Q_B$, where $q_{\text{start}}, q_{\text{accept}} \notin Q_A \cup Q_B$.
   - $\Gamma = \{x, \$\}$.
   - δ is defined as follows.

$$\begin{cases} \delta(q_{\text{start}}, \varepsilon, \varepsilon) = \{(q_A, \$)\} & \\ \delta(q, a, \varepsilon) = \{(q', x) \mid q' \in \delta_A(q, a)\} & q \in Q_A \text{ and } a \neq \varepsilon \\ \delta(q, \varepsilon, \varepsilon) = \{(q', \varepsilon) \mid q' \in \delta_A(q, \varepsilon)\} & q \in Q_A \\ \delta(q, \varepsilon, \varepsilon) = \{(q_B, \varepsilon)\} & q \in F_A \\ \delta(q, a, x) = \{(q', \varepsilon) \mid q' \in \delta_B(q, a)\} & q \in Q_B \text{ and } a \neq \varepsilon \\ \delta(q, \varepsilon, \varepsilon) = \{(q', \varepsilon) \mid q' \in \delta_B(q, \varepsilon)\} & q \in Q_B \\ \delta(q, \varepsilon, \$) = \{(q_{\text{accept}}, \varepsilon)\} & q \in F_B \\ \delta(q, a, t) = \emptyset & \text{otherwise} \end{cases}$$

It should be clear that $L(M) = A \diamond B$; we omit the detailed proof. $\square$

9. Prove, using the pumping lemma, that $\{a^m b^n c^{m+n} \mid m, n \geq 1\}$ is not context free.

   Solution. $\{a^m b^n c^{m+n} \mid m, n \geq 1\}$ is actually context free. It was intended to be $\{a^m b^n c^{m \times n} \mid m, n \geq 1\}$. which is not context free. The proof is left as an exercise. $\square$

10. For languages A and B, let the *perfect shuffle* of A and B be the language $\{w \mid w = a_1 b_1 \cdots a_k b_k$, where $a_1 \cdots a_k \in A$ and $b_1 \cdots b_k \in B$, each $a_i, b_i \in \Sigma\}$. Show that the class of context-free languages is *not* closed under perfect shuffle.

    Solution. Let A be the language $\{0^{2i} 1^i \mid i \geq 1\}$ and B be $\{0^i 1^{2i} \mid i \geq 1\}$. Both are clearly context free. Their perfect shuffle equals $\{(00)^i (01)^i (11)^i \mid i \geq 1\}$, which is not context free. (Note: a string in the perfect shuffle must be the result of shuffling two strings of the *same* length.) $\square$

Appendix

- (Myhill-Nerode Theorem) Given a language $L \subseteq \Sigma^*$, define an equivalence relation R_L over Σ^*:

  $$x R_L y \text{ iff } \forall z \in \Sigma^* (xz \in L \leftrightarrow yz \in L).$$

  With R_L defined as above, the following are equivalent:

  1. L is regular.
  2. R_L is of finite index.

  Moreover, the index of R_L equals the number of states in the smallest DFA that recognizes L.

  Note: the *index* of an equivalence relation is the number of equivalence classes it induces.

- (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 $i \geq 0$, $uv^i xy^i z \in A$, (2) $|vy| > 0$, and (3) $|vxy| \leq p$.