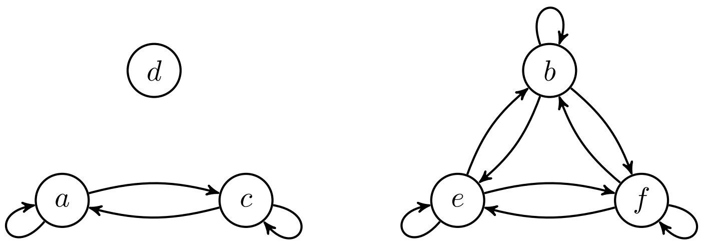
Suggested Solutions to Midterm Problems

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

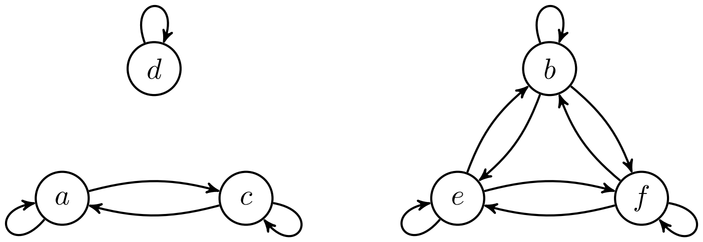
   Solution. (Hung-Wei Hsu)



□

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

   Solution. (Hung-Wei Hsu)



□

2. Let L be a language over Σ (i.e., $L \subseteq \Sigma^*$). Two strings x and y in Σ^* are *distinguishable by L* if for some string z in Σ^*, exactly one of xz and yz is in L. When no such z exists, i.e., for every z in Σ^*, either both of xz and yz or neither of them are in L, we say that x and y are *indistinguishable by L*. Is indistinguishability by a language an equivalence relation (over Σ^*)? Please justify your answer.

   Solution.

   Let us refer to the "indistinguishability by a language L" relation as R_L. R_L is an equivalence relation, as it satisfies the following three conditions:

   - Reflexivity (xR_Lx): For every w in Σ^*, xw and xw are identical and either both or neither of them are in L. Hence, xR_Lx.
   - Symmetry (xR_Ly if and only if yR_Lx): If xR_Ly, i.e., for every w in Σ^*, either both of xw and yw or neither of them are in L, then for every w in Σ^*, both of yw and xw or neither of them are in L and hence yR_Lx.
   - Transitivity (xR_Ly and yR_Lz implies xR_Lz): Suppose xR_Ly and yR_Lz, i.e., for every w in Σ^*, (a) either both of xw and yw or neither of them are in L and (b) either both of yw and zw or neither of them are in L. If both of xw and yw are in L, then

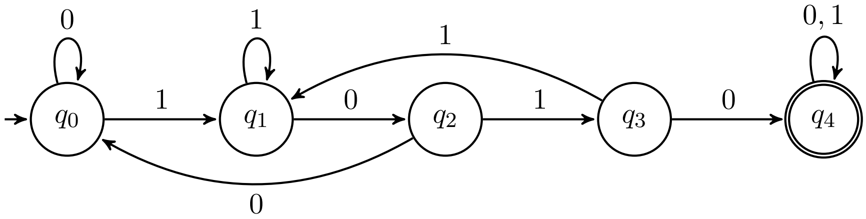
both of yw and zw are also in L and hence both of xw and zw are in L. If neither of xw and yw are in L, then neither of yw and zw are in L and hence neither of xw and zw are in L. So, for every w in Σ^*, either both of xw and zw or neither of them are in L and hence xR_Lz.

□

3. (20 points) Give the state diagrams of DFAs recognizing the following languages. In all parts, the alphabet is $\{\mathtt{0}, \mathtt{1}\}$.

   (a) $\{w \mid w$ contains the substring $\mathtt{1010}$, i.e., $w = x\mathtt{1010}y$ for some x and $y\}$.

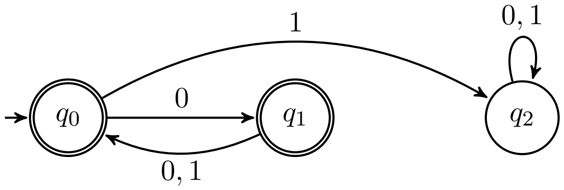
   Solution. (Hung-Wei Hsu)



□

   (b) $\{w \mid$ every odd position of w is a $\mathtt{0}\}$ (Note: see w as $w_1w_2\cdots w_n$, where $w_i \in \{\mathtt{0}, \mathtt{1}\}$).

   Solution. (Hung-Wei Hsu)

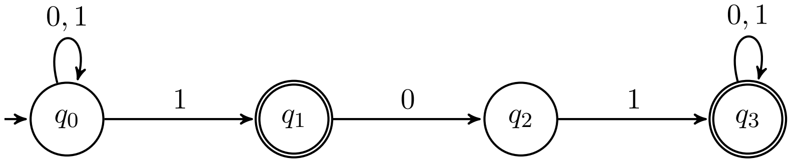


□

4. Let $L = \{w \in \{0, 1\}^* \mid w$ contains 101 as a substring or ends with a 1$\}$.

   (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.

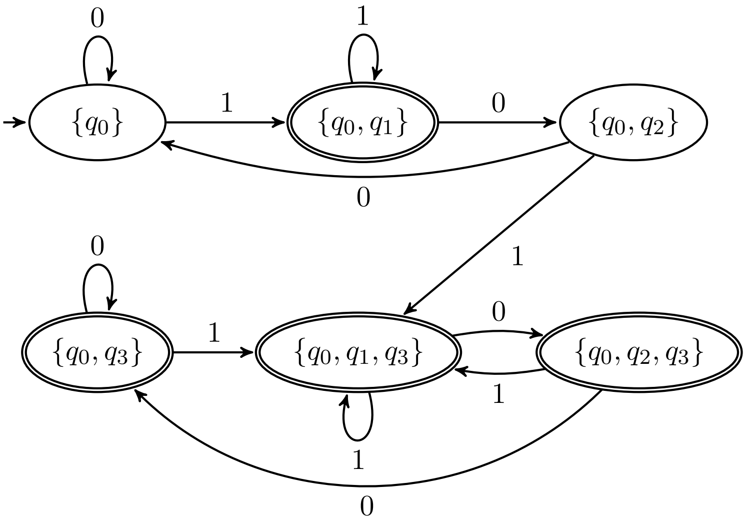
   Solution. (Hung-Wei Hsu)



□

   (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. (Hung-Wei Hsu)

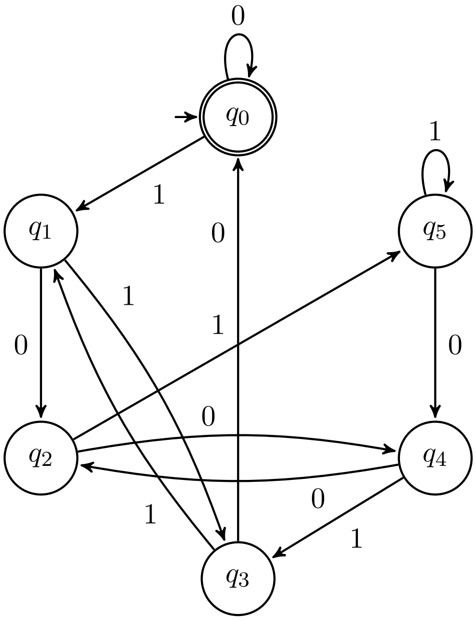


□

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

$$C_6 = \{x \mid x \text{ is a binary number that is a multiple of } 6\}.$$

Solution. (Hung-Wei Hsu)



□

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

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

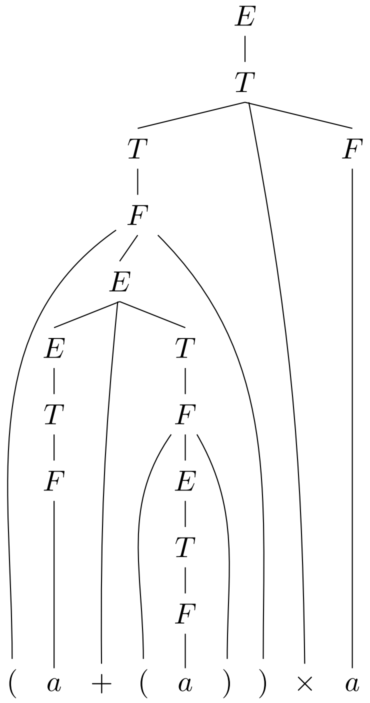
Give the (leftmost) derivation and parse tree for the string $(a + (a)) \times a$.

Solution. (Hung-Wei Hsu)

The leftmost derivation.

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

The parse tree.



□

7. Give a context-free grammar that generates the following language: $\{w \in \{a, b, c\}^* \mid$ the number of a's in w equals that of b's or c's$\}$ (no restriction is imposed on the order in which the input symbols may appear). Please make the CFG as simple as possible and explain the intuition behind it.

Solution. There are several plausible ways of defining the CFG. Below is a more intuitive version:

$$\begin{aligned}
S &\rightarrow T \mid U \\
T &\rightarrow abT \mid aTb \mid Tab \mid baT \mid bTa \mid Tba \mid TT \mid c \mid \varepsilon \\
U &\rightarrow acU \mid aUc \mid Uac \mid caU \mid cUa \mid Uca \mid UU \mid b \mid \varepsilon
\end{aligned}$$

It is equivalent to the following much simpler version:

$$\begin{aligned}
S &\rightarrow T \mid U \\
T &\rightarrow aTb \mid bTa \mid TT \mid c \mid \varepsilon \\
U &\rightarrow aUc \mid cUa \mid UU \mid b \mid \varepsilon
\end{aligned}$$

The production rules for T (analogous for U) can be seen as derived from those for generating strings of balanced parentheses, the main difference being that T allows pairs of parentheses to appear in reversed order. T may be forced to appear in a particular position of a string (of terminals and non-terminals) during a derivation, which means c can be placed in any desired position of a generated string. □

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 nondeterministically 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. □

9. Prove, using the pumping lemma, that $\{1^{n^2} \mid n \geq 0\}$ is not context free.

   Solution. Assume toward contradiction that p is the pumping length for $\{1^{n^2} \mid n \geq 0\}$. Consider a string $s = 1^{p^2}$ in the language. Suppose that s can be pumped by dividing s as $uvxyz = 1^i 1^j 1^k 1^l 1^{p^2-i-j-k-l}$, where $j+l > 0$ ($|vy| \geq 0$) and $j+k+l \leq p$ ($|vxy| \leq p$). If we pump s up to $1^i(1^j)^2 1^k (1^l)^2 1^{p^2-i-j-k-l} = 1^{i+2j+k+2l+p^2-i-j-k-l} = 1^{p^2+j+l}$. As $0 < j+l \leq p$, $p^2 < p^2 + j + l \leq p^2 + p < p^2 + 2p + 1 = (p+1)^2$ and hence $1^i(1^j)^2 1^k (1^l)^2 1^{p^2-i-j-k-l}$ is not in $\{1^{n^2} \mid n \geq 0\}$. So, s cannot be pumped, a contradiction. □

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. (Page 162 of [Sipser 2013])
    Let A be the language $\{0^i 1^i \mid i \geq 0\}$ and B be $\{a^j b^{3j} \mid j \geq 0\}$ (here, the alphabet Σ is $\{0, 1, a, b\}$). Both are clearly context free. Their perfect shuffle equals $\{(0a)^k (0b)^k (1b)^{2k} \mid k \geq 0\}$, which is not context free (a proof by the pumping lemma is similar to that for $\{a^n b^n c^n \mid n \geq 0\}$ and is omitted).

    (Note: a string in the perfect shuffle must be the result of shuffling two strings of the *same* length. So, the total number of 0's and 1's in a string in the perfect shuffle of A and B must equal the total number of a's and b's.) □

Appendix

- Common 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.
- (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) $|vxy| \leq p$.