Suggested Solutions to Midterm Problems

1. 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:

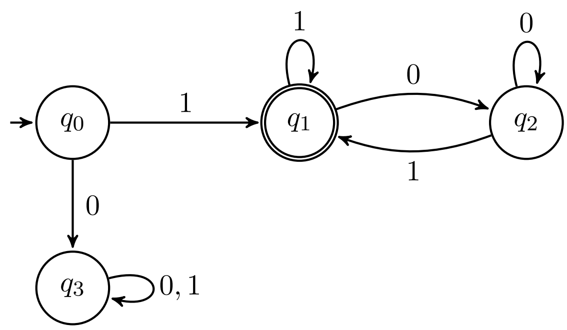
   - Reflexivity (for every x in Σ^*, xR_Lx): For every w in Σ^*, xw and xw are identical and either both or neither of them are in L. Hence, xR_Lx.
   - Symmetry (for every x and y in Σ^*, xR_Ly if and only if yR_Lx): If xR_Ly, i.e., for every w in Σ^*, either both of xw and yw or neither of them are in L, then, for every w in Σ^*, both of yw and xw or neither of them are in L and hence yR_Lx; and vice versa.
   - Transitivity (for every x, y, and z in Σ^*, 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.

   □

2. Give the state diagrams of DFAs, with as few states as possible, recognizing the following languages.

   (a) $\{w \in \{0,1\}^* \mid w \text{ begins with a 1 and also ends with a 1}\}$.

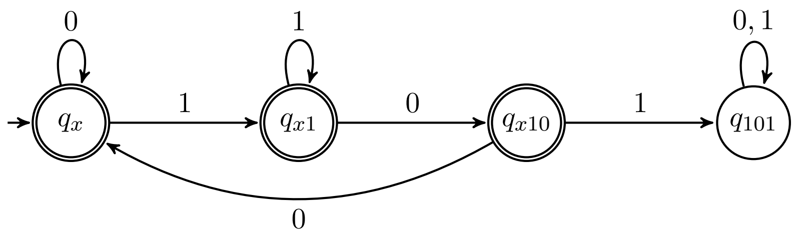
   Solution.



   □

   (b) $\{w \in \{0,1\}^* \mid w \text{ doesn't contain the substring 101}\}$.

   Solution.

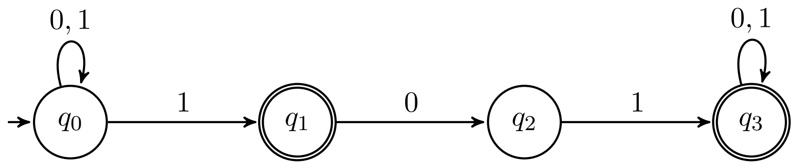


□

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

   (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.



□

   (b) Give a regular expression that describes L. The shorter your regular expression is, the more points you will be credited for this problem.

   Solution. $(0 \cup 1)^* 1(01(0 \cup 1)^* \cup \epsilon)$ or $\Sigma^* 1(01\Sigma^* \cup \epsilon)$, where Σ is a shorthand for $(0 \cup 1)$. □

4. For languages A and B, let the *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 regular languages is closed under shuffle.

   Solution. Let $M_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ and $M_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$ be two DFAs that recognize A and B, respectively. An NFA $M = (Q, \Sigma, \delta, q_0, F)$ that, in each step, simulates either a step of M_A or M_B will recognize the shuffle of A and B. Formally, it is defined as follows:

   - $Q = Q_A \times Q_B$,
   - $\delta((x, y), a) = \{(\delta_A(x, a), y), (x, \delta_B(y, a))\}$ for every $x \in Q_A, y \in Q_B, a \in \Sigma$,
   - $q_0 = (q_A, q_B)$,
   - $F = F_A \times F_B$.

□

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

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

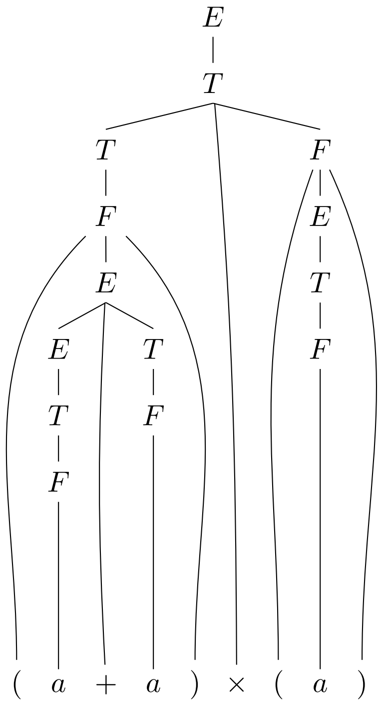
(a) (10 points) Give the (leftmost) derivation and parse tree for the string $(a + a) \times (a)$.

Solution.

The leftmost derivation

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

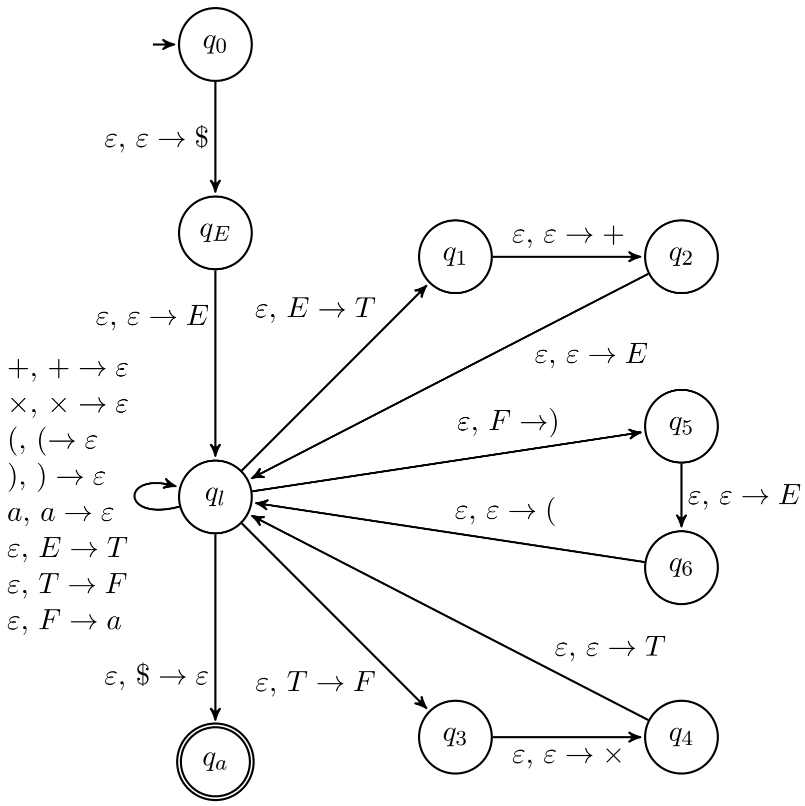
The parse tree



□

(b) (10 points) Convert the grammar into an equivalent PDA (that recognize the same language).

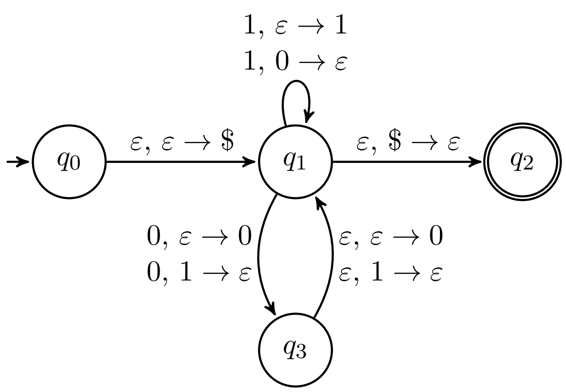
Solution.



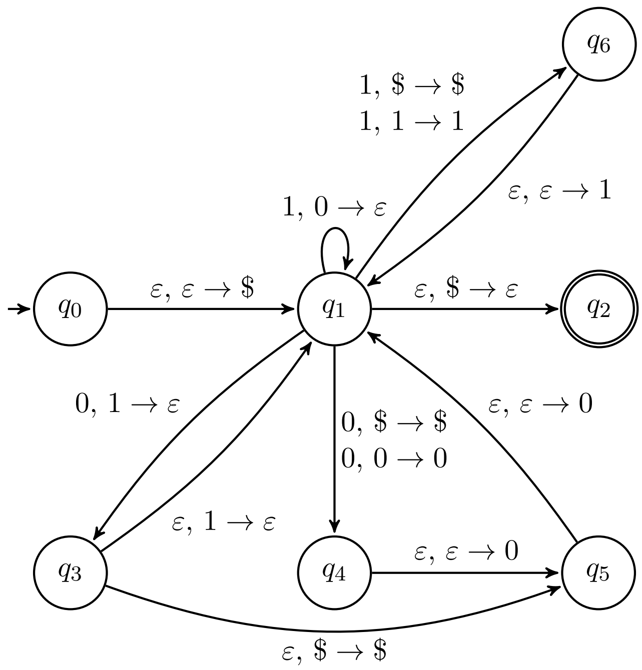
□

6. Draw the state diagram of a PDA that recognizes the following language: $\{w \in \{0,1\}^* \mid w \text{ has twice as many 1s as 0s}\}$. Please make the PDA as simple and deterministic as possible and explain the intuition behind the PDA.

   Solution. A PDA that recognizes the language is shown below. The basic idea is to cancel out every two 1s by a subsequent 0 or the other way around, using the stack to remember outstanding (yet-to-be-cancelled-out) occurrences of 0 or 1. The case when a 0 is read with a 1 outstanding on the stack is effectively the same as a 0 immediately followed by a 1, leaving a 0 on the stack to be cancelled out by a subsequent 1. So, when reading a 1, the PDA pushes a 1 onto the stack or pops a 0 from the stack. When reading a 0, the PDA pushes two 0s onto the stack, pops two 1s from the stack, or (to allow the case when a 0 is read with a 1 outstanding on the stack) pops a 1 from and pushes a 0 onto the stack.



The PDA above is simple enough, but highly nondeterministic. For instance, while there is an outstanding 0 on the stack, the PDA may choose to push a 1 (rather than correctly cancelling out the 0) when reading a 1, even though this choice will turn out to be futile. The following is a more deterministic PDA for the same language.



□

7. Prove each of the following statements:

   (a) (2 points) The class of context-free languages is closed under *union.*

   Solution. Let A and B be two context-free languages. Suppose they may be generated by CFGs (V_A, Σ, R_A, S_A) and (V_B, Σ, R_B, S_B) respectively, where V_A and V_B are disjoint. Then, $(V_A \cup V_B, \Sigma, \{S \rightarrow S_A \mid S_B\} \cup R_A \cup R_B, S)$ will be a CFG that generates $L(A) \cup L(B)$. □

   (b) (4 points) The class of context-free languages is not closed under *intersection.*

   Solution. Let $A = \{a^n b^n c^m \mid n, m \geq 0\}$ and $B = \{a^m b^n c^n \mid n, m \geq 0\}$, which are context free. $A \cap B = \{a^n b^n c^n \mid n \geq 0\}$ is not context free. □

   (c) (4 points) The class of context-free languages is not closed under *complement.*

   Solution. Intersection may be expressed in terms of complement and union: $A \cap B = \overline{\overline{A} \cup \overline{B}}$. From (a) and (b), the class of context-free languages is closed under the union operation, but it is not closed under the intersection operation. If the class of context-free languages were closed under the complement operation, then it would be closed under intersection, contradicting the result in (b). □

8. Let A be the language of all palindromes over $\{0, 1\}$ with equal numbers of 0s and 1s. Prove, using the pumping lemma, that A is not context free. (Note: a *palindrome* is a string that reads the same forward and backward.)

Solution. We take s to be $1^p0^p0^p1^p$, where p is the pumping length, and show that s cannot be pumped. There are basically three ways to divide s into $uvxyz$ such that $|vy| > 0$ and $|vxy| \leq p$:

Case 1: vxy falls (entirely) within the first occurrence of 1^p0^p. No matter what strings v and y get from the division, when we pump down (i.e., $i = 0$), we will lose some 1s or 0s (or both) in the resulting string s'. If we lose some 1s, then there will not be a sufficient number of 1s to match the 1^p in the suffix 0^p1^p and s' is on longer a palindrome. If all 1s remain, then we must lose some 0s and there will be fewer 0s than 1s in s'.

Case 2: vxy falls within the substring 0^p0^p. No matter what strings v and y get from the division, when we pump down (i.e., $i = 0$), there will be fewer 0s than 1s in the resulting string.

Case 3: vxy falls within the second occurrence of 0^p1^p. This is analogous to Case 1.

□

9. Find a regular language A, a non-regular but context-free language B, and a non-context-free language C over $\{0, 1\}$ such that $C \subseteq B \subseteq A$.

   Solution. $A = \{0^i1^j0^k \mid i, j, k \geq 0\}$ is regular. $B = \{0^i1^j0^k \mid i, j, k \geq 0 \text{ and } i \leq j\}$ is context-free but not regular. $C = \{0^i1^j0^k \mid i, j, k \geq 0 \text{ and } i \leq j \leq k\}$ is not context-free. It is apparent that $C \subseteq B \subseteq A$. □

Appendix

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