1. For each statement below, check “True” if the statement is *always* true and “False” otherwise. Each correct answer is worth +1 point; each incorrect answer is worth $-\frac{1}{2}$ point; checking “I don’t know” is worth $+\frac{1}{4}$ point; and flipping a coin is (on average) worth $+\frac{1}{4}$ point.

(a) If the moon is made of cheese, then Jeff is the Queen of England.

Solution: The implication $p \rightarrow q$ is logically equivalent to $\overline{p} \lor q$. The premise (the moon is made of cheese) is false, so the implication is true, whether or not Jeff is actually the Queen of England.

(b) The language $\{0^m1^n \mid m, n \geq 0\}$ is not regular.

Solution: This is the language described by the regular expression $0^*1^*$. 

(c) For all languages $L$, the language $L^*$ is regular.

Solution: $L^*$ is regular for every *regular* language $L$ by definition, but if $L$ is not a regular language, $L^*$ may or may not be regular. For example, if $L$ is the set of all balanced strings of parentheses — described by the grammar $S \rightarrow \epsilon \mid SS \mid (S)$ — then $L^*$ is not regular (because $L^* = L$).

(d) For all languages $L \subset \Sigma^*$, if $L$ is recognized by a DFA, then $\Sigma^* \setminus L$ can be represented by a regular expression.

Solution: If $L$ is regular, then $\Sigma^* \setminus L$ is also regular.

(e) For all languages $L$ and $L'$, if $L \cap L' = \emptyset$ and $L'$ is not regular, then $L$ is regular.

Solution: Recall that the complement of any regular language is regular. It follows immediately that the complement of any non-regular language is non-regular. So let $L'$ be any non-regular language, and let $L = \Sigma^* \setminus L$. Then $L \cap L' = \emptyset$ and $L$ is non-regular.

Alternatively: Consider $L = \{0^n1^n \mid n \geq 1\}$ and $L' = \{2^n3^n \mid n \geq 1\}$.
(f) For all languages $L$, if $L$ is not regular, then $L$ does not have a finite fooling set.

Solution: A fooling set is a set of strings in which every pair of elements has a distinguishing suffix. So any subset of a fooling set is also a fooling set. In particular, the empty language is (vacuously) a fooling set for every language $L$. A language $L$ is regular if every fooling set for $L$ is finite.

(g) Let $M = (\Sigma, Q, s, A, \delta)$ and $M' = (\Sigma, Q, s, Q \setminus A, \delta)$ be arbitrary DFAs with identical alphabets, states, starting states, and transition functions, but with complementary accepting states. Then $L(M) \cap L(M') = \emptyset$.

Solution: This is precisely the canonical construction of a DFA for the complement of a regular language. For any string $w \in \Sigma^*$, if $M$ accepts $w$, then $\delta^*(s, w) \in A$, so $\delta^*(s, w) \not\in Q \setminus A$, and therefore $M'$ does not accept $w$.

(h) Let $M = (\Sigma, Q, s, A, \delta)$ and $M' = (\Sigma, Q, s, Q \setminus A, \delta)$ be arbitrary NFAs with identical alphabets, states, starting states, and transition functions, but with complementary accepting states. Then $L(M) \cap L(M') = \emptyset$.

Solution: For any string $w \in \Sigma^*$, if the set $\delta^*(s, w)$ contains both a state in $A$ and a state in $Q \setminus A$, then both machines accept $w$.

(i) For all context-free languages $L$ and $L'$, the language $L \cdot L'$ is also context-free.

Solution: Suppose $L$ is generated by a context-free grammar with starting variable $A$, and $L'$ is generated by a context-free grammar with starting variable $B$, where the two grammars have disjoint variables. Then $L \cdot L'$ is generated by the union of of the two grammars, with a new start state $S$ and a new production $S \rightarrow AB$.

(j) Every non-context-free language is non-regular.

Solution: Every regular language is context-free. See the lecture notes for a proof.

Rubric: +1 for each correct answer, $-\frac{1}{2}$ for each incorrect answer, $+\frac{1}{4}$ for each “I don’t know.” Explanations (in gray) are not required.
2. For each of the following languages over the alphabet $\Sigma = \{0, 1\}$, either prove that the language is regular or prove that the language is not regular. **Exactly one of these two languages is regular.**

(a) $\{0^n w 0^n \mid w \in \Sigma^+ \text{ and } n > 0\}$

**Solution:** This language is regular. Specifically, this is the language $\{0^\infty \mid w \in \Sigma^+ \text{ and } n > 0\}$.

For every string $x \in L$, we have $x = 0^n w 0^n$ for some $w \in \Sigma^+$ and $n > 0$; in particular, we have $x = 0 y 0$ for some string $y = 0^{n-1} w 0^{n-1} \in \Sigma^+$. Conversely, for every non-empty string $w \in (0 + 1)^+$, the string $0 w 0 \in \Sigma^*$ is in $L$.

**Solution:** This language is regular. Here is a DFA that recognizes it:

![DFA Diagram]

**Rubric (tentative):** 5 points = 2 for “regular” + 3 for DFA/NFA/regular expression (as in HW1 problem 4). These are not the only correct answers. A proof that the given DFA, NFA, or regular expression is correct is not required. Similarly, an English description/explanation is not required, but it may help us give you partial credit.

(b) $\{w0^n w \mid w \in \Sigma^+ \text{ and } n > 0\}$

**Solution:** This language is not regular.

Consider the set $F = 1^\infty 0$.

Let $x$ and $y$ be arbitrary distinct strings in $F$.

Then $x = 1^i 0$ and $y = 1^j 0$ for some positive integers $i \neq j$.

Let $z = 1^i$.

Then $xz = 1^i 0 1^i \in L$, but $yz = 1^j 0 1^i \notin L$.

So $F$ is a fooling set for $L$. Because $F$ is infinite, $L$ cannot be regular.

**Rubric:** 5 points = 2 for “not regular” + 1 for infinite fooling set + 2 for fooling set proof (as in HW2 problem 2). This is not the only correct solution.
3. Let \( L = \{ 0^{2i}1^{i+2j} \mid i, j \geq 0 \} \) and let \( G \) be the following context-free-grammar:

\[
S \rightarrow AB \\
A \rightarrow \varepsilon \mid \varepsilon A1 \\
B \rightarrow \varepsilon \mid 11B0
\]

(a) Prove that \( L(G) \subseteq L \).

**Solution:** First, we claim that every string in \( L(A) \) is equal to \( 0^{2i}1^i \) for some non-negative integer \( i \).

**Proof:** Let \( w \) be an arbitrary string in \( L(A) \). Assume for every string \( x \in L(A) \) such that \( |x| < |w| \) that \( x = 0^{2j}1^j \) for some non-negative integer \( j \). There are two cases to consider, mirroring the productions from \( A \).

- If \( w = \varepsilon \), then \( w = 0^{20}1^0 \).
- Suppose \( w = \varepsilon x1 \) for some \( x \in L(A) \). The induction hypothesis implies \( x \in L_1 \), so \( x = 0^{2j}1^j \) for some non-negative integer \( j \). It follows that \( w = \varepsilon 000^{2j}1^j1 = 0^{2j+1}1^j+1 \).

In both cases, we conclude that \( w = 0^{2i}1^i \) for some integer \( i \geq 0 \), as claimed. \( \square \)

After swapping \( \varepsilon \leftrightarrow 1 \) everywhere, exactly the same proof implies that every string in \( L(B) \) is equal to \( 1^{2i}0^j \) for some non-negative integer \( i \).

Finally, let \( w \) be an arbitrary string in \( L(G) = L(S) \). The production \( S \rightarrow AB \) implies that \( w = xy \) for some string \( x \in L(A) \) and \( y \in L(B) \). The two previous claims imply that \( x = 0^{2i}1^i \) and \( y = 1^{2j}1^j \), and therefore \( w = 0^{2i}1^i \cdot 1^{2j}1^j = 0^{2i+2j}1^j+1 \) for some integers \( i \) and \( j \).

**Rubric:** 5 points = 3 for \( L(A) \) (standard induction rubric) + 1 for \( L(B) \) + 1 for \( L(S) \).

(b) Prove that \( L \subseteq L(G) \).

**Solution:** First, we claim that \( 0^{2n}1^n \in L(A) \) for every non-negative integer \( i \).

**Proof:** Let \( n \) be an arbitrary non-negative integer. Assume for all \( 0^{2m}1^m \in L(A) \) for every non-negative integer \( m \leq n \). There are two cases to consider.

- If \( n = 0 \), then \( 0^{2n}1^n = \varepsilon \), so the production rule \( A \rightarrow \varepsilon \) implies \( w \in L(A) \).
- If \( n > 0 \), then \( 0^{2n}1^n = 000x1 \), where \( x = 0^{2n-2}1^{n-1} \). The induction hypothesis implies \( x \in L(A) \). So the production rule \( A \rightarrow 00A1 \) implies \( A \rightarrow 00A1 \rightarrow 00x1 = w \).

In both cases, we conclude that \( 0^{2n}1^n \in L(A) \), as claimed. \( \square \)

After swapping \( \varepsilon \leftrightarrow 1 \) everywhere, exactly the same proof implies \( 1^{2n}0^n \in L(B) \) for every non-negative integer \( n \).

Finally, let \( i \) and \( j \) be arbitrary non-negative integers. The two previous claims imply \( A \rightarrow^* 0^{2i}1^i \) and \( B \rightarrow^* 1^{2j}0^j \), so the production \( S \rightarrow AB \) implies the derivation

\[
S \rightarrow AB \rightarrow^* 0^{2i}1^i B \rightarrow^* 0^{2i}1^i 1^{2j}0^j = 0^{2i+2j}0^j.
\]

We conclude that \( 0^{2i+2j}0^j \in L(S) \). \( \square \)

**Rubric:** 5 points = 3 for \( L(A) \) (standard induction rubric) + 1 for \( L(B) \) + 1 for \( L(S) \).
4. For any language $L$, let $\text{Suffixes}(L) := \{ x | \text{there exists } y \in \Sigma^* \text{ such that } x = yx \}$ be the language containing all suffixes of all strings in $L$. For example, if $L = \{000, 100, 110, 111\}$, then $\text{Suffixes}(L) = \{\epsilon, 0, 1, 00, 10, 11, 000, 100, 110, 111\}$.

Prove that for any regular language $L$, the language $\text{Suffixes}(L)$ is also regular.

**Solution (one new state):** Let $M = (\Sigma, Q, s, A, \delta)$ be an arbitrary DFA that recognizes $L$.

Without loss of generality, we assume that every state in $Q$ is reachable from the start state $s$; that is, for every $q \in Q$, there is some string $w \in \Sigma^*$ such that $\delta(s, w) = q$. Otherwise, we simply discard any unreachable states to get a smaller DFA that still recognizes $L$.

We define an NFA $M' = (\Sigma, Q', s', A', \delta')$ as follows:

- $Q' = Q \cup \{s'\}$
- $s'$ is a new explicit state
- $A' = A$
- $\delta'(s', \epsilon) = Q$
- $\delta'(s', a) = \emptyset$ for all $a \in \Sigma$
- $\delta'(q, \epsilon) = \emptyset$ for all $q \in Q$
- $\delta'(q, a) = \{\delta(q, a)\}$ for all $q \in Q$ and $a \in \Sigma$

In other words, we add a new start state $s'$ with $\epsilon$-transitions to every other state. ■

**Solution (ghost copy):** Let $M = (\Sigma, Q, s, A, \delta)$ be an arbitrary DFA that recognizes $L$.

We define an NFA $M' = (\Sigma, Q', s', A', \delta')$ as follows:

- $Q' = Q \times \{0, 1\}$
- $s' = (s, 0)$
- $A' = \{(q, 1) | q \in Q\}$
- $\delta'( (q, 0), \epsilon ) = \{ \delta(q, a), 0 \} \cup \{ (q, 1) \}$ for all $q \in Q$
- $\delta'( (q, 0), a ) = \emptyset$ for all $q \in Q$ and $a \in \Sigma$
- $\delta'( (q, 1), \epsilon ) = \emptyset$ for all $q \in Q$
- $\delta'( (q, 1), a ) = \{ \delta(q, a), 1 \}$ for all $q \in Q$ and $a \in \Sigma$

In other words, we make a duplicate “ghost” copy of $Q$, replace all the transitions in the ghost copy with $\epsilon$-transitions, and then add an $\epsilon$-transition from each ghost state to its original counterpart. Only states in the second copy actually accept. The ghost copy of $M$ “reads” the missing prefix $y$ before passing control to the original $M$. ■
Rubric (tentative): 10 points =
+ 2 for a formal, complete, and unambiguous description of an NFA. (No credit for the rest of the problem if this is missing.)
+ 6 for a correct NFA = 3 for accepting all suffixes + 3 for accepting only suffixes, except:
  — 1 for a single typo or similar mistake
  — 1 for $\varepsilon$-transitions to inaccessible states
  — 2 for rejecting $\varepsilon$ even when $\varepsilon \notin L$. ($\varepsilon$ is a suffix of every string.)
+ 2 for a brief English explanation. A formal proof of correctness not required.
(This rubric is similar to HW2 problems 1 and 3.) These are not the only correct solutions.
5. For each of the following languages $L$, give a regular expression that represents $L$ and describe a DFA that recognizes $L$. You do not need to prove that your answers are correct.

(a) The set of all strings in $\{0, 1\}^*$ that do not contain the substring $0110$.

**Solution:** Here are three regular expressions:

- $1^* (00^*(1 + 1111^*))^* 0^* 1^*$ — Every run of 1s, except possibly the runs that start and end the string, either has length 1 or has length at least 3.
- $1^* (\emptyset + 01 + \emptyset 1111^*)^* 1^*$ — Every $\emptyset$ except the last is followed immediately by zero, one, or at least three 1s.
- $(1 + \emptyset (0 + 10)^* 111)(\emptyset + \emptyset (0 + 10)^*(\epsilon + 1 + 11))$ — Extracted from the DFA below using Han and Wood's algorithm.

**Rubric:** 5 points = 2½ for regular expression + 2½ for DFA (using rubric from HW1). Only one regular expression is required. Explanation (in gray) is not required. These are not the only correct solutions.
(b) The set of all strings in \( \{0,1\}^* \) that contain exactly one of the substrings 01 or 10.

**Solution:** \( 0^+ 1^+ + 1^+ 0^+ = 0^* 011^* + 1^* 100^* \)

Any string that does *not* contain the substring 10 must be in the language \( 0^+ 1^* \), and any string \( w \in 0^* 1^* \) contains the substring 01 if and only if \( \#(0,w) > 0 \) and \( \#(1,w) > 0 \).

![DFA Diagram]

**Rubric:** 5 points = 2½ for regular expression + 2½ for DFA (using rubric from HW1). Only one regular expression is required. Explanation (in gray) is not required. These are not the only correct solutions.

During the exam several people asked which of the following interpretations of the phrase “exactly one of the substrings” is correct:

- \( \#(01,w) + \#(10,w) = 1 \)
- \( \#(01,w) > 0 \iff \#(10,w) = 0 \)

These two interpretations are actually equivalent!