CS/ECE 374 B: Algorithms & Models of Computation (Spring 2020)

Midterm 1

Monday Feburary 24, 2020
7pm – 9pm in ECEB 1002

Solution to midterm 1

1 (20 PTS.) For each statement below, check "True" if the statement is ***always*** true and "False" otherwise. Each correct answer is worth two points; each incorrect answer is worth nothing – checking "I don't know" is worth $\frac{1}{2}$ a point.

1.A. For every language L, if L is not regular, then L does not have a finite fooling set. **False**: X **True**: ☐

Solution:

Any language has a finite fooling set which is the empty set $\emptyset$. Any finite subset of an infinite fooling set is also a fooling set.

1.B. The language $\left\{0^i 1^j 0^k \mid i + j + k \geq 2020\right\}$ is not regular. **False**: X **True**: ☐

Solution:

This language is $0^*1^*0^* \setminus \left\{0^i 1^j 0^k \mid i + j + k < 2020\right\}$. $0^*1^*0^*$ is regular. $\left\{0^i 1^j 0^k \mid i + j + k < 2020\right\}$ is finite and regular. Regular languages are closed under set difference. Hence, the language is regular.

1.C. The language $\left\{0^i 1^j 2^k 3^\ell \mid 0 \leq i \leq j \leq k \leq \ell\right\}$ is not regular. **False**: ☐ **True**: X

1.D. For every regular language L, the language $\left\{0^{|w|} \mid w \in L\right\}$ is context-free. **False**: ☐ **True**: T

Solution:

Yep. Given a DFA for L, we can construct and NFA for $\left\{0^{|w|} \;\middle|\; w \in L\right\}$ by replacing every transition on 1 with a transition on 0. Hence, $\left\{0^{|w|} \;\middle|\; w \in L\right\}$ is regular which is also context free.

1.E. Consider two languages L and L' that are not regular. If $L' \setminus L$ is regular, then $L \cup L'$ not regular. Hint: $L = (L \cup L') \cap (L \cup \overline{L'})$

False: $\square$ **True:** $\boxed{\text{T}}$

Solution:

The statement is is true. $L = (L \cup L') \cap (L \cup \overline{L'}) = (L \cup L') \cap \overline{(L' \setminus L)}$

$\overline{(L' \setminus L)}$ is regular since regular languages are closed under complement. Assume $L \cup L'$ is regular, then L is regular since regular languages are closed under intersection. This is a contradiction since L is not regular. Hence, $L \cup L'$ is not regular.

1.F. Consider two languages L and L'. If L is not regular, L' is regular, and $L \cup L'$ is regular, then $L \cap L'$ is regular.

False: $\boxed{\text{X}}$ **True:** $\square$

Solution:

The statement is is false. Consider $L' = \Sigma^*$, then$L$ is not regular, L' is regular, and $L \cup L' = L'$ is regular, but $L \cap L' = L$ is not regular.

1.G. The language 1^*0^* is generated by the CFG: $S \to \varepsilon \mid 1S0$.

False: $\boxed{\text{X}}$ **True:** $\square$

Solution:

The CFG generates $\{0^n1^n \mid n \geq 0\}$.

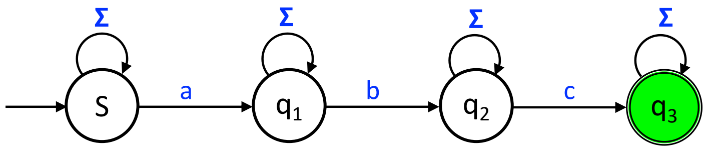
1.H. Consider a language L. If there is an NFA that rejects every string $w \notin L$, then L is regular.

False: $\boxed{\text{X}}$ **True:** $\square$

Solution:

This statement is False. Consider $L = \{0^n1^n \mid n \geq 0\}$ and consider an NFA that accepts the language $L' = \varepsilon$. Then, the NFA rejects every string $w \notin L$. It also rejects most string in L as well.

1.I. The following NFA accepts the language of all strings that contain the substring abc.



False: X **True**: ☐

Solution:

It accepts $\Sigma^* a \Sigma^* b \Sigma^* c \Sigma^*$

1.J. Let $M(\{0,1\}, Q, s, A, \delta)$ be an arbitrary DFA and $M'(\{0,1\}, Q, s, A, \delta')$ be the DFA obtained from M by changing every 0-transition into a 1-transition and vice versa. More formally, M and M' have the same states, input alphabet, starting state, accepting state, but $\delta'(q,0) = \delta(q,1)$ and $\delta'(q,1) = \delta(q,0)$, Then, $L(M) \cap L(M') = \emptyset$.

False: X **True**: ☐

Solution:

Consider the machine M that accepts $L(M) = \{001, 110, 11\}$. Then, $L(M') = \{110, 001, 00\}$. Hence, $L(M) \cap L(M') \neq \emptyset$.

2 In the following, $\Sigma = \{a, b\}$. You can provide a short justifications to clarify your answer but you **do not** need to prove your answer is correct.

2.A. (6 PTS.) Provide a regular expression for the following language where x^R is the reverse string of x.

$$L = \left\{ xwx^R \mid w \in \Sigma^*, x \in \Sigma^+ \right\}.$$

Solution:

$a(a+b)^*a + b(a+b)^*b$

2.B. (6 PTS.) Provide a regular expression for the set of all strings in Σ^* that either end with aaa and have odd length or end with aba and have even length.

Solution:

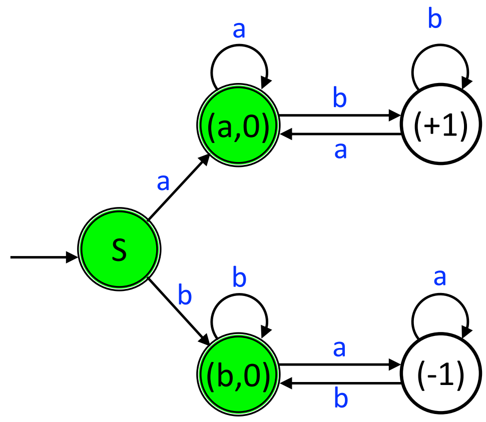
$(aa+ab+ba+bb)^*aaa + (aa+ab+ba+bb)^*(a+b)aba$

2.C. (8 PTS.) Provide a DFA for the set of all strings in Σ^* in which substrings ab and ba appear the same number of times. You can either draw the DFA or describe it formally. In both cases, you have to clearly describe your states, start state, accept states, and transition function.

Solution:

The idea is to only count the difference between the number of times ab appears and the number of times ba appears. The difference can only be $0, +1$, or -1 since after ab appears, it will not appear again until ba appears and vice versa. For the case where the difference is 0, we should also track the last seen symbol. Formally, $M = (\Sigma, Q, s, A, \delta)$ be the DFA then,

$$\begin{aligned}
s &= s \\
Q &= \{s, (a,0), (b,0), (-1), (+1)\} \\
A &= \{s, (a,0), (b,0)\} \\
\delta(s, a) &= (a, 0); \quad \delta(s, b) = (b, 0); \\
\delta((a,0), a) &= (a, 0); \quad \delta((a,0), b) = (+1); \\
\delta((b,0), a) &= (-1); \quad \delta((b,0), b) = (b, 0); \\
\delta((-1), a) &= (-1); \quad \delta((-1), b) = (b, 0); \\
\delta((+1), a) &= (a, 0); \quad \delta((+1), b) = (+1);
\end{aligned}$$



3 Consider the following two definitions, defined for any language L.

$$\mathit{RemoveMiddleThird}(L) := \{xz \mid xyz \in L \text{ for some } y, |x| = |y| = |z|\}$$

$$\mathit{RemoveLastThird}(L) := \{xy \mid xyz \in L \text{ for some } z, |x| = |y| = |z|\}$$

For example, if $L = \{11, 010, 100\}$, then $\mathit{RemoveMiddleThird}(L) = \{00, 10\}$ whereas $\mathit{RemoveLastThird}(L) = \{01, 10\}$.

3.A. (10 PTS.) Let $L = 0^*10^*$, prove that $L' = \mathit{RemoveLastThird}(L)$ is regular.

Solution:

After removing the last third, the remaining string must be even length. There are 3 cases:

- 1 is part of the last third and, hence, the remaining strings are in $(00)^+$.
- 1 is not part of the last third and the remaining strings have an odd number of zeros to the left of 1 i.e. $(00)^*01(00)^*$
- 1 is not part of the last third and the remaining strings have an odd number of zeros to the right of 1 i.e. $(00)^*10(00)^*$

Thus, L' is regular because it can be recognized by the following regular expression: $(00)^+ + (00)^*10(00)^* + (00)^*01(00)^*$

3.B. (10 PTS.) Let $L = 0^*10^+10^*$, prove that $L' = \mathit{RemoveMiddleThird}(L)$ is not regular. (Hint: Find an infinite fooling set.)

Solution:

Consider the fooling set $F = 0^*11$.

Let u and v be arbitrary strings in F.

Then, $u = 0^m11$ and $v = 0^n11$ for some non-negative integers $m < n$.

Let $s = 0^m$ be a distinguishing suffix.

Lemma 0.1. $us = 0^m110^m \in L'$

Proof: Let $w = (0^m1)(0^{m+1})(10^m) \in L$, then $w = xyz$ where $|x| = |y| = |z|$, $x = (0^m1), y = (0^{m+1})$, and $z = (10^m)$. Then, $xz \in L'$, but $xz = 0^m110^m = us$. Hence, $us \in L$. ∎

Lemma 0.2. $vs = 0^n110^m. \notin L'$

Proof: Assume $vs \in L'$ then $\exists w = xyz \in L$ such that $|x| = |y| = |z|$ and $xz = vs = 0^n110^m$. Since $m < n$, then 11 is a substring in z. Since w contains only two 1, $y \in 0^*$. Then, $w \in 0^*110^*$. Hence, $w \notin L$ since the 1s in L must be separated by at least one 0. But this is a contractions. Hence, $vs \notin L'$.

Thus, F is a fooling set for L'.

Because F is infinite, L' cannot be regular.

4 (20 PTS.) Assume $\Sigma = \{0, 1\}$. Let *lnum* : $\Sigma^* \to \mathbb{N}$ be the numeric value of a binary string in "*little endian*" form, i.e. the most significant bit (MSB) is on the right. Define *lnum* recursively as follows:

$$lnum(w) = \begin{cases} 0 & w = \varepsilon \\ a + 2 \times lnum(x) & w = ax \end{cases}$$

For example, $lnum(1011) = 13$, $lnum(01100) = 6$.

Let L_1 and L_2 be two arbitrary regular languages. Define the language $Add(L_1, L_2)$ as follows:

$$Add(L_1, L_2) = \{w \mid lnum(w) = lnum(x) + lnum(y), x \in L_1 \text{ and } y \in L_2, |x| = |y| = |w|\}.$$

Roughly speaking, this language consists of all strings w such that you can take some string x from L_1 and some string y from L_2 that add up to w. However, the three strings x, y, and w must have equal length which means there is no truncation or overflow/wraparound. For example, if $L_1 = \{10110, 001, 00\}$ and $L_2 = \{01100, 01\}$, then $Add(L_1, L_2) = \{11001, 01\}$.

Prove that $Add(L_1, L_2)$ is regular. In praticular, construct an NFA that accepts the language. Briefly argue why your construction is correct, i.e., why the NFA accepts all strings in the language and why all strings accepted by the NFA must be in the language.

Solution:

Solution: The basic approach is to use a 1-bit State machine to keep track of the carry bit, while also simulating a DFA for each of L_1 and L_2.

Each time we read a symbol c, it's the next symbol of the sum w, hence it corresponds sum of *some* symbol a in x and *some* symbol b in y (plus the current carry bit cb). We use NFA multiple transitions to "guess" the possible next symbols (a, b) that are consistent with the truth table (the symbol a and the carry bit cb), and then simulate the corresponding steps in L and R.

This truth table works through which values of a, b are consistent with the given sum c and current carry bit cb, and if so what the new carry bit *cb'* should be:

cb	c	$a = 0, b = 0$	$a = 0, b = 1$	$a = 1, b = 0$	$a = 1, b = 1$
0	0	$cb' = 0$			$cb' = 1$
1	0		$cb' = 1$	$cb' = 1$	
0	1		$cb' = 0$	$cb' = 0$	
1	1	$cb' = 0$			$cb' = 1$

Let $M_1 = (Q_1, \delta_1, s_1, A_1)$ be a DFA for L_1, $M_2 = (Q_2, \delta_2, s_2, A_2)$ be a DFA for L_2.

The states are $Q = Q_1 \times Q_2 \times \{0, 1\}$, or in other words $\{(q_1, q_2, cb) \mid q_1 \in Q_1, q_2 \in Q_2, cb \in \{0, 1\}\}$. The transitions are:

$$\delta((q_1, q_2, 0), 0) := \{(\delta_1(q_1, 0), \delta_2(q_2, 0), 0), (\delta_1(q_1, 1), \delta_2(q_2, 1), 1)\}$$

$$\delta((q_1, q_2, 1), 0) := \{(\delta_1(q_1, 0), \delta_2(q_2, 1), 1), (\delta_1(q_1, 1), \delta_2(q_2, 0), 1)\}$$

$$\delta((q_1, q_2, 0), 1) := \{(\delta_1(q_1, 0), \delta_2(q_2, 1), 0), (\delta_1(q_1, 1), \delta_2(q_2, 0), 0)\}$$

$$\delta((q_1, q_2, 1), 1) := \{(\delta_1(q_1, 0), \delta_2(q_2, 0), 0), (\delta_1(q_1, 1), \delta_2(q_2, 1), 1)\}$$

The accepting states are: $A := \{(q_1, q_2, 0) \mid q_1 \in A_1, q_2 \in A_2\}$ where q_1 is accept in L_1 and q_2 is accept in L_2, and the final carry bit at the end must be zero.

The start state is $s := (s_1, s_2, 0)$.

5 (20 PTS.) CONTEXT FREE LANGUAGES.

Construct a CFG for the following languages. Assume $\Sigma = \{0, 1\}$. In order to get full credit you should briefly explain how your grammar works, and the role of each nonterminal. (You do not need to prove your grammar is correct.)

5.A. (8 PTS.) $L_1 = \left\{0^{3i}1^{2i+j}0^{2j} \mid i, j \geq 0\right\}$

Solution:

$$
\begin{aligned}
S_1 &\rightarrow AB \\
A &\rightarrow \varepsilon \mid 000A11 \\
B &\rightarrow \varepsilon \mid 1B00
\end{aligned}
$$

5.B. (8 PTS.) $L_2 = \left\{1^i0^j1^k \mid i = j \text{ or } j = k, \quad i, j, k \geq 0\right\}$

Solution:

$$
\begin{aligned}
S_2 &\rightarrow A \mid B \\
A &\rightarrow CD \\
C &\rightarrow \varepsilon \mid 1C0 \\
D &\rightarrow \varepsilon \mid D1 \\
B &\rightarrow EF \\
E &\rightarrow \varepsilon \mid 1E \\
F &\rightarrow \varepsilon \mid 0F1
\end{aligned}
$$

5.C. (4 PTS.) $L_3 = L_1L_2 + L_1^*$. where L_1 and L_2 are the languages from parts A and B. (Hint: rename the start terminals of L_1 and L_2 to S_1 and S_2 respectively and use closure properties to construct the CFG).

Solution:

$$
\begin{aligned}
S_3 &\rightarrow S_1S_2 | A \\
A &\rightarrow \varepsilon \mid S_1A
\end{aligned}
$$