HW 4 Solution

CS/ECE 374 B: Algorithms & Models of Computation, Spring 2020 Version: **1.0**

Submitted by:

- **≪Michael Jiang≫**: **≪minhaoj2≫**
- **≪Yumai Sun≫**: **≪yumais2≫**
- **≪Shuchen Zhang≫**: **≪szhan114≫**

1 Problem 10:

Solution 10.A:

Language L represents all strings over Σ with odd length.

Solution 10.B:

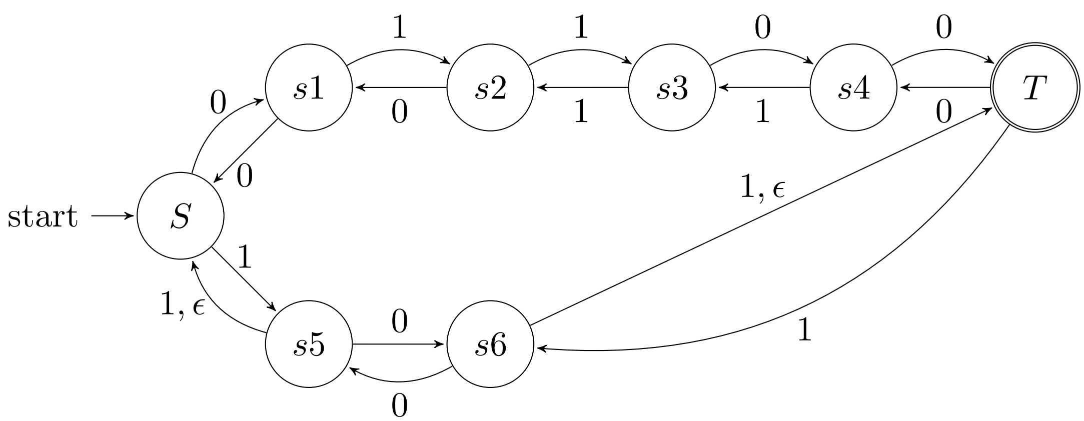

Figure 1: NFA

Solution 10.C:

Suppose the DFA for L is $M_1 = \{Q, \Sigma, \delta, s, A\}$. Suppose we can construct an NFA, $M = \{Q', \Sigma', \delta', s', A'\}$ which accepts all strings in language f(L). And the NFA is defined as follows:
Suppose $\omega_n \in \Sigma, c \in \{0, 1, 2, ..N\}$ where N = max(length(f(s_n))). For any string ω that can be accepted by L, suppose $|\omega|$ = M, where ω can be written as $\omega = \omega_1\omega_2\omega_3..\omega_M$, where for ω_1 to $\omega_M \in \Sigma$.

- Q' = Q $\times \Sigma \times \{0, 1, 2, ..N\}$
- $\Sigma' = \{0, 1\}$
- A' = (A, ω_M, length(f(ω_M)))
- s = (s, ω_1, 0)
- And δ' can be defined as:

  As $n \in \{1, 2, ...M\}$

$$\delta'((q,\omega_n,c),\epsilon) = \begin{cases} (\delta(q,\omega_n),\omega_{n+1},0), & \text{if } f(\omega_n)=\epsilon \\ (q,\omega_n,c), & \text{Otherwise.} \end{cases}$$

$$\delta'((q,\omega_n,c),a) = \begin{cases} (q,\omega_n,c+1), & \text{if } c+1<k, \text{ where k = length(f(}\omega_n\text{)) and a = f(}\omega_n\text{)[c].} \\ (\delta(q,\omega_n),\omega_{n+1},0), & \text{if } c+1=k \text{ ,and a = f(}\omega_n\text{)[c]} \\ (q,\omega_M,\text{length}(\omega_M)), & \text{if c + 1 = k and a = f(}\omega_M\text{)[length(}\omega_M\text{-1)] and n = M} \end{cases}$$

In this definition, f(ω_n)[c] represents the character at c^{th} position, where the indexing starts from 0.

Proof of correctness: To prove the correctness of this construction of NFA, that the NFA M accepts all strings of f(L), we need to show that f(L) $\subseteq$ L(M) and L(M) $\subseteq$ f(L).

Claim 1: f(L) $\in$ L(M), that is, every string in f(L) can be accepted by M.

We will prove by induction that this claim holds for every string in f(L). Suppose the string ω is in L.

Base Case: When $\omega = \epsilon$, that is, no input for f function. Therefore, in this case, the claim is naturally true.

Inductive Hypothesis: Suppose for some $n \in \mathbb{N}$, for any string $\omega \in$ L with length less than k, the claim holds for ω, that is, f(ω) is accepted by M.

Inductive Step: Let the string u with length k be in language L. By the definition of string, we can write u as $\omega \bullet a$ where $a \in \Sigma$. By the construction of our NFA and inductive hypothesis, our transition for string u is $\delta'((q,\omega_{M-1},f(\omega_{k-1})[-1]),f(\omega_k)[0])$ where $f(\omega_{k-1})[-1])$ represents the last digit of $f(\omega_k)$. In other words, with the transition from ω_{k-1} to ω_k in L, we add states between this transition for f(ω_k) where f(ω_k) = $a_1a_2..a_n$. And using the non-determinism of NFA, the transition between these states are only valid when $a = a_i$ for each $a \in \Sigma'$ and $a_i \in f(\omega_k)$. In this case, our construction of this NFA would create a path from ω_{k-1} to ω_k. In other words, the string f(u) would be accepted by this NFA M. Therefore, by the property of induction, we proved that every string in f(L) will be accepted by M.

In conclusion, we proved that f(L) $\in$ L(M).

Claim 2: L(M) $\in$ f(L), that is, every string that can be accepted by M belongs to f(L).

We can consider from the following two cases. Suppose string ω can be accepted by M.

- When $\omega = \epsilon$, by definition, f(ϵ) = ϵ. By the construction of NFA, it will be accepted by M. Since f(ϵ) $\in$ f(L) since $\epsilon \in$ L. Therefore, $\omega \in$ f(L).
- When ω is non-empty. Since ω can be accepted by the NFA, by the construction of NFA, $\omega = f(x_1)f(x_2)$ ...$f(x_n)$ for $x_i \in \Sigma$. By the definition of code f, $f(x_1)f(x_2)..f(x_n) = f(x_1x_2..x_n)$. Therefore, since $x_1x_2..x_n \in \Sigma^*$ by the definition of language L, $\omega \in f(L)$.

Therefore, we proved that for any string that can be accepted by NFA M is in language L.

Together with Claim 1 and Claim 2, we proved that the NFA defined above is indeed the desired language f(L).

2 Problem 11:

Solution 11.A:

Consider the infinite set $F = \{0^n1 | n \geq 0\}$. Let x and y be any arbitrary strings in F such that $x = 0^i1$ and $y = 0^j1$ for some integers $i \neq j$.

In this case, we can have a suffix $\omega = 2^i$. Therefore, with this choice of ω, we have that $x\omega \in L$ since $i + 1 = (i) + 1$. And $y\omega \notin L$ since $j + 1 \neq i + 1$. Thus, we get that for every pair of distinct strings in F has a distinguishing suffix. In other words, F is a fooling set for L.

Because F is infinite, L cannot be regular.

Solution 11.B:

Suppose F = L itself is the fooling set for L. Let x and y be any arbitrary strings in F, such that $x = 0^{i^3}$ and $y = 0^{j^3}$ for some integers $i \neq j$. And without the loss of generality, assume $i < j$.

In this case, we can have a suffix $\omega = 0^{3i^2+3i+1}$. Therefore, with this choice of ω, we have that $x\omega \in L$ since $x\omega = 0^{i^3+3i^2+3i+1} = 0^{(i+1)^3}$. And $y\omega \notin L$ since $y\omega = 0^{j^3+3i^2+3i+1}$ where $j^3 + 3i^2 + 3i + 1$ cannot be represented as any integer's cube since $i < j, j^3 < j^3 + 3i^2 + 3i + 1 < (j + 1)^3$. Therefore, $y\omega \notin L$. Thus, we get that for every pair of distinct strings in F has a distinguishing suffix. In other words, F = L is a fooling set for L.

Because L itself is infinite, L cannot be regular.

Solution 11.C:

Consider the infinite set $F = \{0(01)^n | n \geq 0\}$. Let x and y be any arbitrary strings in F such that $x = 0(01)^i$ and $y = 0(01)^j$ for some integers $i \neq j$.

In this case, we can have a suffix $\omega = (10)^i1$. Therefore, with this choice of ω, we have that $x \in L$ since $x\omega = 0(01)^i(10)^i1$, which suffices the description of language L. And $y\omega \notin L$ since $y\omega = 0(10)^j(01)^i1$ where $i \neq j$. Therefore, $y\omega \notin L$. Thus, we get that for every pair of distinct strings in F has a distinguishing suffix. In other words, F is a fooling set for L.

Because F is infinite, L cannot be regular.

Solution 11.D:

Lemma 1: Regular language is closed under difference, that is, suppose regular language A and regular language B, then A - B is also regular.

Proof of Lemma 1: Since we have a regular language A and a regular language B. Suppose $M_1 = \{Q_1, \sum_1, \delta_1, s_1, \mathrm{A}_1\}$ is a DFA that accepts language A and $M_2 = \{Q_1, \sum_1, \delta_1, s_1, \mathrm{A}_1\}$ is a DFA that accepts language B. Therefore, we can easily construct a DFA M_3 that is the product automata of M_1 and M_2 where the accepting states of M_3 is $\{(q_1, q_2) | q_1 \in A_1, q_2 \notin A_1\}$. Therefore, M_3 would be the DFA that accepts the language A - B, which is regular by trivial.

Now we will prove by contradiction that if L is not regular, and $L \cup L'$ where L' is finite is not regular.

Suppose L is not regular, L' is finite and $L \cup L'$ is regular. Since any finite language must be regular, so L' is regular. And let L_1 be $(L \cup L') - L$, which is a subset of L'. Since L' is a finite language, L_1 must also be finite, which is simply regular.

In this case, we have $L = (L \cup L') - L_1$, where $L \cup L'$ and L_1 are both regular, by lemma 1, L is regular, which is a contradiction to the assumption that L is not regular.

Therefore, by the proof of contradiction, $L \cup L'$ is not regular.

Example for $L \cup L_1$ is regular when L' is infinite: Let L be $\{0^n 1^n | n \geq 0\}$ and $L' = 0^* 1^*$. In this case, $L \cup L' = 0^* 1^*$ which is naturally regular.

3 Problem 12:

Solution 12.A:

Since $i + j + k + l = t$, we can define a CFG, G = {V, T, P, S} where the start symbol S = S, terminals T = $\{0, 1, 2, 3, 4\}$, the set of non-terminals V = $\{S, S_1, S_2, S_3, S_4\}$ and the production rules P are as:

$S \to S_1S_2S_3S_4 \vert \epsilon$	
$S_1 \to 0S_13 \vert S_2 \vert \epsilon$	0 occurs with 3
$S_2 \to 1S_23 \vert S_3 \vert \epsilon$	1 occurs with 3
$S_3 \to 2S_33 \vert \epsilon$	2 occurs with 3
$S_4 \to 3S_44 \vert \epsilon$	3 occurs with 4, 4 is behind 3

Solution 12.B:

We can write the complement of $\{0^n1^n|n \geq 0\}$ as $\{0^n1^m|m \neq n\} \cup (\{0,1\}^* \setminus 0^*1^*)$ since $\{0^n1^n|n \geq 0\} \cup \{0^n1^m|m \neq n\} = 0^*1^*$. Therefore, a CFG can be constructed G = $\{V, T, P, S\}$, where the set of variables V = $\{S, A, B, C, D, E, F\}$, the terminals T = $\{0, 1\}$, the start symbol S = S, and the production rules are as:

$S \to A \vert B$	
$A \to C10C$	$(0+1)^*10(0+1)^*$
$C \to \epsilon \vert 0C \vert 1C$	$(0+1)^*$
$B \to D \vert E$	$\{0^n1^m \vert n \neq m\}$
$D \to 0D \vert 0F$	$\{0^n1^m \vert n > m\}$
$E \to E1 \vert F1$	$\{0^n1^m \vert n < m\}$
$F \to \epsilon \vert 0F1$	$\{0^n1^m \vert n = m\}$

Solution 12.C:

Since $2(i + j) = k$, we can convert this into that every occurrence of 0 or 1 is accompanied with two 2's. Therefore, we can construct the CFG that G = {V, T, P, S}, where the set of variables V = $\{S, S_1\}$, terminals T = $\{0, 1, 2\}$, start symbol S = S and the production rules are as:

$S \to 0S22 \vert S_1 \vert \epsilon$	022 occurrence
$S_1 \to 1S_122 \vert \epsilon$	122 occurrence

Solution 12.D:
We separately prove L $\subseteq$ L(G) and L(G) $\subseteq$ L as follows:

Claim 1: L(G) $\subseteq$ L, that is, every string in L(G) satisfies that $\{0^i1^j2^k|k = 2(i+j)\}$.

Proof: Let $\Delta(u) = \#(2,u) - 2(\#(1,u) + \#(0,u))$. We need to prove that $\Delta(\omega) = 0$ for every string $\omega \in$ L(G). Let ω be an arbitrary string in L(G), and consider an arbitrary derivation of ω of length k. Assume that $\Delta(x) = 0$ for every string $x \in$ L(G) that can be derived with fewer than k productions. We can consider in cases, depending on the first one or two productions of ω.

- If $\omega = \epsilon$, which can be derived from S $\rightarrow \epsilon$ or from S $\rightarrow$S$_1$ $\rightarrow \epsilon$, then $\#(2,u) = 2(\#(1,u) + \#(0,u))$ by definition, so $\Delta(\omega) = 0$.
- Suppose the deviation is S $\rightarrow 0S22 \rightarrow^* \omega$. Then $\omega = 0x22$ for some $x \in$ L(G), where x can be derived with fewer than k productions. The inductive hypothesis implies that $\Delta(x) = 0$. It immediately follows that $\Delta(\omega) = 0$.
- Suppose the deviation is S $\rightarrow S_1 \rightarrow 1S_122 \rightarrow^* \omega$. Then $\omega = 1x22$ for some $x \in$ L(G). The inductive hypothesis implies that $\Delta(x) = 0$. It immediately follows that $\Delta(x) = 0$.

In all cases, we conclude that $\Delta(\omega) = 0$, as required.

Claim 2: L $\subseteq$ L(G), that is, G generates every string as $\{0^i1^j2^k|k = 2(i+j)\}$.

Proof: Let $\Delta(u) = \#(2,u) - 2(\#(1,u) + \#(0,u))$. Let ω be an arbitrary string with the form $\{0^i1^j2^k|k = 2(i+j)\}$. Assume that G generates every string x that is shorter than ω and has the same form. There are two cases to consider:

- If $\omega = \epsilon$, then $\epsilon \in$ L(G) because of the production S $\rightarrow \epsilon$ and the production S $\rightarrow$S$_1$ $\rightarrow \epsilon$.
- Suppose ω is non-empty. Then we can have the following three cases:
  - Suppose $\omega = 0^i2^{2i}$. In this case, we can rewrite ω as $00^{i-1}2^{2i-2}22$. As indicated by the inductive hypothesis, $0^{i-1}2^{2i-2}$, which $\in$ L, can be produced by L(G). Therefore, the production rule S $\rightarrow$ 0S22 implies that $\omega \in$ L(G).
  - Suppose $\omega = 1^j2^{2j}$. In this case, we can rewrite ω as $11^{j-1}2^{2j-2}22$. As indicated by the inductive hypothesis, $1^{j-1}2^{2j-2}$, which $\in$ L, can be produced by L(G). Therefore, the production rule S $\rightarrow S_1 \rightarrow$ $1S_122$ implies that $\omega \in$ L(G).
  - Suppose $\omega = 0^i1^j2^{2(i+j)}$. In this case, we can rewrite ω as $0(0^{i-1}1^j2^{2j}2^{2i-2})22$. As proved in the case I and indicated in the inductive hypothesis, we know that since $0^{i-1}1^j2^{2j}2^{2i-2} \in$ L(G), with the same reasoning in case I, by the production rule S $\rightarrow$ 0S22, $\in$ L(G).

In all cases, we concluded that G generates every string with the form $\{0^i1^j2^k|k = 2(i+j)\}$.

Together, with Claim 1 and Claim 2, we proved that L = L(G).