CSC240 Winter 2024 Homework Assignment 9

My name and student number: Haoyun (Bill) Xi 1009992019
The list of people with whom I discussed this homework assignment: None.

1. Consider the following iterative algorithm that finds the length of the longest increasing subsequence in the array $A[1..n]$.

```
1  L[1] ← 1
2  for i ← 2 to n do
3      L[i] ← 1
4      for j ← 1 to i − 1 do
5          if (A[j] < A[i]) and (L[j] ≥ L[i]) then L[i] ← L[j] + 1
6  m ← L[n]
7  for i ← 2 to n − 1 do
8      if L[i] > m then m ← L[i]
```

   (a) Give a precise statement of what it means for this algorithm to be partially correct.
   Solution.
   Denote $\{A_i\}_{i=1}^n$ as the sequence representation of $A[1..n]$.
   Definitions:
   $I = \{I_i\}_{i=1}^n$ is a sequence of ordered indices of $\{A_i\}_{i=1}^n$ if $\forall c, j \in [k], 1 \leq I_c \leq n$ AND $((j < i)$ IMPLIES $I_j < I_c)$
   S is an subsequence of A if $S = \{A_{I_i}\}_{i=1}^k$ for some sequence of ordered indices I of A and $K = |I|$. In this case we say $S \in I_{sub}(A)$ where $I_{sub}(A)$ contains all inc-sub-seq. of A

   Define the predicate for all $n \in \mathbb{Z}^+$, $P(n)$ = "for all $A[1..n]$ (arrays of elements from a totally ordered domain), if the above program terminates, it will ends with $m = \max\{len(s)|s \in I_{sub}(A)\}$". This algorithm is partially correct iff $\forall n \in \mathbb{Z}^+.P(n)$.

   (b) Prove that this algorithm is partially correct.
   Solution.

   Proof.
   Let $n \in \mathbb{Z}^+$ be arbitrary and consider an arbitrary array $A[1..n]$ with a defined total order. Line 1 will initialize the first element of L to 1. We will analyze the effect of line $2-5$ by proving the below loop invariant immediately after ith $(2 \leq i \leq n-1)$ iteration ends.
   Lemma 1. $L[i] = \max\{len(s)|s \in I_{sub}(A[1..i])$ AND $s_{-1} = A_i\}$ where negative indices count from the back to start and we measure the iteration number by the value of i.
   Furthermore, we asusme that the program eventually terminates (so every loop terminates).
   Proof. Before the loop starts, $L[1] = 1$. The only subsequence (which is increasing) is $\{A_i\}_{i=1}^1$. Thus, $L[1]$ does capture the maximum length of increasing subsequence.

   Consider an arbitrary ith iteration $(2 \leq i \leq n)$. Here the "first" iteration is the $(i = 2)$th iteration. Assume that for all $c < j, L[c] = \max\{len(s)|s \in I_{sub}(A[1..c])$ AND $s_{-1} = A_c\}$ (Hypothesis 1). We will prove a second lemma to assist the analysis of line $4-5$.

Lemma 2. If $1 \leq j \leq i-1$, immediately after the jth iteration of line 4-5, we have $L[i] = \max(\{len(s) | s \in I_{sub}(A[1..i]) \text{ AND } s_{-1} = A_i \text{ AND } s_{-2} \in \{A\}_1^j\} \cup \{1\})$.
Proof: Before the inner loop starts, $L[i] = 1$ and since the first set is empty because A_0 is undefined, the maximum of such set union with $\{1\}$ is simply 1.
Let $1 \leq j \leq i-1$ be an arbitrary iteration number and assume for all $k < j$, after the kth iter, $L[i] = \max(\{len(s) | s \in I_{sub}(A[1..i]) \text{ AND } s_{-1} = A_i \text{ AND } s_{-2} \in \{A\}_1^j\} \cup \{1\})$.
In plain word, this loop invariant means the maximum length of sub-sequence which the last element is A_i and the second last element must be from elements before and including A_j if there exists such element to make the sub-sequence an increasing one.
We will call L as the list before execution of the jth iter, and L' is the one after.
Case 1: $A[i] > A[j]$ and $L[j] \geq l[i]$

By specialization of our hypothesis 1 under the assumption of lemma 1 and $j < i$, we have $L[j] = \max\{len(s) | s \in I_{sub}(A[1..j]) \text{ AND } s_{-1} = A_j\}$, which denote the length of the longest sub-sequence ending with A_j. Call such sequence s and define $s' = s \cdot A[i]$. Since $A[i] > A[j]$, $s' \in I_{sub}(A[1..i])$ is a increasing sub-sequence too. In addition, we have $|s'| = |s| + 1$. Since s is already the longest inc.sub.seq without $A[i]$, s' must be the longest inc.sub.seq with $A[i]$ as the last element while the second last element is in $\{A\}_1^j$. Hence, $L'[i] = \max(\{len(s) | s \in I_{sub}(A[1..i]) \text{ AND } s_{-1} = A_i \text{ AND } s_{-2} \in \{A\}_1^j\} \cup \{1\})$ does hold under this case, because the condition on line 5 is true and will update the maximum by one (from $|s| \to |s'|$).

Case 2. $A[i] > A[j]$ but $L[j] < L[i]$

In this case, the condition on line 5 is false and $L[i]$ will not change. By our assumption of the inner loop invariant, after executing line 5, the new $L'[i] = L[i]$ will represent the maximum length of $s \in I_{sub}(A[1..i])$ where the last element of s is A_i and the second last (if any) must from $\{A\}_1^{j-1}$. Consider the new set of $s' \in I_{sub}(A[1..i])$ where the last element of s' is A_i and the second last (if any) must from $\{A\}_1^j$. There are only two possibilities about s'. First, $s'_{-2} \in \{A\}_1^{j-1}$. Second, $s'_{-2} = A_j$. In the first possibility, such maximum length of s is $L[i] = L'[i]$ as above stated. In the second possibility, $s' = \omega \cdot A_j$ for some $\omega \in I_{sub}(A[1..j])$ and $\omega_{-1} = A_j$. By our first loop hypothesis, the length of $|\omega|$ is at most $L[j]$. Thus, $|s'| \leq L'[i]$. Therefore we can see, $L'[i]$ does represent the maximum length of those $s \in I_{sub}(A[1..i])$ where the last element of s is A_i and the second last (if any) must from $\{A\}_1^j$

Case 3: $A[i] \leq A[j]$

Just like Case 2 described, the condition fails and $L'[i] = L[i]$ remains its property above. Again consider the new set of $s' \in I_{sub}(A[1..i])$ where the last element of s' is A_i and the second last (if any) must from $\{A\}_1^j$. We will continue the discussion of two possibilities. First, as justified above, if $s'_{-2} \in \{A\}_1^{j-1}$., $L[i]$ is the maximum by induction hypothesis. Second, if $s'_{-2} = A_j$, then $s' \notin I_{sub}(A[1..i])$ as $A_i \leq A_j$. Thus, the maximum length of all such s' is $L'[i]$.

Hence, by proof by cases, lemma 2 is true.

Since the loop will terminate by our hypothesis above, also from line 4, the inner loop will finish with $j = i-1$. Hence after line 4-5, $L[i] = \max(\{len(s) | s \in I_{sub}(A[1..i]) \text{ AND } s_{-1} = A_i \text{ AND } s_{-2} \in \{A\}_1^{j-1}\} \cup \{1\})$.
Since for every $s \in I_{sub}(A[1..i])$ AND $s_{-1} = A_i$, it is either that $|s| = 1$, or that to

satisfy the definition of sub-sequence, $|s| \geq 1$ and $s_{-2} \in \{A\}_1^{j-1}$. Hence, the maximum length of s is $L[i]$ after executing line 4-5. Thus, we proved lemma 1.

Since the outer loop will also terminate, also from line 2, line 2-5 will end with $i = n$. By lemma 1, for each $1 \leq i \leq n$, $L[i] = \max\{len(s) | s \in I_{sub}(A[1..i]) \text{ AND } s_{-1} = A_i\}$.

Lemma 3. For an arbitrary $2 \leq i \leq n-1$, immediately after the ith iteration of line 7-8, $m = \max\{len(s) | s \in I_{sub(A)} \text{ AND } (s_{-1} \in (A_n \cup \{A\}_1^i))\}$.

Proof. For base case, before the $(i = 2)$th (first) iteration starts, $m = L(n)$. By the conclusion after lemma 1, $m = \max\{len(s) | s \in I_{sub(A)} \text{ AND } (s_{-1} = A_n)$. Consider an arbitrary iteration i and assume before it starts, m satisfy our hypothesis. We will show m', the value immediately after the iteration, also satisfy the hypothesis.

Case 1: $L[i] > m$

Condition on line 8 is true. Thus, $m' = L[i]$. Consider an arbitrary $s \in I_{sub(A)}$ AND $(s_{-1} \in (A_n \cup \{A\}_1^i))$. If $s_{-1} \in (A_n) \cup \{A\}_1^{i-1}$, by the invariant hypothesis, $|s| \leq m$. If $s_{-1} = A_i$, then the maximum of such $|s|$ is given by $L[i]$. Since $L[i] > m$, $m' = L[i]$ does represent the maximum length of $s \in I_{sub(A)}$ AND $(s_{-1} \in (A_n \cup \{A\}_1^i))$.

Case 2: $L[i] \leq m$

Condition on line 8 is false. Thus, $m' = m$. Consider the same definition and cases in case 1. If $s_{-1} \in (A_n) \cup \{A\}_1^{i-1}$, the maximum of such s_{-1} is given by $m = m'$. If $s_{-1} = A_i$, the maximum of such $|s|$ is $L[i]$. Since $l[i] \leq m$, the overall maximum of all possibilities of s is $m' = m$.

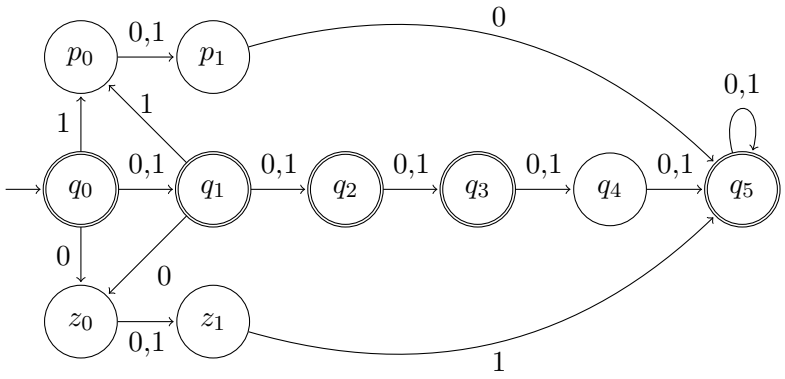
Hence, lemma 3 is also true.

Since the loop will terminate with $i = n - 1$, after executing line 8, the program ends with $m = \max\{len(s) | s \in I_{sub}(A) \text{ AND } (s_{-1} \in (A_n \cup \{A\}_1^{n-1}))\}$. Observe that $(s_{-1} \in (A_n \cup \{A\}_1^{n-1}))$ is always true as s is never an empty sub-sequence. Therefore, $m = \max\{len(s) | s \in I_{sub}(A)\}$. Thus, $P(n)$ is true as $A[1..n]$ is arbitrary.

Since $n \in \mathbb{Z}^+$ is arbitrary, by generalization, $\forall n \in \mathbb{Z}^+.P(n)$. The algorithm is partially correct.

2. For each $k \in \mathbb{Z}^+$, let $X_k = \{x \in \{0,1\}^* : \text{NOT}(\exists y \in \{0,1\}^k.(x = y \cdot y))\}$
and consider the NFA $N_k = (Q, \{0,1\}, \delta, q_0, F)$, where:
$Q = \{q_i : 0 \leq i \leq 2k+1\} \cup \{p_i : 0 \leq i \leq k-1\} \cup \{z_i : 0 \leq i \leq k-1\}$,
$F = \{q_i : 0 \leq i \leq 2k-1\} \cup \{q_{2k+1}\}$,
$\delta(q_i, 0) = \{q_{i+1}, z_0\}$ for $0 \leq i \leq k-1$,
$\delta(q_i, 1) = \{q_{i+1}, p_0\}$ for $0 \leq i \leq k-1$,
$\delta(q_i, 0) = \delta(q_i, 1) = \{q_{i+1}\}$ for $k \leq i \leq 2k$,
$\delta(q_{2k+1}, 0) = \delta(q_{2k+1}, 1) = \{q_{2k+1}\}$,
$\delta(z_i, 0) = \delta(z_i, 1) = \{z_{i+1}\}$ for $0 \leq i \leq k-2$,
$\delta(z_{k-1}, 1) = \{q_{2k+1}\}$,
$\delta(z_{k-1}, 0) = \emptyset$,
$\delta(p_i, 0) = \delta(p_i, 1) = \{p_{i+1}\}$ for $0 \leq i \leq k-2$,
$\delta(p_{k-1}, 0) = \{q_{2k+1}\}$,
$\delta(p_{k-1}, 1) = \emptyset$, and
$\delta(q, \lambda) = \emptyset$ for all $q \in Q$.

Here is a drawing for N_2:



(a) For each state $q \in Q$, describe the set of strings $w \in \{0,1\}^*$ such that $q \in \delta^*(q_0, w)$. Your descriptions should not mention δ.

Solution.
For each $q \in Q$, let $R(q) = \{\omega \in \{0,1\}^* : q \in \delta^*(q_0, w)\}$.

1. for $1 \leq i \leq 2k, R(q_i) = \{\omega \in \{0,1\}^* : |\omega| = i\}$
Explanation: for all $0 \leq i' \leq 2k-1, q_{i'+1}$ can and only can be obtained from $\delta(q_{i'}, 0)$ or $\delta(q_{i'}, 1)$. The $(2k+1)^{\text{th}}$ term can be reached by other means.

2. $R(p_0) = \{\omega \in \{0,1\}^* : \exists j \in [k].w_j = 1 \text{ AND } |\omega| = j\}$
Explanation: p_0 can be reached from $\delta(q_i, 1)$ where $0 \leq i \leq k-1$. Since q_i can be reached and stopped by any string of length i, p_0 must have a 1 after some arbitrary length i string. Thus, p_0 can and only can be stopped by strings having a 1 in the first k digits and as last digit.

3. $R(z_0) = \{\omega \in \{0,1\}^* : \exists j \in [k].w_j = 0 \text{ AND } |\omega| = j\}$
Explanation: for the exact same reasoning above.

4. for $1 \leq i \leq k-1, R(p_i) = \{\omega \in \{0,1\}^* : \exists j \in [k].w_j = 1 \text{ AND } |\omega| = i+j\}$
Explanation: for $0 \leq i' \leq k-2$, $p_{i'+1}$ is only from $\delta(p_{i'}, 0)$ or $\delta(p_{i'}, 1)$. Thus, the set can stop at p_i is the set of elements in the previous one add by another 0 or 1. Hence, the i^{th} one's set is the string from p_0 with additional length i.

5. for $1 \leq i \leq k-1, R(z_i) = \{\omega \in \{0,1\}^* : \exists j \in [k].w_j = 0 \text{ AND } |\omega| = i+j\}$
Explanation: same with above and from "$\delta(z_i, 0) = \delta(z_i, 1) = \{z_{i+1}\}$ for $0 \leq i \leq k-2$".

6. $R(q_{2k+1}) = \{\omega \in \{0,1\}^* : |\omega| \geq 2k+1\} \cup \{\omega \cdot 0 \cdot \gamma : \omega \in R(p_{k-1}), \gamma \in \{0,1\}^*\} \cup \{\omega \cdot 1 \cdot \gamma : \omega \in R(z_{k-1}), \gamma \in \{0,1\}^*\}$
Explanation: there are in total four path leads to q_{2k+1}. First, by adding a 0 or 1 to an

arbitrary string in $R(q_{2k})$. Second, by adding a zero to an arbitrary string in $R(p_{k-1})$. Third, by adding a one to an arbitrary string in $R(z_{k-1})$. Since p_{2k+1} also points to itself for both extra 0s and 1s, in the above three sets, after satisfying the above one of three conditions, it can be concatenated with any finite strings and still stays at q_{2k+1}.

(b) Prove that $L(N_k) = X_k$ for all $k \in \mathbb{Z}^+$.

Proof. By definition, $L(N_k) = \{x \in \{0,1\}^* : \delta^*(q_0, x) \cap (\{q_i : 0 \leq i \leq 2k-1\} \cup \{q_{2k+1}\}) \neq \phi\}$ from definition 2.6, chapter 2, An Introduction to Formal Languages and Automata.

Let $k \in \mathbb{Z}^+$ be arbitrary.
We will prove $L(N_k) = X_k$ by proving the "IFF": $\forall x \in \{0,1\}^*.[x \in L(N_k)$ IFF $x \in X_k]$.

Let $x \in \{0,1\}^*$ be arbitrary.
Case 1: $|x| \neq 2k$
It is easy to see that NOT$(\exists y \in \{0,1\}^k.(x = y \cdot y))$ is true. If the inside is true, then $|x| = |y| + |y| = k + k = 2k$ must be true too, which contradicts with $|x| \neq 2k$.
Thus, $x \in X_k$ is true ($X_k = \{x \in \{0,1\}^* :$ NOT$(\exists y \in \{0,1\}^k.(x = y \cdot y))\}$).
We also claim that $x \in L(N_k)$ is true.
Assume $|x| = j < 2k$ for some $j \in \mathbb{N}$.
From part one, we know that $x \in R(q_j)$. This is equivalent with $q_j \in \delta^*(q_0, x)$.
Since $0 \leq j \leq 2k-1, q_j \in F$. Hence, $q_j \in (\delta^*(q_0, x) \cap F)$ so $\delta^*(q_0, x) \cap F \neq \phi$.
This means $|x| < 2k$ IMPLIES $x \in L(N_k)$
Assume $|x| > 2k$. ($|x| \geq 2k + 1$)
Thus, $x \in R(q_{2k+1})$. Since $R(q_{2k+1}) \in F$, as above deduced, $\delta^*(q_0, x) \cap F \neq \phi$.
This means $|x| > 2k$ IMPLIES $x \in L(N_k)$.
Combine the two cases and $|x| \neq 2k$, we have $x \in L(N_k)$ is true as well.
Thus, $|x| \neq 2k$ IMPLIES ($x \in L(N_k)$ IFF $x \in X_k$) as both $x \in L(N_k)$, $x \in X_k$ are true.

Case 2: $|x| = 2k$ and $\exists y \in \{0,1\}^k.(x = y \cdot y)$.
By definition of $X_k, x \in X_k$ is False. Since $x = y \cdot y$, it is direct that for all $i \in [k]$, $x_i = x_{i+k}$ because the first half and second half are equal strings.
We claim that $x \notin F$ as well. First since $|x| = 2k$, it cannot be the case that $x \in q_{i'}$ for some $1 \leq i' \leq 2k-1$. Hence the only possibility for x to be accepted is in $R(q_{2k+1})$.
Consider the first part of the union $\{\omega \in \{0,1\}^* : |\omega| \geq 2k+1\}$. $|2k| \geq 2k+1$ is false, so x is not in the first part. Consider the second part $\{\omega \cdot 0 \cdot \gamma : \omega \in R(p_{k-1}), \gamma \in \{0,1\}^*\}$, assume for contradiction that x is in it and define ω, γ as above. $\omega \in R(p_{k-1})$ suggests $\exists j \in [k].w_j = 1$ AND $|\omega| = k - 1 + j$. However, by specialization, $x_j = x_{j+k} = 1$.
Thus, 0 is not concatenated after ω, a contradiction that says $x \notin$ in the second part.
Consider the third part $\{\omega \cdot 1 \cdot \gamma : \omega \in R(z_{k-1}), \gamma \in \{0,1\}^*\}$. With the above logic, we have $\exists j \in [k].w_j = 0$ AND $|\omega| = k - 1 + j$ but $x_j = x_{j+k} = 0$. Thus, $x \notin R(q_{2k+1})$.

We have exhausted every final states to conclude x is not a path from p_0 to any of them. Hence $\delta^*(q_0, x) \cap \phi$ so $x \in L(N_k)$ is False as well. (Recall that $x \in X_k$ is False)
$|x| = 2k$ and $\exists y \in \{0,1\}^k.(x = y \cdot y)$ IMPLIES ($x \in L(N_k)$ IFF $x \in X_k$)

Case 3: $|x| = 2k$ and NOT$(\exists y \in \{0,1\}^k.(x = y \cdot y))$

Since x is a concatenation of 2 length-k strings (but not the same one), the first half must differs at some indices i with the second half. Thus, $\exists i \in [k].x_i \neq x_{i+k}$.

Assume $x_i = 1$. Consider the string $\omega = x[1 : i + (k-1)]$ (denotes the sub-string of x from index 1 to index $i + (k-1)$, inclusive), $|w| = i + k - 1$.

By construction, $\exists j \in [k].w_j = 1$ AND $|\omega| = k - 1 + j$ if we change all i to j.

Hence, $\omega \in R(p_{k-1})$. Since $x_i \neq x_{i+k}$, we have $x_{i+k} = 0$.

Consider $\gamma = x[i + k + 1 : 2k]$ (where $\gamma = \lambda$ if $i = k$), we can rewrite x as $\omega \cdot 0 \cdot \gamma$, where $\omega \in R(p_{k-1}), \gamma \in \{0,1\}^*$. Hence, $x \in \{\omega \cdot 0 \cdot \gamma : \omega \in R(p_{k-1}), \gamma \in \{0,1\}^*\}$ $\subseteq R(q_{2k+1})$. Hence, $q_{2k+1} \in (\delta^*(q_0, x) \cap F)$ so $\delta^*(q_0, x) \cap F \neq \phi$.

Thus, $x_i = 1$ IMPLIES $x \in L(N_k)$.

Assume $x_i = 0$. For the same construction of ω, γ as above, $x = \omega \cdot 1 \cdot \gamma$, where $\omega \in R(z_{k-1})$ and $\gamma \in \{0,1\}^*$. Hence, $x \in \{\omega \cdot 1 \cdot \gamma : \omega \in R(z_{k-1}), \gamma \in \{0,1\}^*\} \subseteq R(q_{2k+1})$

It follows the same reasoning that $x_i = 0$ IMPLIES $x \in L(N_k)$.

Combine both cases, $x \in L(N_k)$. Since the assumption NOT$(\exists y \in \{0,1\}^k.(x = y \cdot y))$ also says $x \in X_k$, we have $x \in X_k$ IFF T IFF $x \in L(N_k)$.

$|x| = 2k$ and NOT$(\exists y \in \{0,1\}^k.(x = y \cdot y))$ IMPLIES $[x \in L(N_k)$ IFF $x \in X_k]$.

Case 2, 3 together says $|x| = 2k$ IMPLIES $[x \in L(N_k)$ IFF $x \in X_k]$. Combine with case 1 where $|x| \neq 2k$, we have proved by cases that $[x \in L(N_k)$ IFF $x \in X_k]$ is always true.

Since $x \in \{0,1\}^*$ is arbitrary, $\forall x \in \{0,1\}^*.[x \in L(N_k)$ IFF $x \in X_k]$. Equivalently, $L(N_k) = X_k$.

Since $k \in \mathbb{Z}^+$ is arbitrary, $L(N_k) = X_k$ for all $k \in \mathbb{Z}^+$.