CS 420
Final Exam Practice Question Solutions

Below are some practice questions for the final exam. There will be questions on the actual exam that do not resemble any of these questions. Other questions on the exam may ask about similar material but be phrased in a different way. Still, doing all of these practice questions will help you prepare for the exam.

1. Let G be the following Chomsky Normal Form grammar.

$$\begin{aligned} S &\rightarrow AB|CA|a \\ A &\rightarrow BC|a|b \\ B &\rightarrow CC|c \\ C &\rightarrow AC|b \end{aligned}$$

Here is a partially filled in table for the algorithm described in Theorem 7.16 to determine if a particular string belongs to $L(G)$.

i \ j	1	2	3	4	5
1	B	A	$\emptyset$	?	?
2	X	A, C	S	B, C	?
3	X	X	A, S	C	B, C, S
4	X	X	X	A, C	B, C, S
5	X	X	X	X	A, C

(a) What is the string w for this table? (For this grammar, you can tell just by looking at the table.)
<u>*cbabb*</u>

(b) What are the three missing entries in the table? Give complete answers.

$$\begin{aligned} table(1,4) &= \underline{A, C} \\ table(2,5) &= \underline{A, B, C, S} \\ table(1,5) &= \underline{A, B, C, S} \end{aligned}$$

(c) Is w in $L(G)$? <u>Yes</u>

(d) How does your answer to (c) follow from your answer to (b)?
<u>S is in $table(1,5)$</u>

2. Let A be the language $\{w\#w^R | w \in \{0,1\}^*\}$.

(a) Give an implementation-level description of a one-tape Turing machine that decides the language A in time $O(n^2)$. Give a brief explanation of why your Turing machine runs in time $O(n^2)$.

Solution:

The language A is decided by the following one-tape Turing machine:

$M =$"On input w

1. Scan the input and *reject* if it does not contain exactly one $\#$.
2. Repeat while there are uncrossed-off symbols on both sides of the $\#$.
3. Scan the tape and cross off the first uncrossed-off symbol to the left of the $\#$ and the last uncrossed-off symbol to the right of the $\#$. *Reject* if the two symbols are not the same.
4. If all symbols except the $\#$ have been crossed-off, *accept*, else *reject.*"

Each scan of the tape takes time $O(n)$. There is one scan in Step 1 and one in Step 4. Each time through the loop in Steps 2 and 3, 2 symbols are crossed off, so there are no more than $\lfloor n/2 \rfloor$ scans in this loop. Thus the number of scans is $O(n)$ and the running time is $O(n)O(n) = O(n^2)$.

(b) Give an implementation-level description of a multi-tape Turing machine that decides A in time $O(n)$. (You do not need to explain why the running time is $O(n)$.)

Solution:

A is decided by the following two-tape Turing machine:

$N =$"On input x

1. Scan the input on tape 1 and *reject* if it does not contain exactly one $\#$.
2. Scan the input on tape 1 again and copy the symbols up to the $\#$ onto tape 2.
3. Continue reading symbols and moving right on tape 1. For each symbol after the $\#$ on tape 1, move left on tape 2 and *reject* if the symbols on the two tapes do not match.
4. When all symbols on tape 2 are read, *accept* if all symbols to the right of the $\#$ have been read on tape 1 and *reject* otherwise."

3. An undirected graph G is called *bipartite* if there is a set of vertices R such that every edge connects a vertex in R with a vertex not in R. Let

$$BIPARTITE = \{\langle G \rangle | G \text{ is a bipartite graph}\}$$

Show that BIPARTITE is in NP.

Solution:

BIPARTITE is decided by the following nondeterministic Turing machine:

$T =$"On input $\langle G \rangle$

1. Guess a set of vertices R.
2. For each edge (u, v) of G check that one of u and v is in R and the other isn't.
3. If all edges pass the test in Step 2, *accept*, else *reject*."

Each step is executed once and in Step 2, each edge is processed once, so T runs in polynomial time.

4. Show that the grammar given in Exercise 2.13 of the text is ambiguous.

Solution: The grammar is ambiguous because the string #0# has the following two leftmost derivations:

$$S \Rightarrow TT \Rightarrow T0T \Rightarrow \#0T \Rightarrow \#0\#$$

$$S \Rightarrow TT \Rightarrow \#T \Rightarrow \#0T \Rightarrow \#0\#$$

5. If A is language, then Problem 1.40 on page 89 of the text gives the definition of another language $NOPREFIX(A)$. Show that if A is decidable, then $NOPREFIX(A)$ is decidable, by completing the following proof.

Let M be a Turing machine that decides A. Then a Turing machine N that decides $NOPREFIX(A)$ is given by

N="On input w

1. Run the Turing machine M that decides A on w. If M rejects, *reject.*
2. Run M on all the proper prefixes of w. If M rejects all the proper prefixes, then *accept*, else *reject.*"

6. Which one of the following statements correctly describes the running time of the Turing machine M_2 given in Example 3.7? (No explanation for your answer is required.)

(a) ☐ M_2 runs in time $O(n^2)$ but not in time $O(n \log n)$.
(b) ☒ M_2 runs in time $O(n \log n)$ but not in time $O(n)$.
(c) ☐ M_2 runs in time $O(n)$.

7. Does the Turing machine R given in the proof of Theorem 4.8 run in polynomial time? Explain your answer.

Solution: Yes. Steps 1 and 4 are done once. The loop in Steps 2 and 3 is done no more times than the number of variables of G. Step 3 involves examining each rule once, so this step can be implemented in polynomial time on a Turing machine as can all the other steps.