NATIONAL UNIVERSITY OF SINGAPORE

CS 4232: Theory of Computation

Semester 1; AY 2019/2020; Midterm Test 1

Time Allowed: 40 Minutes

INSTRUCTIONS TO CANDIDATES

1. Please write your Student Number. Do not write your name.
2. This assessment paper consists of FOUR (4) questions and comprises NINE (9) printed pages.
3. Students are required to answer **ALL** questions.
4. Students should answer the questions in the space provided.
5. This is a **CLOSED BOOK** assessment.
6. Every question is worth FIVE (5) marks. The maximum possible marks are 20.

STUDENT NO: ______________________________

This portion is for examiner's use only

Question	Marks	Remarks
Question 1:		
Question 2:		
Question 3:		
Question 4:		
Total:		

Provide a deterministic finite automaton which recognises all ternary numbers without leading zeroes which are not multiples of five. Ternary numbers have the digits $0, 1, 2$ and examples of numbers to be accepted are $1, 2, 10, 11, 20, 21, 22, 100$ and numbers to be rejected are $0, 12, 101$ (zero, five, ten) and $00, 01, 0021$ (leading zeroes). The dfa should be complete and have for every state and symbol exactly one successor. Do not use more than eight states.

Solution. Solutions are not unique when making automata. One possible solution is given by the following table:

state q	$\delta(q, 0)$	$\delta(q, 1)$	$\delta(q, 2)$	type
s	r	t_1	t_2	rej, start
r	r	r	r	rej
t_0	t_0	t_1	t_2	rej
t_1	t_3	t_4	t_0	acc
t_2	t_1	t_2	t_3	acc
t_3	t_4	t_0	t_1	acc
t_4	t_2	t_3	t_4	acc

The automaton accepts 121 with the run s, t_1, t_0, t_1 and rejects 001 with the run s, r, r, r and 1212 with the run s, t_1, t_0, t_1, t_0.

Student Dagobert Hugoson decides to use the following pumping lemma: Given a regular language L, there is a constant k such that for all words $u \in L$ with $|u| \geq k$, one can find x, y, z with $u = xyz$ and $1 \leq |y| \leq k$ and $\{x\} \cdot \{y\}^* \cdot \{z\} \subseteq L$.

How does this pumping lemma of Dagobert Hugoson differ from the traditional pumping lemma for regular languages?

Show that the language $L = \{0^h 1^i 2^j : h \neq i \wedge h \neq j \wedge i \neq j\}$ satisfies the pumping lemma of Dagobert Hugoson and determine the best possible choice of k.

Show that the language L does not satisfy the traditional pumping lemma.

Solution. The difference to the traditional pumping lemma is that it has the additional condition $|xy| \leq k$. With this, $|y| \leq k$ is redundant and so one usually only postulates $y \neq \varepsilon$ what is equivalent to $|y| \geq 1$.

The value of k is four. The reason is that there is one word 122 (with $h = 0, i = 1, j = 2$) which cannot be pumped down, as the word $\varepsilon, 1, 2, 12$ are all not in L. Now assume that $0^h 1^i 2^j \in L$ and has at least length 4. Without loss of generality assume that $h < i < j$ and $j \geq 3$, as otherwise the word considered is 122 and has length 3. Now one chooses $\ell \in \{1, 2, 3\}$ such that $j - \ell$ differs from both h, i and let $z = \varepsilon$, $y = 2^\ell$ and x be the prefix of u of length $|u| - \ell$. Thus when omitting y, the numbers of $0, 1, 2$ in the word are all different. When repeating y several times, the number of twos grows while the other digits remain at the same quantity and therefore again the pumped up words are all in L. So L satisfies the pumping lemma of Dagobert Hugoson.

The normal pumping lemma is not satisfied. Assume that k is the pumping constant and consider the word $0^k 1^{k+1} 2^{k+k!}$. Then any pump has to be a sequence of zeroes and there are at most k of them; therefore when pumping up, one can pump them so often that exactly $k!$ zeroes are added and the resulting word has as many zeroes as twos, thus is not in L.

Consider the regular expression $(\{00\}\cdot\{000\}^*)\cup(\{000\}\cdot\{0000\}^*)\cup(\{00000\}\cdot\{000000\}^*)$. Construct an nfa recognising this language with as few states as possible, the nfa can have multiple start states. However, every transition in the automaton should use exactly one symbol.

Solution. The third part of the union is redundant, as it is a subset of the first part. So one can construct the following nfa, where only the choice of the start state is non-deterministic. A possible solution is given by the following table:

state	successor at 0	type
s	p	rej, start
p	q	rej
q	r	rej
r	s	acc
s'	p'	rej, start
p'	q'	rej
q'	s'	acc

A dfa for this set needs 12 states, as the language given by above regular expression might be written as $\{0^2, 0^3, 0^5, 0^7, 0^8, 0^{11}\}\cdot\{0^{12}\}^*$ and here the length five is the unique length n (modulo 12) such that there are words of length $n-3, n-2, n, n+2, n+3$ in the language; thus there are twelve different derivatives. The nfa can only save on states by separating out the two loops and then the number of states is the sum of the states needed for the loops, so it is seven.

Consider the language $H = \{0^n 1^{nm} 2^m : 1 \leq n, 1 \leq m \leq 3\}$ and determine the exact level of the Chomsky hierarchy of H:

- ☐ regular;
- ☐ context-free but not regular;
- ☐ context-sensitive but not context-free;
- ☐ recursively enumerable but not context-sensitive.

Provide a grammar for H on the chosen level and use as few non-terminal symbols as possible.

Solution. The language H is context-free but not regular.

A possible context-free grammar has the non-terminals $\{S, U, V, W\}$, the terminals $\{0, 1, 2\}$, the start symbol S and the following rules:

$$
\begin{aligned}
&S \to U2|V22|W222,\\
&U \to 0U1|01,\\
&V \to 0V11|011,\\
&W \to 0W111|0111.
\end{aligned}
$$

Note that there are always many ways on how to make a grammar.

END OF PAPER