NATIONAL UNIVERSITY OF SINGAPORE

CS 5236: Advanced Automata Theory

Semester 1; AY 2022/2023; Midterm Test

Time Allowed: 60 Minutes

INSTRUCTIONS TO CANDIDATES

1. Please write your Student Number. Do not write your name.
2. This assessment paper consists of FIVE (5) questions and comprises ELEVEN (11) 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 with one helpsheet.
6. You are not permitted to communicate with other people during the exam and you are not allowed to use additional material beyond the helpsheet.
7. Every question is worth SIX (6) marks. The maximum possible marks are 30.

STUDENT NO: ______________________________

This portion is for examiner's use only

Question	Marks	Remarks
Question 1:		
Question 2:		
Question 3:		
Question 4:		
Question 5:		
Total:		

Let $\Sigma = \{0, 1, 2, 3\}$. Consider the language H of all nonempty words which have as many 0 as 1 as 2 as 3. Construct a context-sensitive grammar for this language and provide the derivation of 01233210. Please also include a short verbal description of the grammar. Furthermore, show that this language does not satisfy the pumping lemma for context-free languages, thus that the language cannot be context-free.

For context-sensitive grammars, the following two constraints must be satisfied for each rule $l \to r$: The left side l contains at least one nonterminal and the right side is at least as long as the left side. Note that the language does not contain the empty word ε and therefore no exception rules to generate this word are needed.

The standard pumping lemma for context-free languages says that there is a constant k such that any given word in H of length k or more can be split into $uvwxy$ such that $vx \neq \varepsilon$, $|vwx| \leq k$ and for all $\ell \geq 0$ the word $uv^\ell wx^\ell y$ is in H. One has to show that this pumping lemma fails.

Solution. The language H is generated by the following grammar: Terminals $0, 1, 2, 3$, nonterminals S, T, U, V, W, start symbol S, rules $S \to STUVW|TUVW$ and all rules of the type $XY \to YX$ with $X, Y \in \{T, U, V, W\}$, that is, the rules $TU \to UT$, $TV \to VT$, $TW \to WT$, $UT \to TU$, $UV \to VU$, $UW \to WU$, $VT \to TV$, $VU \to UV$, $VW \to WV$, $WT \to TW$, $WU \to UW$, $WV \to VW$; the rules $T \to 0$, $U \to 1$, $V \to 2$, $W \to 3$. So the grammar generates an equal number of T, U, V, W and then allows these nonterminals to swap order arbitrarily. Afterwards the four nonterminals are converted into digits.

The sample derivation is $S \Rightarrow STUVW \Rightarrow TUVWTUVW \Rightarrow TUVWTUWV \Rightarrow TUVWTWUV \Rightarrow TUVWWTUV \Rightarrow TUVWWTVU \Rightarrow TUVWWVTU \Rightarrow TUVWWVUT \Rightarrow 0UVWWVUT \Rightarrow 01VWWVUT \Rightarrow 012WWVUT \Rightarrow 0123WVUT \Rightarrow 01233VUT \Rightarrow 012332UT \Rightarrow 0123321T \Rightarrow 01233210$.

Assume that the language satisfies the standard pumping lemma with constant k. Then on the word $0^k 1^k 2^k 3^k$, the two pumps can only cover two of the four digits and therefore the pumps v, x of the splitting $uvwxy$ will modify at least one and at most two digits when pumped up or down, thus for $\ell \neq 1$ the number of digits in the pumped word is no longer equal for all four digit types.

The most easy form of Jaffe's Pumping Lemma is this: A language I is regular iff there is a constant k such that for all words x of length at least k there is a splitting of x into u, v, w such that $|vw| \leq k$, $v \neq \varepsilon$ and the derivative $I_{uv^\ell w} = I_x$ for all $\ell \geq 0$. Here the derivative I_x is the language of all y with $xy \in I$.

The Theorem of Myhill and Nerode says that a language I is regular iff I has only finitely many derivatives, that is, there are finitely many words $w_1, \ldots, w_k$ such that for each word v, $I_v = I_u$ for some $u \in \{w_1, \ldots, w_k\}$. Both theorems characterise regular languages for finite alphabets.

(a) Prove that every language satisfying the pumping condition of Jaffe's Pumping Lemma also satisfies the criterion for regularity of the Theorem of Myhill and Nerode, that is, has only finitely many derivatives.

(b) If one would allow an infinite alphabet Σ, then does every language satisfying the pumping condition of Jaffe's Pumping Lemma also have only finitely many derivatives? Either prove that this is true or provide a counter example language.

Solution. (a) If x is a word longer than k then there is by Jaffe's Pumping Lemma a word uw shorter than x having the same derivative, that is, satisfying $I_{uw} = I_x$. Thus one looks, for each derivative I', at the shortest word x with $I_x = I'$. This word x has at most length $k-1$, as otherwise one can pump down. Now there are at most $1 + |\Sigma| + |\Sigma|^2 + \ldots + |\Sigma|^{k-1} \leq k \cdot |\Sigma|^k$ many different derivatives. Thus the criterion of the Theorem of Myhill and Nerode is satisfied for all languages satisfying the pumping condition of Jaffe's Pumping Lemma.

(b) If the alphabet Σ is infinite then the following language $I = \cup_{a \in \Sigma} \{a\}^*$ satisfies Jaffe's Pumping Lemma: If the word x has at least length 3 and two different letters, then one can select the pump v such that avoids the first occurrences of a, b in the word and thus $I_{uw} = I_x = I_{uv^\ell w} = \emptyset$ for all $\ell \geq 0$. If $w \in \{a\}^*$ for some single letter $a \in \Sigma$ and $|x| \geq 3$, then one choses u to be all of x but the last letter, $v = a$ and $w = \varepsilon$ and $I_{uw} = I_x = I_{uv^\ell w} = \{a\}^*$ for all $\ell \geq 0$. The last item also shows that the letter a uniquely defines the derivative and thus there are infinitely many derivatives for I, as Σ is infinite. So the implication proven for finite alphabets in (a) does not hold for infinite alphabets in (b).

Consider the context-free language $J \subseteq \{0,1,2\}^+$ which contains all nonempty words in which the number of zeroes is same as the sum of the number of ones and the number of twos.

So J contains the words 01, 02, 1002 and 1100, but it does not contain the words ε and 012.

For the derived languages in (a) and (b) below, either construct a context-free grammar or explain why such a grammar does not exist:

(a) $J \cap (\{0\}^+ \cdot \{1,2\}^+)$ and

(b) $J - (\{0,1,2\}^* \cdot \{11,2\} \cdot \{0,1,2\}^*)$.

Solution. Both grammars exist, the second one can even to be taken regular; note that regular grammars are also context-free.

(a) For the first grammar, one just makes sure that the $1,2$ are all at the back and the 0 at the front. The grammar needs only one nonterminal S and terminals $0,1,2$: $S \to 0S1|0S2|01|02$.

(b) For the second grammar, one has the property that there is no 2 and that between any two 1 there is a 0. So there are three possibilities: First $(01)^+$, second $(10)^+$, third $(10)^+ \cdot (01)^+$. Thus the language is even regular and one can use a regular grammar. The grammar has two nonterminals S,T and the following rules: $S \to 10S|01T|10|01$, $T \to 01T|01$.

Consider the following game: Two players play. In a binary string each one can do either of the following two replacements: Replace a subword 1 by 0 or replace a subword 10 by 01. The player which makes the last bit 0 wins. Anke starts to move and the players move alternately.

Which player wins the game starting with the string 11011? Make a choice:

☐ Anke ☐ Boris

It can be shown that the same player wins all games which are given by a starting word from the regular language $K = ((\{1\} \cdot \{00\}^* \cdot \{1\}) \cup \{0\})^+$. Describe a winning strategy for this player for the words in K.

As an example, consider that the start word is 1100. The possible moves for Anke are 0100, 1000, 1010. Then, on 1010, the possible moves for Boris are 0010, 1000, 0110, 1001.

Solution. ☒ Boris: Boris wins the game on 11011 and for all words in K.

For general starting words in K, note that they consider of blocks which are either 0 or $10^{2k}1$. The two ones in the latter blocks are called pair. Now it is described how, for each move of Anke, Boris reacts to get the game back into K; note that Anke selects a pair of the form $10^{2k}1$ and then does one of the following three move-types with this pair:

- Anke changes one 1 of the pair to 0: Boris does the same with the other 1.
- Anke moves the first 1 of the pair to the back (this requires that the distance between the two 1s of the pair is at least 2): Boris also moves the first 1 of the pair again so that the distance is restored to be even.
- Anke moves the second 1 of the pair (this requires that the next block is 0): Boris moves the first 1 of the pair in order to restore the old distance.

Thus, after each pair of moves by Anke and then by Boris, the word is again in the set K. Furthermore, note that when one interprets the words as binary numbers, then every move reduces the value of the corresponding binary number. Thus the game will have only finitely many moves and as Boris can always answer to Anke's move, the last move in the game is by Boris and will result in the all-0 word, as otherwise Anke could make a move by changing a 1 to 0. So Boris wins the game, if Anke has to start with a word in K. Note that there are also other situations where Boris can win, for example on the word 10101. One can show that Boris wins the game iff the binary value of the start number is a multiple of 3.

Construct a deterministic Büchi automaton recognising the ω-language $L = \{0\}^* \cdot \{01, 011\}^\omega$ of infinite words.

Here the automaton should be deterministic and complete, that is, for each combination of state and symbol there is exactly one successor. Furthermore an ω-word is in L iff the automaton goes infinitely often through an accepting state when processing the ω-word.

Solution. Note that L can also be written as $\{0\}^+ \cdot \{10, 110\}^\omega$. The deterministic Büchi automaton M for L has states s, t, u, v, w, x. The transitions are as following: On 0, M goes from s, t to t, from u, v to w and from w, x to x. On 1, M goes from t, w to u, from u to v and from s, v, x to x. The state s is the starting and rejecting state, u, v, w are accepting states and t, x are rejecting states.

END OF QUESTION PAPER.