NATIONAL UNIVERSITY OF SINGAPORE

CS 5236: Advanced Automata Theory

Semester 1; AY 2020/2021; 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 an **OPEN BOOK** assessment.
6. You are not permitted to communicate with other people during the exam and you are not allowed to post in forums or other such entries during the exam.
7. Every question is worth SIX (6) marks. The maximum possible marks are 30.

STUDENT NO: _____

Question 1 [6 marks] **CS 5236 – Solutions**

Let $\Sigma = \{0, 1\}$. Consider the following nfa: The set $Q = \{q_1, q_2, \ldots, q_n\}$. q_n is the accepting state and q_1 is the starting state. For $m < n$, the nfa can on input 0 or input 1 go from q_m to q_{m+1}; furthermore, it can on input 1 go from any q_m to q_1. Determine how many states a complete dfa needs to recognise the same language as the given nfa and prove the result.

Solution. Let L be the language recognised by the nfa. Let $Q(x)$ be the states of the nfa which can be reached on input word x. Note that $0^{n-m} \in L_x$ iff $q_m \in Q(x)$. Now it will be shown that every subset of $\{q_1, q_2, \ldots, q_n\}$ can be a possible set $Q(x)$. Thus a complete dfa will need 2^n states to recognise the language L. Assume that $Q(x) \subseteq \{q_1, q_2, \ldots, q_n\}$ and that $Q(x)$ is not empty. Now $Q(x0) = \{q_{k+1} : q_k \in Q(x) \wedge k < n\}$ and $Q(x1) = Q(x0) \cup \{q_1\}$. Furthermore, $Q(\varepsilon) = \{q_1\}$, as that is the start state. Now for any word x of length $m < n$, the set $Q(x) = \{q_{m+1}\} \cup \{q_k : x_k = 1\}$ where $x = x_m x_{m-1} \ldots x_2 x_1$ as a bit-sequence. Furthermore, $Q(0^n) = \emptyset$. Thus one can see that every subset $P \subseteq \{q_1, q_2, \ldots, q_n\}$ equals some $Q(x)$ for some binary word x.

Question 2 [6 marks] **CS 5236 – Solutions**

In the following, Anke, Boris, Claude and Doris have their versions of the block pumping lemma. Let L be any language and k be a constant. Let $u_0, u_1, \ldots, u_{k-1}, u_k$ be strings ("blocks") with $u_0 u_1 \ldots u_k \in L$.

Anke: If $u_1, u_2, \ldots, u_{k-1}$ are nonempty then there must be i, j with $1 \leq i < j \leq k$ and

$$u_0 \ldots u_{i-1} \cdot (u_i \ldots u_{j-1})^* u_j \ldots u_k \subseteq L.$$

Boris: If $u_0, u_1, u_2, \ldots, u_{k-1}, u_k$ are nonempty then there must be i, j with $1 \leq i < j \leq k$ and

$$u_0 \ldots u_{i-1} \cdot (u_i \ldots u_{j-1})^* u_j \ldots u_k \subseteq L.$$

Claude: There must be i, j with $1 \leq i < j \leq k$ and

$$u_0 \ldots u_{i-1} \cdot (u_i \ldots u_{j-1})^* u_j \ldots u_k \subseteq L.$$

Doris: There must be i, j with $0 \leq i \leq j \leq k$ and

$$u_0 \ldots u_{i-1} \cdot (u_i \ldots u_j)^* u_{j+1} \ldots u_k \subseteq L.$$

In other words, Doris allows to include the first block u_0 and last block u_k into the pump and does not require any string to be nonempty.

Anke provides the official definition of block pumpable. For those of Boris, Claude and Doris, check whether these pumping definitions (a) apply to the same languages with the same corresponding constant, (b) apply to the same languages but with different constants or (c) do not apply to the same languages. Give reasons for the answer.

For example, there are languages which satisfy the context-free pumping lemma but not the block pumping lemma, hence the context-free pumping lemma applies to other languages than the block pumping lemma.

Solution. Here the three notions compared with Anke's notion.

Boris requires all the blocks to be nonempty. This applies to the same languages but with different constants. For example, the regular language $\{1\} \cdot \{0\}^* \cdot \{1\}$ needs for Anke's version of the block pumping lemma constant $k = 4$, that is, it needs three innner blocks out of which the middle one then can be pumped; for Boris' version, the constant $k = 2$ is enough, as the nonemptyness of the border blocks requires that the 1 is in them, so that the zeroes in the middle can be pumped.

Claude's version is equivalent to the standard one. If some inner block is empty, then this block can be pumped, as the only requirement for pumping is that some neighbouring inner blocks form the pump, but not that this pump is nonempty (otherwise no language would be block pumpable, as all inner blocks could be empty). Thus the pumping is only required to be real in the case that all inner blocks are nonempty and then the condition is equivalent to Anke's. So both conditions can be translated into each other and apply to the same languages with the same constants.

Some languages of the form L^* are not block pumpable. For example the language $L = \{10^{n^2} : n \in \mathbb{N}\}$. Then L^* contains a word 10^m iff m is a square number. Now taking a large word of the form 10^{n^2} and choosing blocks only in the part with the zeroes, pumping up would destroy the property that the number of zeroes is a square and therefore the pumped up words would witness that L^* is not block pumpable. However, every language of this type satisfies Doris' version of the block pumping lemma, as that allows to pump the whole word, irrespectively of the ways the border are taken. Thus Doris' version and Anke's version do not apply to the same languages.

Question 3 [6 marks] **CS 5236 – Solutions**

Provide a context-sensitve grammar for the language $L = \{10^{n^2} : n \geq 1\}$. The grammar can use all rules of the form $l \to r$ which (a) contain at least one nonterminal on the left side l and (b) satisfy that r is at least as long as l. Explain how 1000000000 is derived.

Solution. The grammar uses these rules: $S \to 10|10000|1TUVW$, $V \to UW|UVW$, $TW \to T0$, $UW \to W0U$, $0W \to W0$, $1T \to 1R$, $R0 \to 0R$, $RU \to 0R|00$. Furthermore, the terminals are $0, 1$, the nonterminals are R, S, T, U, V, W and the start symbol is S.

The sample derivation is as follows. At the beginning, S is converted to $1TUVW$ and then the V to UW, as only two U and two W are needed: $S \Rightarrow 1TUVW \Rightarrow 1TUUWW$.

Now one moves the W to the front over the U where jumping over an U generates a 0: $1TUUWW \Rightarrow 1TUW0UW \Rightarrow 1TW0U0UW \Rightarrow 1TW0U0W0U \Rightarrow 1TW0UW00U \Rightarrow 1TW0W0U00U \Rightarrow 1TWW00U00U$.

Now the W will be converted to 0 and the more back W jump over these 0 to reach T: $1TWW00U00U \Rightarrow 1T0W00U00U \Rightarrow 1TW000U00U \Rightarrow 1T0000U00U$.

Now T becomes an R and R moves to the back until reaching an U. $1T0000U00U \Rightarrow 1R0000U00U \Rightarrow 10R000U00U \Rightarrow 100R00U00U \Rightarrow 1000R0U00U \Rightarrow 10000RU00U$.

When meeting the first U, it will be converted to a 0 with the R preserved, but when meeting the second and last U, both R and U will be converted to a 0: $10000RU00U \Rightarrow 100000R00U \Rightarrow 1000000R0U \Rightarrow 10000000RU \Rightarrow 1000000000$.

Question 4 [6 marks] **CS 5236 – Solutions**

Consider the following Büchi-game: There are n decimal digits $a_{n-1}a_{n-2}\dots a_2a_1a_0$ defining the natural number $d = \sum_k 10^k \cdot a_k$. Furthermore, let m be a natural number. Anke modifies a_0 in move 0 and then Boris modifies a_1 in move 1 and so on so that for even t, Anke modifies a_k with k being the remainder of t by n and for odd t, Boris modifies the corresponding digit. Now Anke wins the game iff there are infinite many t so that the value of d_t after move t is a multiple of m.

Assume that $n = 3$. For each $m \in \{4, 11, 40, 50, 101, 125\}$ say whether Anke or Boris or nobody has a winning strategy; either sketch the winning strategy or say why none of the players has one.

Solution. Anke has winning strategies for $4, 11, 50, 101$ and Boris has for winning strategies for $40, 125$.

Winning strategy for Anke, $m = 4, n = 3$: Whenever it is Anke's turn to choose a_0 (that is the case when t is a multiple of 6) then she checks the current value of a_1. If that value is even she takes $a_0 = 0$ (as all numbers ending with $00, 20, 40, 60, 80$ are multiples of 4), if that value is odd, she takes $a_0 = 2$ (as all numbers ending with $12, 32, 52, 72, 92$ are multiples of 4).

Winning strategy for Anke, $m = 11, n = 3$: Whenever it is Anke's turn to choose a_0, she chooses 0. Then it is Boris turn to choose a_1. Afterwards Anke chooses $a_2 = a_1$ resulting in a decimal number of the form $bb0$ which is a multiple of 11. This happens every sixth move and so infinitely often, hence Anke wins the game.

Winning strategy for Boris, $m = 40, n = 3$: When $t = 6s + 3$ then Boris puts $a_0 = 1$. Then Anke puts a_1 to be an even number (otherwise she would destroy herself). If that is $2, 6$ then Boris puts $a_2 = 0$ as $20, 60$ are are not multiples of 40 else Boris puts $a_2 = 1$ as $100, 140, 180$ are not multiples of 40. Then Anke puts $a_0 = 0$. Then Boris puts $a_1 = 1$ and Anke puts a_2 to be something. Now comes move $t = 6(s + 1) + 3$ and above cycle repeats. Thus during the whole cycle, no multiple of 40 is created and from $t = 3$ onwards, d_t is not a multiple of 40. Hence Boris wins the game.

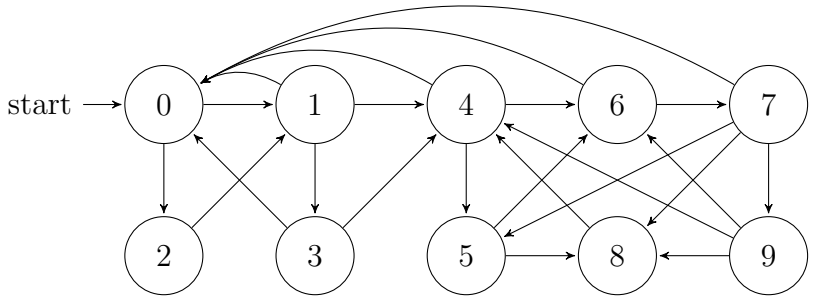
Winning strategy for Anke, $m = 50, n = 3$: Whenever Anke has to choose a_1, she chooses $a_1 = 5$. Then Boris chooses $a_2 = b$ for some b and Anke chooses $a_0 = 0$. Now the number is $b50$ and each number of that form is a multiple of 50. Hence this method gives a winning strategy for Anke. This method really requires that n is odd; if n is even and $m = 50$, then Boris has a winning strategy.

Winning strategy for Anke, $m = 101, n = 3$: When Anke chooses a_1, then she chooses 0. Now Boris chooses for a_2 some value b. Afterwards Anke chooses for a_0 also the value b so that the result is $b0b$. This value is $b \cdot 101$. Note that $b = 0$ is here explictly allowed. So Anke has a winning strategy.

Winning strategy for Boris, $m = 125, n = 3$: Boris just puts the digit 4 whenever he can choose a digit. Thus after move $t = 1$, every value d_t contains a digit 4. However, the multiples of 125 below 1000 are $0, 125, 250, 375, 500, 625, 750, 875$ and none of these has the digit 4. Thus Boris wins the game.

Question 5 [6 marks] **CS 5236 – Solutions**

Consider the following parity game.



Recall that in a parity game a play is an infinite sequence of alternate moves by players Anke and Boris, each move is given by the number on the target node. Let b be the largest number which occurs infinitely often in the play. If b is even then Anke wins the play else Boris wins the play. Anke makes the first move.

Determine which player has a winning strategy and table up that winning strategy, note that the winning strategy is memoryless, so one has only to say for each node where to move.

Solution. Player Anke has a winning strategy for this parity game. Note that Anke will not move from 6 to 7 and not move from 7 to 9; as 9 can only be reached by the sequence 6 - 7 - 9, it means that the play never goes there (except when it starts in 9 and is there exactly once). All moves a - b with a different from 9 by Anke's table satisfy that the maximum of a and b is even, thus if one has an infinite play and splits it in pairs of nodes with first component Anke has to move and second component, Boris has to move, then all these pairs will have an even maximum and thus the overall supremum of the play will be even. The table of Anke's winning strategy is as follows:

Node:	0	1	2	3	4	5	6	7	8	9
Move:	2	4	1	4	0	8	0	8	4	4