NATIONAL UNIVERSITY OF SINGAPORE

CS 5236: Advanced Automata Theory

Semester 1; AY 2018/2019; Midterm Test

Time Allowed: 50 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.
6. It is permitted to use calculators, provided that all memory and programs are erased prior to the assessment; no other material or devices are permitted.
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:		

Provide a context-free grammar for the following language:

$$\{vw : |v| = |w| \wedge v \in \{0,1\}^+ \cdot \{2\}^+ \wedge w \in \{3\}^+ \cdot \{0,1\}^+\}.$$

Solution. Solutions are not unique when making grammars. One possible solution is the following one:

1. Non-terminals $\{S, T, U, V, W\}$,
2. Terminals $\{0, 1, 2, 3\}$,
3. Start symbol S,
4. Rules $S \rightarrow TST|TUT|TVT|TWT$, $T \rightarrow 0|1$, $U \rightarrow 2U3|23$, $V \rightarrow 2VT|2UT$, $W \rightarrow TW3|TU3$.

The idea of the grammar is that it builds a word in the language by making first the two outmost symbols and then more and more symbols, each one for the front and one for the end, until the final 23 in the centre is made.

Here T is a non-terminal to make either 0 or 1. U is a non-terminal which make the final 23 and, on the way to this, for arbitrary rounds, a 2 before and a 3 behind the U. The V is a special symbol for the case that there are more 2 than 3 and it will make a 2 before the V while a T behind the V which will be transformed into 0 or 1. W does the converse, making a T before the W while making a 3 behind it. S generates first the same amound ot T at the beginning and the end of the word and then makes one of U, V, W being between T.

Consider the following versions (a), (b) and (c) of the pumping lemma:

For a language L there is a constant ℓ such that all words $u \in L$ which are longer than ℓ can be represented as $u = xyz$ such that the following conditions hold:

1. $y \neq \varepsilon$;
2. $\{x\} \cdot \{y\}^* \cdot \{z\} \subseteq L$;
3. Depending on its version, the following length-constraint holds:
   (a) $|xy| \leq \ell$ (traditional pumping lemma);
   (b) $|y| \leq \ell$;
   (c) $|x| + |z| \leq \ell$.

Note that version (b) is less restrictive than version (a), thus every regular language satisfies (b).

Is version (c) satisfied by every regular language? ☐ Yes, ☐ No.
If the answer is "yes" then provide a proof else provide a counterexample.

Let $H = \{u \in \{0,1,2\}^*$: the number i of all 0 in u and the number j of all 1 in u and the number k of all 2 in u satisfy that either one or all three of the conditions $i = j$, $i = k$ and $j = k$ hold$\}$.

Which of the versions ☐ (a) ☐ (b) ☐ (c) of the pumping lemma above does the language H satisfy (tick the corresponding boxes)? Give reasons for the answer.

Solution. The first answer is "no", a counterexample is $\{0\}^* \cdot \{1\} \cdot \{0\}^*$. Let ℓ be the pumping constant and consider the word $u = 0^\ell 1 0^\ell$ and consider a splitting into xyz according to (c). Now 1 is inside the y due to the length constraints and therefore $xyyz \notin \{0\}^* \cdot \{1\} \cdot \{0\}^*$.

The language H satisfies the pumping lemmas (b) and (c), but not (a). Here the reasons. Let ℓ again be the pumping constant, without loss of generality, $\ell > 3$. For (b) and (c), let u be a word in H of length ℓ or more and let i be the number of 0, j be the number of 1 and k be the number of 2 in u.

The counter example for (a) is $0^\ell 1^\ell 2$. Now y must be a nonempty string of 0s and therefore $xyyz$ has all cardinalities i, j, k be different and so $xyyz \notin H$.

For (b), as $u \in H$, at least one of the equalities is true, say $i = j$. Now $i = k$ iff $j = k$, so $i = j$ is enough to imply that $u \in H$. Thus pumping a subword with as many 0 as 1 keeps the word inside H. If there is a 2 in u then one let $y = 2$ and chooses x, z accordingly. The numbers i, j are for all $xy^h z$ the same and so $xy^h z \in H$. If there is no subword 2 in u then either 01 or 10 is a subword of u, one chooses y to be this subword and x, z accordingly. Now $xy^h z$ has $i + h - 1$ 0s and $j + h - 1$ 1s, as $i = j$ the word $xy^h z$ is in H.

To see (c), note that one can take $y = u$, $x = \varepsilon$ and $z = \varepsilon$. Now consider $xy^h z = u^h$. For u^0, all digits occur zero times and the word is in H. For u^h with $h > 0$, the digits occur ih, jh, kh times, respectively, and as $i = j$, also either one or three of the equations $ih = jh, ih = kh, jh = kh$ are true. So the pumped words are in H.

A game has the nodes $\{000, 001, 010, 011, 100, 101, 110, 111\}$ and 000 and 111 are the target nodes. The following moves are possible:

$001 \to 010, 011;$
$010 \to 100, 101;$
$011 \to 110, 111;$
$100 \to 000, 001;$
$101 \to 010, 011;$
$110 \to 100, 101.$

In words, given a three-bit state abc which is not a target, the player selects a bit d and moves to bcd. The first player who reaches either of 000 and 111 wins.

Which are winning nodes and which are draw nodes and which are losing nodes for Anke? That is, when does Anke win, draw or lose when she starts moving from the corresponding node and both players move as good as possible.

Furthermore, write Anke's best move at the entry "Best Move", for losing nodes that entry can be left blank.

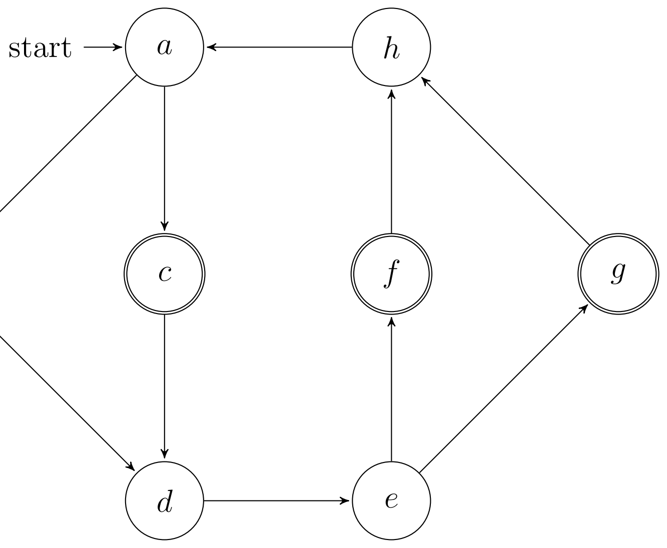
Node	Win/Draw/Loss	Best Move
001		
010		
011		
100		
101		
110		

Solution. The solution is given by the following table.

Node	Win/Draw/Loss	Best Move
001	Draw	010
010	Draw	101
011	Win	111
100	Win	000
101	Draw	010
110	Draw	101

Question 4 [6 marks]

The following game is an update game.



Recall that in an update game, player Anke wins a play iff it goes through all the final nodes (b, c, f, g) infinitely often and player Boris wins a play if at least one of the final nodes is visited only finitely often. Anke moves first.

Determine which player has a winning strategy for the game and describe how this winning strategy works. Explain why the player succeeds with the winning strategy.

Solution. One can see that for every node, there is only one player to move all the times: Anke moves on a, d, f, g and Boris moves on b, c, e, h. Boris has only a choice when the game is in e. He can then always move to f so that the game goes never into g and this strategy is a memoryless winning strategy for Boris for this update game.

Provide a complete and deterministic Büchi automaton recognising the following language of ω-words over the alphabet $\{0,1,2\}$:

$$\{0\}^* \cdot \{1\} \cdot \{0\}^\omega \;\cup\; \{0,1\}^* \cdot \{2\} \cdot (\{0,1\}^* \cdot \{2\} \cdot \{0,1\}^* \cdot \{2\})^\omega.$$

Use at most five states.

Solution. The Büchi automaton can be done with four states. Here the successor-table of the states.

state	0	1	2	type
s	s	u	w	start, rejecting
u	u	v	w	accepting
v	v	v	w	rejecting
w	v	v	w	accepting

For the explanation, note that the Büchi automaton accepts those ω-words which either contain one 1 and no 2 or contain infinitely many 2. States u is for accepting those ω-words which contain one 1 and no 2 and state w is for accepting those ω-words which contain infinitely many 2. On each 2, the Büchi automaton goes to state w and therefore the state w is infinitely often visited iff there are infinitely many 2 in the word. Furthermore, the Büchi automaton goes to state u iff it reads the first 1 after finitely many 0 and no other symbol and it stays in u iff subsequently only 0 are following. Once it leaves u, it never returns there.