NATIONAL UNIVERSITY OF SINGAPORE

CS 5236 – Advanced Automata Theory
(Semester 1: AY 2016/2017)

Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This assessment paper consists of TEN (10) questions and comprises ELEVEN (11) printed pages plus ONE (1) empty page.
2. Students are required to answer **ALL** questions.
3. This is a **CLOSED BOOK** examination.
4. 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.
5. Every question is worth SIX (6) marks. The maximum possible marks are 60.
6. Please write your Student Number below (including all digits and letters). Do not write your name.

MATRICULATION NO: ______________________________

This portion is for examiner's use only

Question	Marks	Remarks	Question	Marks	Remarks
Q01:			Q06:		
Q02:			Q07:		
Q03:			Q08:		
Q04:			Q09:		
Q05:			Q10:		
			Total:		

Construct a **context-sensitive grammar** for the language $L = \{00w00w00w : w \in \{0,1\}^*\}$. For example, $000000000000 \in L$, $001001001 \in L$ and $010101 \notin L$. Any grammar is allowed which satisfies $|l| \leq |r|$ and $l \in (N \cup \Sigma)^* - \Sigma^*$ for every rule $l \to r$.

Solution. The non-terminals are S, U, V, W, X, Y, Z, the terminals are $0, 1, 2$ and the start symbol is S. The rules are the following: $S \to UVW$, $U \to U0X|U1Y|00Z$, $X0 \to 0X$, $X1 \to 1X$, $XV \to V0X$, $XW \to W0$, $Y0 \to 0Y$, $Y1 \to 1Y$, $YV \to V1Y$, $YW \to W1$, $Z0 \to 0Z$, $Z1 \to 1Z$, $ZV \to 00Z$, $ZW \to 00$.

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

Consider the set $L = \{0^i 1^j 0^k : i, j, k \in \mathbb{N} \wedge j \leq i + k \leq 2j\}$.

What is the level of L in the Chomsky hierarchy?

☐ regular, ☐ context-free and not regular,
☐ context-sensitive and not context-free,
☐ recursively enumerable and not context-sensitive.

Give a justification for your choice, by providing the corresponding grammars or automata to show that the language is in that level and by applying the pumping lemmas or similar methods to show that it is in no better level (if applicable).

Solution. The language is context-free. The grammar is $(\{S, T, U\}, \{0, 1\}, P, S)$ with P containing the rules $S \to TU$, $T \to 0T1|00T1|\varepsilon$ and $U \to 1U0|1U00|\varepsilon$. However, the language is not regular. As it is infinite, one can apply the pumping lemma and there is a word uvw with $uv^*w \subseteq L$ and $v \neq \varepsilon$. As there are only two alternations between 0 and 1, the word v must from $\{0\}^+ \cup \{1\}^+$. When pumping up, the number of occurrences of one of the digits is increased in an arbitrary amount while the number of the occurrences of the other digit remains constant. This contradicts the condition that the zeroes occur at least as often as a ones and at most twice often as the ones.

A well-known result says that every context-free subset of $\{0\}^*$ is regular. One tries to generalise this statement by considering subsets of Σ^* for arbitrary alphabet Σ, but with the side condition that there is a constant c such that for each length n, $\Sigma^n \cap L$ has at most c members. Note that subsets of $\{0\}^*$ satisfy this condition with $c = 1$. Thus one considers the **following generalisation**:

($*$) If a context-free language $L \subseteq \Sigma^*$ satisfies $\exists c \forall n \, [|\Sigma^n \cap L| \leq c]$ then L is regular.

Is the **statement** ($*$) **true?** ☐ Yes, ☐ No.

Prove your answer.

Solution. The answer is "no". The reason is that the language $\{0^m 1 0^m : m \in \mathbb{N}\}$ has for each n at most one word of length n, namely when $n = 2m + 1$ the word $0^m 1 0^m$ and when $n = 2m$ it does not contain any word. The language is context-free by a grammar with non-terminal and start symbol S and rules $S \rightarrow 0S0|1$. One can see that L is not regular, as each derivative $L_{0^n 1}$ is $\{0^n\}$ and so the language has infinitely many different derivatives and is not regular by the Theorem of Myhill and Nerode.

Consider the following game: Anke and Boris start with the number 99999999 and both players alternately reduce one digit by 1 or by 2, but not by more. The digits are decimal digits from $\{0, 1, 2, \ldots, 9\}$ and a digit 1 can only be reduced to 0 and a digit 0 cannot be modified. That player who makes the last digit 0 wins the game, Anke moves first. **Which player** has a winning strategy for this game?

Some example moves in the game are 99999999 $\rightarrow$ 99999799 $\rightarrow$ 99999789 $\rightarrow$ 99999788 $\rightarrow$ 99999786.

Solution. Boris has a winning strategy for the game. His strategy is to keep all digits to be a multiple of 3. When Anke reduces one digit by $a \in \{1, 2\}$, this digit is no longer a multiple of 3 and Boris restores it to be a multiple of 3 by subtracting $3 - a$ from the digit. This is always possible, as the digit is at least $3 - a$ when it is not a multiple of 3. So eventually all digits will become 0 and Boris is always the player who converts a digit from 1 or 2 to 0. Thus he wins the game.

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

In a Büchi game, both players move alternately with player Anke starting at start node s. Player Anke wins an infinite play iff the play goes infinitely often through nodes in the set W of accepting nodes.

Let $V = \{0, 1, \ldots, 31\}$, $E = \{(x, y) : \text{if } x \leq 15 \text{ then } y \in \{2x, 2x+1\} \text{ else } y \in \{2x-32, 2x-31\}\}$, $s = 0$ and $W = \{0, 1, 2, 3\}$. **Which player** has a winning strategy for the **Büchi game** (V, E, s, W)? Can this winning strategy be chosen to be **memoryless**?

Solution. Boris has a winning strategy. One could interpret the numbers in the game as five-bit numbers. Each move shifts the digits by one bit to the front, discards the first bit and choses a new last bit. Anke wins if the game goes infinitely often through a number of the form $000xy$. However, Boris can avoid this by always putting the bit 1. Then, from the fifth move onwards, each number in the game is either of the form $1x1y1$ or $x1y1z$ and therefore the numbers will never again have the first three bits to be 000. Note that this, in the numerical setting, coincides with Boris moving to $2x + 1$ in the case that $x \leq 15$ and moving to $2x - 31$ in the case that $x \geq 16$; thus Boris winning strategy is also memoryless.

A deterministic Rabin automaton $(Q, \Sigma, \delta, s, \{(E_1, F_1), \ldots, (E_n, F_n)\})$ has states Q, alphabet Σ, start state s, a deterministic transition function $\delta : Q \times \Sigma \to Q$. The automaton accepts an ω-word $b_0b_1b_2 \ldots \in \Sigma^\omega$ iff the run $q_0q_1q_2 \ldots$ of the Rabin automaton on the word $b_0b_1b_2 \ldots$ satisfies that there is an $m \in \{1, 2, \ldots, n\}$ for which there are infinitely many s with $q_s \in E_m$ and only finitely many s with $q_s \in F_m$.

(a) **Give the formula** which determines the run $q_0q_1q_2 \ldots$ from the ω-word $b_0b_1b_2 \ldots$ on the input.

(b) Let $\Sigma = \{0, 1, 2, \ldots, 9\}$. Construct a **deterministic Rabin automaton** which accepts an ω-word $b_0b_1b_2 \ldots$ iff the maximal digit d which occurs infinitely often in the ω-word is even.

Solution. (a) The formula is $q_0 = s \wedge \forall m\, [q_{m+1} = \delta(q_m, b_m)]$.

(b) The alphabet Σ is given as $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. The Rabin automaton uses $Q = \Sigma$, so one denotes every state also by a symbol. Furthermore, 0 is the start state. The update function is $\delta(q, b) = b$, so the successor state of any state is the currently read symbol. Now there are five pairs (E, F) in the Rabin condition, for each even number d one takes the pair $(\{d\}, \{d + 1, d + 2, \ldots, 9\})$. More explicitly, the pairs are $(\{0\}, \{1, 2, 3, 4, 5, 6, 7, 8, 9\})$, $(\{2\}, \{3, 4, 5, 6, 7, 8, 9\})$, $(\{4\}, \{5, 6, 7, 8, 9\})$, $(\{6\}, \{7, 8, 9\})$ and $(\{8\}, \{9\})$. Now the Rabin automaton accepts an ω-word iff for one of these pairs (E, F) one state in E and no state in F occurs infinitely often in the run of the automaton. Thus there is an even number d for which d appears infinitely often as a state in the run but no digit larger than d appears infinitely often in the run. As δ copies, with one step delay, the ω-word $b_0b_1b_2 \ldots$ just into a sequence of states of the same name, the run of the Rabin automaton on this input is $0b_0b_1b_2 \ldots$ and if the largest d in the run is even, then the condition $(\{d\}, \{d+1, d+2, \ldots, 9\})$ is satisfied, as the element d of $\{d\}$ appears infinitely often as a state in the run while no element of $\{d + 1, d + 2, \ldots, 9\}$ appears infinitely often.

Question 7 [6 marks] **CS 5236 – Solutions**

Let $L = \{0,1\}^* \cdot \{01, 011\}$. Construct the **minimal deterministic automaton** for L. Note that this dfa has 4 states. Furthermore, describe the **syntactic monoid** of the minimal dfa: Make a table which contains, for each member of the monoid, a word w and how the function f_w defined by the word maps the states to new states; recall that $f_w(q) = \delta(q, w)$ for all words w.

Solution. The automaton consists of the states s, z, o, p and is given by the following transition table:

state	start	acc/rej	successor at 0	successor at 1
s	yes	reject	z	s
z	no	reject	z	o
o	no	accept	z	p
p	no	accept	z	s

The function $f_w(q)$ is $\delta(q, w)$. As 0 maps every state to z, the equality $f_{v0w} = f_{0w}$ is true for all v, w. Furthermore, $f_{0111} = f_{111}$. So one only needs to consider the mappings f_ε, f_0, f_{01}, f_{011}, f_1, f_{11}, f_{111}. They are given by the following table:

function	s	z	o	p
f_ε	s	z	o	p
f_0	z	z	z	z
f_{01}	o	o	o	o
f_{011}	p	p	p	p
f_1	s	o	p	s
f_{11}	s	p	s	s
f_{111}	s	s	s	s

Assume that an automatic representation of $(\mathbb{N}, +)$ is given. Now represent the set $\mathbb{Z}$ as $\mathbb{Z} = \{conv(a, b) : a, b \in \mathbb{N}$ and $a = 0$ or $b = 0\}$ where $conv(a, b)$ represents $a - b$. **Give formulas** which define when $conv(a, b) + conv(a', b') = conv(a'', b'')$, when $conv(a, b) < conv(a', b')$ and when $3 \cdot conv(a, b) = conv(a', b')$? These formulas should use $+$ and $=$ and quantification over members of $\mathbb{N}$. **Are** the so defined functions and relations **automatic**?

Solution. The conditions are the following: $conv(a, b) + conv(a', b') = conv(a'', b'')$ iff $a + a' + b'' = b + b' + a''$; $conv(a, b) < conv(a', b')$ iff $\exists c \in \mathbb{N}\, [a + b' + c + 1 = a' + b]$; $3 \cdot conv(a, b) = conv(a', b')$ iff $a' = a + a + a$ and $b' = b + b + b$. All three formulas are first-order defined from automatic parameters; these formulas use in one case quantification over members of $\mathbb{N}$ and are in two cases quantifier-free. Thus, by a theorem of Khoussainov and Nerode, the so defined functions and relations are automatic.

A Moore machine is a finite automaton where each state contains a word which is output whenever the automaton visits this state; the final output is the concatenation of all these output words, provided that the Moore machine is in an accepting state after processing the whole input. For a non-deterministic Moore machine with input v, all runs leading to an accepting state must provide the same output w. The function computed can be partial and is only defined on those inputs v for which there is an accepting run producing some output.

Construct a Moore machine which does the following: If an input v is of even length then the output is $0^{|v|}$ else the output is $1^{|v|}$. **If** the Moore machine can be made deterministic **then** give a **deterministic machine else** write a short **justification** why it cannot be made deterministic.

Solution. The Moore machine is non-deterministic, since there is no way that the Moore machine can anticipate whether the length of the input is even or odd; instead it must start producing an output symbol for each input symbol, as it cannot store the number of symbols until it has reached the end of the output. The Moore machine has the following transition table, where the choices of the transitions are the same for all symbols $a \in \Sigma$ and only depend on the state.

state	start	output	acc/rej	successor at a
s	yes	ε	accept	ev, od
ev	no	0	reject	ev'
ev'	no	0	accept	ev
od	no	1	accept	od'
od'	no	1	reject	od

Angluin's algorithm learns a dfa with n states with polynomially many queries in m, n where m is the length of the longest counter example given and n is the number of states of the dfa. Her algorithm makes use of the fact that the teacher, whenever it returns the answer "no" to an equivalence query, also provides a counter example where the conjectured language and the language to be learnt differ. **Is the same result also possible when all queries (membership queries and equivalence queries) are only answered by the teacher with "yes" or "no"?** Clearly, as there are exponentially many dfas with n states, the algorithm could just ask for each dfa with 1 state whether it is equal to the target, then for each dfa with 2 states whether it is equal to the target and so on; so the **question** is whether learning in this scenario is possible with **polynomially many queries**.

If the answer above is "yes" then **describe how the learner works**; if the answer above is "no" then **explain why a learner might need exponentially many queries**.

Solution. The answer is **"no"**. In the queries of this proof, let L stand for the language recognised by the unknown dfa to be learnt. For each word $w \in \{0,1\}^*$ there is a dfa with $|w| + 2$ states which accepts w and rejects all other words and never gets stuck. When learning the subclass of these 1-element languages (which the learner also has to learn when the learner learns all regular languages), the learner can find out whether L equals $\{w\}$ by either asking the membership query "Is $w \in L$?" or the equivalence query "Is L generated by DFA($\{w\}$)" where DFA($\{w\}$) denotes the dfa accepting exactly the word w. For each length n there is one $w \in \{0,1\}^{n-2}$ such that the learner asks first all other words v of the same length (by either a membership or an equivalence query) until it checks whether $L = \{w\}$. Thus for all $n \geq 2$ there is an automaton of n states where the learner needs at least 2^{n-2} queries until it finds the corresponding w. It follows that exponentially many queries are needed in the worst case.

Additional Space for Writing

END OF PAPER