NATIONAL UNIVERSITY OF SINGAPORE

CS 4232 – Theory of Computation
(Semester 1: AY 2017/2018)

Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. Please write your Student Number. Do not write your name.
2. This assessment paper consists of TEN (10) questions and comprises TWENTYONE (21) 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 FIVE (5) marks. The maximum possible marks are 50.

STUDENT 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-free grammar for $L = \{0^n 1 0^n 2 0^m 1 0^m : n > 0 \text{ and } m > 0\}$ with as few non-terminals as possible. Explain the choice of the given grammar.

Solution. The grammar is $(\{S, T\}, \{0, 1, 2\}, \{S \to T2T, T \to 0T0|010\}, S)$.

The grammar uses that L consists of the concatenation of three languages: $L = H \cdot \{2\} \cdot H$ where $H = \{0^n 1 0^n : n > 0\}$ and the words which can be derived from T are exactly the members of H. The grammar for H is one of the standard constructions for context-free grammars.

Note that a grammar with one non-terminal is impossible. Assume otherwise. Then the starting symbol S is the only non-terminal. Now one considers a derivation of the word $0^n 1 0^n 2 0^n 1 0^n$ where only one step in the derivation is missing and n is larger than the largest right side of every rule. Thus the currently derived word is of the form vSw and there is a rule $S \to u$ with $vuw = 0^n 1 0^n 2 0^n 1 0^n$. In at least one of v and w, at least one 1 must occur, since the distance between the two 1s in the target word is larger than $|u|$. Now one can also terminalise the derivation by making S into 0102010, as $S \Rightarrow^* 0102010$ by definition of L. The resulting word has at least three occurrences of 1 and the word is not in L. Thus if a grammar with exactly one non-terminal generates all words in L, then this grammar also generates some words outside L. Therefore the grammar above with two non-terminals is optimal.

Consider the language $L = \{u2v2w : u, v, w \in \{0,1\}^+ \text{ and } |u| = |v| = |w|\}$ and let h be a homomorphism with $h(0) = \varepsilon$, $h(1) = 0$ and $h(2) = 1$.

What is the level of $h(L)$ in the Chomsky Hierarchy:

- ☐ **(a)** regular,
- ☐ **(b)** context-free and not regular,
- ☐ **(c)** context-sensitive and not context-free?

When choosing **(a)**, provide a regular expression for $h(L)$; when choosing **(b)**, provide a context-free grammar and a proof that $h(L)$ is not regular; when choosing **(c)**, provide a proof that $h(L)$ is not context-free.

Solution. The correct choice is **(a)** and $h(L)$ is regular. The regular expression for $h(L)$ is $\sigma = \{0\}^* \cdot \{1\} \cdot \{0\}^* \cdot \{1\} \cdot \{0\}^*$. Any word x generated by σ is of the form $x = 0^i 1 0^j 1 0^k$ for some i, j, k and one can see that

$$x = h(0^{j+k+1} 1^i 2 0^{i+k+1} 1^j 2 0^{i+j+1} 1^k);$$

thus $x \in h(L)$. So the range of h covers all words generated by σ. It is also easy to see that the two 2s in each element of L are mapped by h to 1s while every other symbol is mapped to 0 or ε, thus σ generates every element of $h(L)$.

Construct for the alphabet $\{0, 1, 2, 3, 4\}$ a complete deterministic finite automaton (dfa) with four states which recognises the language

$$L = \{w \in \{0, 1, 2, 3, 4\}^* : \text{val}_5(w) \text{ is even and not a multiple of } 5\}.$$

Here $\text{val}_5(w)$ is the value of the number in the number system with base 5. Recall that this system uses the digits $0, 1, 2, 3, 4$; here some examples: $\text{val}_5(10)$ is five, $\text{val}_5(22)$ is twelve and $\text{val}_5(111)$ is thirty-one. To make the dfa simpler, one defines $\text{val}_5(\varepsilon) = 0$ and $\text{val}_5(0w) = \text{val}_5(w)$ for all $w \in \{0, 1, 2, 3, 4\}^*$. Examples of elements of L are 2, 4, 11, 13, 22, 24 and 101; examples of non-elements of L are 0, 10, 200 and 333.

Solution. The finite automaton is given by the following table.

state	succ at 0	succ at 1	succ at 2	succ at 3	succ at 4	type
a	a	d	c	d	c	start,rej
b	b	c	d	c	d	rej
c	a	d	c	d	c	acc
d	b	c	d	c	d	rej

Note that the multiples of 5 end with 0. So a stands for even and multiple of 5, b stands for odd and multiple of 5, c stands for even and not multiple of 5, d stands for odd and not multiple of 5. As 5 modulo 2 is 1, one can add the digits modulo 2 in order to see whether the number is even or odd.

Construct a context-free grammar in Greibach Normal Form for

$$L = \{0^n 1^m 2^k : n + m < k\}$$

and derive the word 012222 by this grammar. How many steps are needed to derive a word of length ℓ?

In Greibach Normal Form, each left side of a rule is a single non-terminal and each right side of a rule is a terminal followed by some (possibly none) non-terminals. As this language does not contain ε, there cannot be any rule having ε on the right side.

Solution. A possible grammar is $(\{S, T, U, V, W\}, \{0, 1, 2\}, P, S)$ where P contains the rules $S \to 0SU|1TU|2U|2$, $T \to 1TU|2U|2$, $U \to 2U|2$. Now the sample derivation is

$$S \Rightarrow 0SU \Rightarrow 01TUU \Rightarrow 012UUU \Rightarrow 0122UU \Rightarrow 01222U \Rightarrow 012222.$$

As in each step, exactly one terminal is generated, one needs ℓ steps to derive a word of length ℓ.

Recall that a language L satisfies the Block Pumping Lemma with constant k such when a word $u \in L$ of length at least k is split into $k+1$ parts $u_0, u_1, \ldots, u_{k-1}, u_k$ such that all the inner blocks $u_1, \ldots, u_{k-1}$ are non-empty, then one can find i, j with $1 \leq i \leq j < k$ such that $u_0 \ldots u_{i-1}(u_i \ldots u_j)^* u_{j+1} \ldots u_k \subseteq L$.

Either find the smallest constant k such that

$$L = \{w \in \{0,1,2\}^* : w \text{ contains each of } 0,1,2 \text{ at least once}\}$$

satisfies the Block Pumping Lemma with constant k and explain why this constant is the correct choice

Or prove that L does not satisfy the Block Pumping Lemma with any constant k.

Solution. L satisfies the Block Pumping Lemma with constant $k = 5$ but not with constant $k = 4$. In the case that $k = 5$, then one can for each of the letters $0, 1, 2$ reserve one block which contains this letter and one of the inner blocks remains unreserved which can be pumped so that, when pumping down, still all letters are in the word. The choice $k = 4$ is not possible, as one can split the word $u = 012$ into $u_0 = \varepsilon$, $u_1 = 0$, $u_2 = 1$, $u_3 = 2$, $u_4 = \varepsilon$ and pumping down would cause that at least one of the letters 0, 1 and 2 is missing in the pumped down word.

An example in the lecture notes shows that the language $L = \{w \in \{0,1,2\}^* : w$ contains as many 0 as $1\}$ and $H = \{w \in \{0,1,2\}^* : w$ contains as many 0 as $2\}$ are deterministic context-free; that is, L and H can be recognised by a deterministic pushdown automata, but they are not regular. Consider the following languages:

- $I = L \cap H$;
- $J = L \cap (\{0\}^* \cdot \{1\}^* \cdot \{2\}^*)$;
- $K = L \cup H$;
- $O = L \cup (\{0,1\}^* \cdot \{2\} \cdot \{0,1,2\}^*)$.

For each of I, J, K, O, determine which of the following choices applies:

- regular;
- deterministic context-free but not regular;
- context-free but not deterministic context-free;
- context-sensitive but not context-free;
- not context-sensitive.

Give for each language a short reason for the choice taken.

Solution. I: This language is context-sensitive and not context-free; the reason is that the intersection of I and $\{0\}^* \cdot \{1\}^* \cdot \{2\}^*$ gives $\{0^n 1^n 2^n : n \in \mathbb{N}\}$ which is the standard example for a language which is context-sensitive but not context-free. However, context-free languages intersected with regular languages are context-free and therefore I cannot be context-free either.

J: Deterministic context-free languages intersected or unioned with regular sets are again deterministic context-free, thus J as the intersection of L and a regular language is deterministic context-free. The language J equals to $\{0^n 1^n 2^m : n, m \in \mathbb{N}\}$ and is not regular, as the dfa cannot memorise the number of 0 in order to compare them to the number of 1 which follows.

K: This is not deterministic context-free, as the pushdown automaton has to guess in advance whether to count 0 versus 1 or 0 versus 2 and this needs non-determinism. However, the language is context-free, as it is the union of two context-free languages.

O: This language is the union of a deterministic context-free language and a regular language. Thus it is deterministic context-free. As the language $O \cap (\{0\}^* \cdot \{1\}^*)$ equals to the nonregular language $\{0^n 1^n : n \in \mathbb{N}\}$ and regular sets are closed under intersection, the language O cannot be regular.

Let f be a recursive function from $\mathbb{N}$ to $\mathbb{N}$, that is, a function for which there exists an algorithm computing it. Furthermore, assume that $f(n+1) \geq f(n)$ for all n.

Is the set $L_f = \{f(2n+5) : n \in \mathbb{N}\}$ decidable? ☐ Yes, ☐ No.

In the case of the answer "yes", provide an algorithm and explain why it is correct; in the case of the answer "no", provide an example of such an f where the set L_f is not decidable.

Solution. The answer is "yes". One defines a new recursive function g by $g(n) = f(2n+5)$. Note that L_f is equal to the range of g.

In the case that there is a value m such that $g(n) = g(m)$ for all $n \geq m$, the algorithm does the following:

> On input k, check for $n = 0, 1, \ldots, m$ whether $g(n) = k$; if this is true for one of these n then output that $k \in L_f$ else output that $k \notin L_f$.

In the case that there is no such value m then the algorithm does the following search:

> While $g(n) < k$ Do Begin $n = n + 1$ End; If $g(n) = k$ then output that $k \in L_f$ else output that $k \notin L_f$.

The search terminates, as for every m there is an $n > m$ with $g(n) > g(m)$ and thus there is an n with $g(n) \geq k$; the first such n founds then gives away whether k is in the range of f or not.

Which of these two algorithms one has to use, depends on f. So one can prove that the algorithm exists; however, one cannot construct the algorithm from a program for f, as one does not know whether m exists and if it exists, what the value of m is.

Recall that $L \subseteq H$ means that L is a subset of H but can also be equal to H; $L \subset H$ means that L is a proper subset of H and H has an element which is not in L. Furthermore, recall that h is a generalised homomorphism iff $h(\emptyset) = \emptyset$, $h(L \cup H) = h(L) \cup h(H)$, $h(L \cdot H) = h(L) \cdot h(H)$ and $h(L^*) = (h(L))^*$ for all regular sets L and H.

Consider now a generalised homomorphism h mapping regular subsets of $\{0,1,2\}^*$ to regular subsets of $\{0,1,2\}^*$. Which of following two statements are true?

(a) If $\{0\} \subset h(\{0\})$ and $\{1\} \subset h(\{1\})$ and $\{2\} \subset h(\{2\})$ then $L \subset h(L)$ for all nonempty regular subsets L of $\{0,1,2\}^*$: ☐ Yes, ☐ No.

(b) If $\{0\} \subseteq h(\{0\})$ and $\{1\} \subseteq h(\{1\})$ and $\{2\} \subseteq h(\{2\})$ then $L \subseteq h(L)$ for all nonempty regular subsets L of $\{0,1,2\}^*$: ☐ Yes, ☐ No.

Prove each of the answers. If needed, use structural induction.

Solution. **(a)** is false. Note that each generalised homomorphism satisfies by definition that $h(\emptyset) = \emptyset$ and $h(\{\varepsilon\}) = h(\emptyset^*) = (h(\emptyset))^* = \emptyset^* = \{\varepsilon\}$. Furthermore, as $h(\{0,1,2\}^*)$ is a subset of $\{0,1,2\}^*$, it cannot be a proper superset of the same.

(b) is true. Note that every regular subset of $\{0,1,2\}^*$ can be constructed from the base sets $\emptyset$, $\{\varepsilon\}$, $\{0\}$, $\{1\}$ and $\{2\}$ using concatenation, union and Kleene star finitely often. As $h(\emptyset) = \emptyset$ and $h(\{\varepsilon\}) = \{\varepsilon\}$, the conditions in **(b)** imply that $L \subseteq h(L)$ for all base sets.

Now consider for the inductive step any two regular sets L and H with $L \subseteq h(L)$ and $H \subseteq h(H)$. Now the inductive step follows from the properties of a generalised homomorphism in all of the following three cases:

- $L \cup H \subseteq h(L) \cup h(H) = h(L \cup H)$;
- $L \cdot H = \{v \cdot w : v \in L \text{ and } w \in H\} \subseteq \{v \cdot w : v \in h(L) \text{ and } w \in h(H)\} = h(L) \cdot h(H) = h(L \cdot H)$;
- $L^* \subseteq (h(L))^* = h(L^*)$.

Thus for all three possibilities of combining L and H, the new set also satisfies that h maps the set to a not necessarily proper superset. This completes the proof of the inductive step and so the statement follows by structural induction.

A subset A of $\mathbb{N}$ is Diophantine iff there is a polynomial p with n variables for some n and with coefficients from $\mathbb{Z}$ such that, for all $x \in \mathbb{N}$,

$$x \in A \Leftrightarrow \exists y_1, \ldots, y_n \in \mathbb{N}\, [p(x, y_1, \ldots, y_n) = 0].$$

For a number x, let $sqrt(x)$ be the uprounded square-root of x, that is, $sqrt(x) = \min\{y \in \mathbb{N} : x \leq y \cdot y\}$. Now let $A = \{x : sqrt(x) \text{ is odd}\}$. Provide a formula which witnesses that A is Diophantine; such a formula exists, as all r.e. sets of natural numbers are Diophantine.

Solution. First note that

$$A = \{x \in \mathbb{N} : \exists y \in \mathbb{N}\, [(2y)^2 + 1 \leq x \leq (2y+1)^2]\}$$

and now the $\leq$ has to expressed by adding natural numbers which is done by

$$A = \{x \in \mathbb{N} : \exists y_1, y_2, y_3 \in \mathbb{N}\, [(2y_1)^2 + 1 + y_2 = x \text{ and } x + y_3 = (2y_1 + 1)^2]\}$$

and one can now deal with the conjunction in the formula by adding two squares which are 0 when the two equalities are true. This gives

$$A = \{x \in \mathbb{N} : \exists y_1, y_2, y_3 \in \mathbb{N}\, [((2y_1)^2 + 1 + y_2 - x)^2 + (x + y_3 - (2y_1 + 1)^2)^2 = 0]\}$$

and the underlying polynomial is the function

$$x, y_1, y_2, y_3 \mapsto ((2y_1)^2 + 1 + y_2 - x)^2 + (x + y_3 - (2y_1 + 1)^2)^2$$

which can be written as a sum of monomials with integer coefficients.

Question 10 [5 marks]

Let $cbrt(x) = \min\{y \in \mathbb{N} : y^3 \geq x\}$. So $cbrt(0) = 0$, $cbrt(1) = 1$, $cbrt(2) = cbrt(3) = \ldots = cbrt(8) = 2$, $cbrt(9) = cbrt(10) = \ldots = cbrt(27) = 3$.

Write a register machine program which computes the cube-root *cbrt* of the input according to this definition. The registers and constants can be added and subtracted and compared. The possible register values are natural numbers (including 0). The program can have conditional and unconditional jump instructions ("Goto", "If condition Then Goto"). The Return-statement identifies the value of the function.

Solution. For input R_1, the program counts R_2 up until the cube R_4 of R_2 is at least R_1. As cubes are difficult to compute, one uses the following formulas: $(x+1)^3 = x^3 + 3x^3 + 3x + 1$; $(x+1)^2 = x^2 + 2x + 1$. So one does not only maintain $R_2 = x$ and $R_4 = x^3$, but also $R_3 = x^2$ and updates these registers in order R_4, R_3, R_2 in each run through the loop.

Line 1: Function Cbrt(R_1);
Line 2: $R_2 = 0$;
Line 3: $R_3 = 0$;
Line 4: $R_4 = 0$;
Line 5: If $R_4 \geq R_1$ Then Goto Line 10;
Line 6: $R_4 = R_4 + R_3 + R_3 + R_3 + R_2 + R_2 + R_2 + 1$;
Line 7: $R_3 = R_3 + R_2 + R_2 + 1$;
Line 8: $R_2 = R_2 + 1$;
Line 9: Goto Line 5;
Line 10: Return(R_2).

END OF PAPER