



University of the Western Cape
 Department of Computer Science
 Theory of Computation—COS311

Final Examination
 Time: 3 hours.

November 2015
 Marks: 120.

UWC number	Surname, first name
	Solution

No calculators or cellphones are allowed. Please answer all the questions directly on this question paper. No other answers will be marked.

1. (a), *First*, for (1) mark, *give* one example of the numbers described below that is negative, has at least ten characters and a positive exponent and then (b) give a *regular expression* in the style of **flex** for:

a *decimal floating point number* that starts with an *optional* ‘-’, and has at most one digit before a *required* decimal point, ‘.’, and from one to eight digits after the point. Follow this with an *optional exponent*: that starts with a *compulsory* exponent indicator in the form of an ‘e’, or ‘E’, followed by an *optional* plus or minus sign and which is followed by one, two or three digits. A correct **flex** regular expression earns (9) marks. [10]

(a) An example $-9.87654321E+01$ and (b) the regular expression
 $[-]?[0-9]?[.][0-9]{1,8}([eE][+-][0-9]{1,3})?$
 or $[-]?[0-9]{0,1}[.][0-9]{1,8}([eE][+-][0-9]{1,3})?$

2. Answer **true** or **false**: One (1) mark is *deducted* for each incorrect answer, so DO NOT GUESS. [15]

- (a) ww^R is a context-free language, where $w \in \Sigma^*$ and $\Sigma = \{0, 1\}$. (1)
- (b) ww is a context-free language, where $w \in \Sigma^*$ and $\Sigma = \{a, b\}$. (1)
- (c) a^*b^* is a context-free language. (1)
- (d) $a^n b^n$ is a regular language. (1)
- (e) $a^n b^n \subseteq a^* b^*$. (1)
- (f) The set of regular languages \subseteq The set of context-free languages. (1)
- (g) If L is context free, then L^* is context free. (1)

- (h) If L_1 and L_2 are context-free then $L_1 \circ L_2$ is context-free. (1)
- (i) If L_1 and L_2 are context-free then $L_1 \cap L_2$ is context-free. (1)
- (j) If L is context free, then L is Turing recognizable. (1)
- (k) The language $\{a^j b^k c^\ell \mid j \geq 0, k \geq 0, \ell \geq 0\}$ is regular. (1)
- (l) The language $\{a^j b^k c^\ell \mid j \geq 0, k \geq 0, \ell \geq 0, j = \ell\}$ is context free. (1)
- (m) The set $A_{REG} = \{\langle R, w \rangle \mid R \text{ is a regular expression that generates the input string } w\}$ is decidable. (1)
- (n) The set $A_{CFG} = \{\langle G, w \rangle \mid G \text{ is a CFG that accepts the input string } w\}$ is decidable. (1)
- (o) The set $A_{TM} = \{\langle M, w \rangle \mid M \text{ is a TM that accepts the input string } w\}$ is decidable. (1)
- (p) The set $\overline{A_{TM}}$, the complement of A_{TM} , is Turing recognizable. (1)
3. In these questions you must fill in a short answer as a number or a string of symbols or letters. [5]
- (a) Suppose that the rational numbers are enumerated using the mapping for J below to map $\frac{p}{q}$ onto the natural numbers. Give the values of p and q that map onto 49.
 $J[p, q] = (p + q - 1)(p + q - 2)/2 + q$ (1) (1)
- (b) Give a regular expression for $\{a^2 b^2 c^n \mid n \geq 1\}$ or write the word *impossible*. (1)
- (c) Given the length of a parsable string $|w| = n$, what is the height of the parse tree found for it when applying the CYK algorithm. (1)
- (d) Given the rules $\mathcal{R} \equiv \{S' \rightarrow aSbSc, S \rightarrow \varepsilon\}$, after removing the $S \rightarrow \varepsilon$, how many rules must be added to make the grammar ε -free? (1)
- (e) Given the same set of rules $\mathcal{R} \equiv \{S' \rightarrow aSbSc, S \rightarrow \varepsilon\}$, add rules to make the grammar ε -free and then alter the rules to turn the grammar into Chomsky normal form. [5]

Add the three rules (1) $S' \rightarrow abSc$, (2) $S' \rightarrow aSbc$, and (3) $S' \rightarrow abc$. All these rules must be rewritten into $X \rightarrow YZ$ form. Let $A \rightarrow a$, $B \rightarrow b$, and $C \rightarrow c$, then (1) become $S' \rightarrow AU_1$, $U_1 \rightarrow BU_2$, $U_2 \rightarrow SC$, and production (2) becomes $S' \rightarrow AU_3$, $U_3 \rightarrow SU_4$, $U_4 \rightarrow BC$, and finally, (3) becomes $S' \rightarrow AU_4$.

4. (a) Given a regular language L , the pumping lemma states that all strings $s \in L$ can be pumped if $|s| \geq p$, where p is the pumping length. The result of the pumping lies in L . The pumping repeats a section of the string $s = xyz$ under the following conditions:
1. for any $i \geq 0$, $xy^iz \in L$,
 2. $|y| \geq 0$ is incorrect. It should be $|y| > 0$ for (1) mark, and
 3. $|xy| < p$ is incorrect. It should be $|xy| \leq p$ for (1) mark.

If the pumping lemma for regular languages, above, is stated wrongly please correct all the errors by marking them clearly directly on the question paper before proceeding with the rest of the question. (2) Let $\Sigma = \{a, b\}$. Is the language $L = \{a^n b^n \mid n \geq 0\}$ regular/not regular? Prove it using the corrected pumping lemma. (9) [7]

Suppose L is regular, then it can be pumped. With p the pumping length, choose $s = a^p b^p$, i.e. $|s| \geq p$. Split s into three pieces, $s = xyz$. So xy^iz should be in L for any $i \geq 0$.

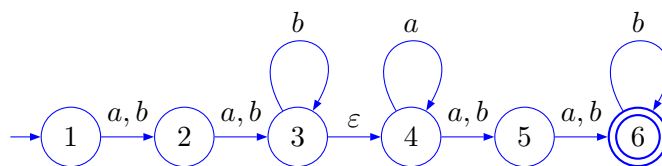
- i. Choose $y = a^m$, where $0 < m < p$, then $x = a^{p-m}$ and $z = b^p$. Taking $i = 0$ in $t = xy^iz$ gives $t = a^{p-m} b^p$ which is not in L and neither is any other t when $i > 0$.
- ii. Similarly taking $y = b^m$, where $0 < m < p$, $z = b^{p-m}$, and $x = a^p$. Taking $i = 0$ in $t = xy^iz$ gives $t = a^p b^{p-m}$ which is not in L and $t \notin L$ for any $i > 1$. Straddling the centre will contain only bs .
- iii. Let y straddle any substring of different letters of $a^p b^p$. For example the the string y has at least one a and at least one b then putting $i = 0$ reduces the length and $t \in L$, but any other value of i will cause the as and bs to appear out of order.

So it is impossible to pump this string. Our assumption that L is regular is thus false, so L is not regular.

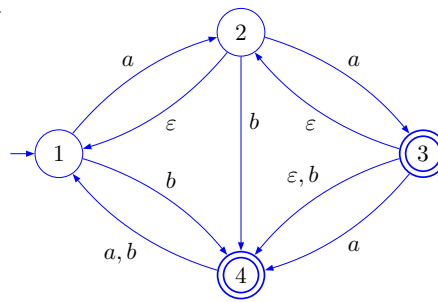
5. Give a regular expression for the language over the alphabet $\Sigma = \{a, b\}$ that accepts all strings that end with exactly two as . [2]

$(a + b)^* b a a + a a$, or alternatively $(a | b)^* b a a | a a$.

6. Convert the regular expression $(a+b)^2 b^* a^* (a+b)^2 b^*$ to an NFA in the form of a state diagram. [3]



7. Consider the state diagram below



(a) What is the ε -closure of each of the states of the above NFA? [2]

$E(1) = \{1\}$, $E(2) = \{1, 2\}$, $E(3) = \{1, 2, 3, 4\}$, and $E(4) = \{4\}$. The answers written as $E(q_2) = \{q_1, q_2\}$, or even as $E(3) = 1234$, are all acceptable.

(b) Use the subset construction to build a *transition table* for a DFA accepting the same language as the NFA in the state diagram above. [8]

We only need to consider *reachable* subsets, starting at state $\{q_1\}$, which we abbreviate conveniently as 1. Accordingly state $\{q_2, q_3, q_4\}$ is denoted by 234, etc. We can immediately draw the final DFA, M , as a state diagram, but instead, here, we build the transition table starting at the start state q_1 , now called state 1, as follows:
The DFA, M as a transition table:

δ	a	b
\emptyset	\emptyset	\emptyset
$\rightarrow 1$	12	4
2	1234	4
*3	4	4
*4	1	1
12	1234	4
*1234	1234	14
*14	12	14

Mark off all the final states with ‘*’. Delete the unreachable states \emptyset and states $q_2 \equiv 2$ and $q_3 \equiv 3$ yielding:

δ	a	b
$\rightarrow 1$	12	4
*4	1	1
12	1234	4
*1234	1234	14
*14	12	14

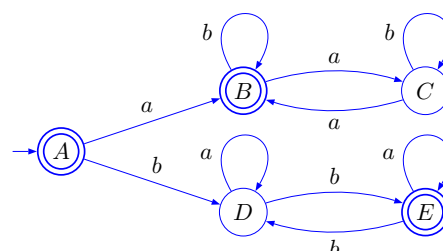
8. (a) Describe the production rules of *context-free grammars*. [2]

Let a *context-free grammar* G be the tuple (V, Σ, R, S) where V is the set of variables or non-terminals, Σ is the set of terminals, R is the set of rules, and S is the start variable, then its production rules have the form $A \rightarrow \gamma$, $A \in V$ and where $\gamma \in (V \cup \Sigma)^*$ which includes ε .

(b) Describe the production rules of *regular grammars*. [2]

A grammar G is regular when its rules have the form $A \rightarrow tA$ or $A \rightarrow t$, where A is a variable and t is a terminal.

(c) Let L be the language recognized by the DFA to the right. Give the productions of a *regular* CFG used to generate the language $L - \{\varepsilon\}$.



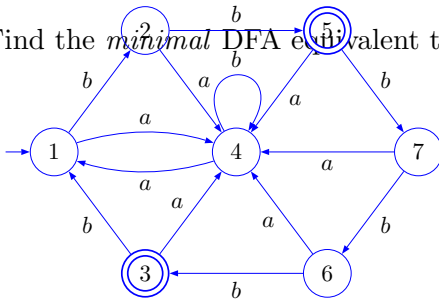
[3]

$A \rightarrow aB \mid bD \mid a$
$B \rightarrow aC \mid bB \mid b$
$C \rightarrow bC \mid aB \mid a$
$D \rightarrow aD \mid bE \mid b$
$E \rightarrow aD \mid bE \mid a$

(d) Give a regular expression for the ε -free language $L - \varepsilon$, recognized by the above state diagram in Question 8(c). [3]

$a(b^* + (ab^*a)^*)^* + ba^*b(a^* + (ba^*b)^*)^*$

9. Find the *minimal* DFA equivalent to the following DFA described by the state diagram:



First, mark in the table on the right all the states that are not equivalent.

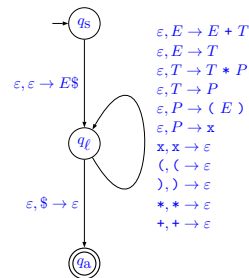
1						
✓	2					
✓	✓	3*				
✓	✓	✓	4			
✓	✓	•	✓	5*		
✓	•	✓	✓	✓	6	
•	✓	✓	✓	✓	✓	7

Now give the minimal DFA as a transition TABLE:

[10]

δ	a	b
\rightarrow 17	4	26
26	4	35
* 35	4	17
4	17	4

10. Given the state diagram below for a nondeterministic PDA that recognizes a language, give a context free grammar G for this language.



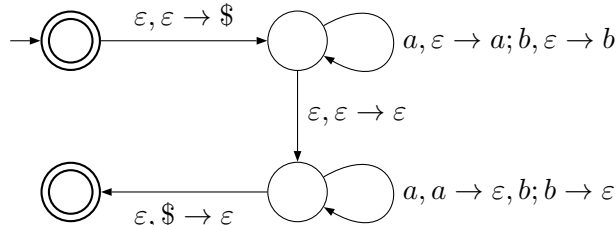
[3]

The context-free grammar is

$$\begin{aligned}
 (V &= \{E, T, P\}, \\
 \Sigma &= \{(\cdot), *, x\}, \\
 \text{Rules} &= \{E \rightarrow E + T \mid T, \\
 &\quad T \rightarrow T * P \mid P, \\
 &\quad P \rightarrow (E) \mid x\}, \\
 E &= \text{the start symbol}).
 \end{aligned}$$

11. The language $L = \{x^R x \mid x \in \{a, b\}^*\}$ is context free. Illustrate two different proofs of this fact. [8]

- (a) The grammar with state S , and rules $S \rightarrow aSa \mid bSb \mid \varepsilon$ recognizes L . Single letters are not palindromes, but the empty string is.
- (b) Build a PDA to recognize palindromic pairs, that works as follows: It first places a $\$$ onto the stack and enters the 'left-half-stacking' state. Here it gobbles the input up to the left of the middle and stacks each letter of the alphabet on the push-down stack. When it passes the middle it moves nondeterministically to the destacking state that matches input letters with the item on top of the stack until it reaches a $\$$, when it enters the accept state.



12. Use the Cocke-Younger-Kasami algorithm to decide if the string $s = vwuxyu$ is generated by the grammar G , i.e., $v \in L(G)$, and that the string $t = uwxyuv$ is *not* generated by the grammar. and

the grammar G is given by the following rules.

$$\begin{aligned} U &\rightarrow US \mid VY \mid \mathbf{u} \\ S &\rightarrow WU \mid YU \\ Y &\rightarrow UX \\ W &\rightarrow \mathbf{w} \\ Y &\rightarrow \mathbf{y} \\ V &\rightarrow \mathbf{v} \\ X &\rightarrow \mathbf{x} \end{aligned}$$

where U is the start symbol. First *complete* the following tables: (8)

Use this table to show that $s \in L(G)$.

0							
	\mathbf{v}						
V	1						
		\mathbf{u}					
s	U	2					
			\mathbf{w}				
\emptyset	\emptyset	W	3				
				\mathbf{u}			
\emptyset	U	S	U	4			
					\mathbf{x}		
U	Y	\emptyset	Y	X	5		
						\mathbf{y}	
\emptyset	\emptyset	\emptyset	U	S	Y	6	
							\mathbf{u}
U	\emptyset	\emptyset	\emptyset	\emptyset	S	U	7

Use this table to show that $t \notin L(G)$.

0							
	\mathbf{u}						
U	1						
		\mathbf{w}					
\emptyset	W	2					
			\mathbf{x}				
\emptyset	\emptyset	X	3				
				\mathbf{u}			
\emptyset	\emptyset	\emptyset	U	4			
					\mathbf{y}		
\emptyset	\emptyset	\emptyset	\emptyset	Y	5		
						\mathbf{u}	
\emptyset	\emptyset	\emptyset	U	S	U	6	
							\mathbf{v}
\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	V	7

Now write out a detailed *rightmost* derivation of $U \Rightarrow^* s$. (2)

[10]

$U \Rightarrow US \Rightarrow UYU \Rightarrow UY\mathbf{u} \Rightarrow U\mathbf{y}\mathbf{u} \Rightarrow VY\mathbf{y}\mathbf{u} \Rightarrow VUX\mathbf{y}\mathbf{u} \Rightarrow VU\mathbf{x}\mathbf{y}\mathbf{u} \Rightarrow VUS\mathbf{x}\mathbf{y}\mathbf{u} \Rightarrow VUWU\mathbf{x}\mathbf{y}\mathbf{u} \Rightarrow VUW\mathbf{u}\mathbf{x}\mathbf{y}\mathbf{u} \Rightarrow VUw\mathbf{u}\mathbf{x}\mathbf{y}\mathbf{u} \Rightarrow Vuw\mathbf{u}\mathbf{x}\mathbf{y}\mathbf{u} \Rightarrow vw\mathbf{u}\mathbf{x}\mathbf{y}\mathbf{u} = s$.

If you guessed that \mathbf{w} stands for $+$, \mathbf{t} stands for $*$, \mathbf{v} stands for $($, and \mathbf{x} stands for $)$, then the correctness of the \mathbf{v} and incorrectness of \mathbf{w} become obvious.

13. Consider the following Turing machine, then answer the questions using this machine

δ	0	1	x	#	\sqcup
$\rightarrow q_1$	q_2, x, R	q_3, x, R	q_R, x, R	$q_8, \#, R$	q_R, \sqcup, R
q_2	$q_2, 0, R$	$q_2, 1, R$	$q_R, 1, R$	$q_4, \#, R$	$q_R, \#, R$
q_3	$q_3, 0, R$	$q_3, 1, R$	$q_R, 1, R$	$q_5, \#, R$	$q_R, \#, R$
q_4	q_6, x, L	$q_R, 1, R$	q_4, x, R	$q_R, \#, R$	q_R, \sqcup, R
q_5	$q_R, 0, R$	q_6, x, L	q_5, x, R	$q_R, \#, R$	q_R, \sqcup, R
q_6	$q_6, 0, L$	$q_6, 1, L$	q_6, x, L	$q_7, \#, L$	q_R, \sqcup, R
q_7	$q_7, 0, L$	$q_7, 1, L$	q_1, x, R	q_R, x, R	q_R, \sqcup, R
q_8	$q_R, 0, R$	$q_R, 1, R$	q_8, x, R	q_R, x, R	q_A, \sqcup, R^*

- (a) While processing the string $w = 01\#01\sqcup\dots$, starting with the machine in the configuration $q_101\#01$, show the configurations the machine traverses through to accept w . [4]

1. $q_101\#01$	10. $xx\#q_5x1$
2. $xq_21\#01$	11. $xx\#xq_51$
3. $x1q_2\#01$	12. $xx\#q_6xx$
4. $x1\#q_401$	13. $xxq_6\#xx$
5. $x1q_6\#x1$	14. $xq_7x\#xx$
6. $xq_71\#x1$	15. $xxq_1\#xx$
7. $q_7x1\#x1$	16. $xx\#q_8xx$
8. $xq_11\#x1$	17. $xx\#xq_8x$
9. $xxq_3\#x1$	18. $xx\#xxq_8$

- (b) Show the configurations this machine goes through in order to reject the string $i0\#01$. [3]

The string $w = 000\sqcup\sqcup\dots$ is not in A .

1. $q_10\#01$
2. $xq_2\#01$
3. $x\#q_401$
4. $xq_6\#x1$
5. $q_7x\#x1$
6. $xq_1\#x1$
7. $x\#q_8x1$
8. $x\#xq_81$

14. What is meant by *Turing-recognizability*. [3]

A language is Turing-recognizable if some Turing machine recognizes it.

15. What is a *decider*? Give an example of a decider. [4]

Turing machines that never loop are called deciders. The example in Question 13 is a decider. Whether or not a given string is accepted by a given DFA is decidable.
--

16. Every context-free language is decidable. Describe a proof for this.

[8]

Every context-free language (CFL) can be written in the form of a CF grammar, and this grammar can in turn always be written in Chomsky normal form. In turn the Cocke-Younger-Kasami decision procedure can be applied to decide whether any sentence is derivable from the start symbol of the grammar. So every sentence of a CFL can be shown to be derivable or not in that language. Hence CFLs are decidable.

17. Consider the Turing recognizability of

$$D = \{P \mid P \text{ is a polynomial over } x \text{ with an integral root}\}$$

Is D Turing recognizable? Answer YES or NO (1), and give an explanation (3).

[4]

YES. The following TM can recognize D . $M_1 =$ "Input is a polynomial P over the variable x .

1. Evaluate P with x set successively to $0, 1, -1, 2, -2, 3, -3, \dots$. If at any point the polynomial evaluates to 0 then accept."

If P has an integral root, M_1 eventually finds it, and accepts. If P does not have an integral root then M_1 runs on in a closed loop. It is a simple matter to extend this to a TM, M that handles the multivariable case. It is also easy to turn this into a decider for the univariate case since the roots of polynomials in a single variable have fixed upper and lower bounds that can easily be calculated from the coefficients. These bounds are $\pm |k \frac{c_{\max}}{c_n}|$ where c_{\max} is the coefficient with the biggest absolute value and c_n is the coefficient of the highest order term and k is the number of terms in the polynomial. After running through all these values without finding a root the decider *rejects* otherwise it will *accepts*.

Total [120]