University of the Western Cape

Department of Computer Science

Theory of Computation—COS311

Final Examination — November 2015

Time: 3 hours. — 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) $ww^{\mathcal{R}}$ is a context-free language, where $w \in \Sigma^*$ and $\Sigma = \{0, 1\}$. (1) true

   (b) ww is a context-free language, where $w \in \Sigma^*$ and $\Sigma = \{a, b\}$. (1) false

   (c) a^*b^* is a context-free language. (1) true

   (d) a^nb^n is a regular language. (1) false

   (e) $a^nb^n \subseteq a^*b^*$. (1) true

   (f) The set of regular languages $\subseteq$ The set of context-free languages. (1) true

   (g) If L is context free, then L^* is context free. (1) true

(h) If L_1 and L_2 are context-free then $L_1 \circ L_2$ is context-free. (1) true

(i) If L_1 and L_2 are context-free then $L_1 \cap L_2$ is context-free. (1) false

(j) If L is context free, then L is Turing recognizable. (1) true

(k) The language $\{a^j b^k c^\ell \mid j \geq 0, k \geq 0, \ell \geq 0\}$ is regular. (1) true

(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) true

(m) The set $A_{REX} = \{\langle R, w\rangle \mid R$ is a regular expression that generates the input string $w\}$ is decidable. (1) true

(n) The set $A_{CFG} = \{\langle G, w\rangle \mid G$ is a CFG that accepts the input string $w\}$ is decidable. (1) true

(o) The set $A_{TM} = \{\langle M, w\rangle \mid M$ is a TM that accepts the input string $w\}$ is decidable. (1) false

(p) The set $\overline{A_{TM}}$, the complement of A_{TM}, is Turing recognizable. (1) false

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) $p = 7$ (1) $q = 4$

(b) Give a regular expression for $\{a^2b^2c^n | n \geq 1\}$ or write the word *impossible*. (1) $aabbcc^*$

(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) n

(d) Given the rules $\mathscr{R} \equiv \{S' \to aSbSc, S \to \varepsilon\}$, after removing the $S \to \varepsilon$, how many rules must be added to make the grammar ε-free? (1) 3

(e) Given the same set of rules $\mathscr{R} \equiv \{S' \to aSbSc, S \to \varepsilon\}$, 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) $S' \to abSc$, (2) $S' \to aSbc$, and (3) $S' \to abc$. All these rules must be rewritten into $X \to YZ$ form. Let $A \to a$, $B \to b$, and $C \to c$, then (1) become $S' \to AU_1$, $U_1 \to BU_2$, $U_2 \to SC$, and production (2) becomes $S' \to AU_3$, $U_3 \to SU_4$, $U_4 \to BC$, and finally, (3) becomes $S' \to AU_4$.

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 $L = \{a^nb^n \mid n \geq 0\}$ 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 $s = a^pb^p$, 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 $t = a^pb^{p-m}$ which is not in L and $t \notin L$ for any $i > 1$. Straddling the centre will contain only bs.

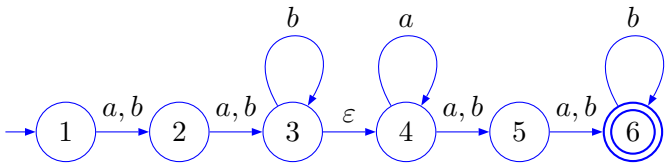
   iii. Let y straddle any substring of different letters of a^pb^p. For example the the string y has at least one a and at least one b then putting $i = 0$ reduces the length and $t \in L$, but any other value of i will cause the as and bs to appear out of order.

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

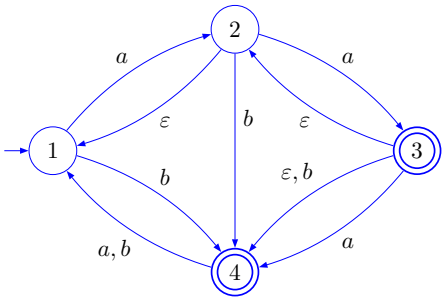
5. Give a regular expression for the language over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ that accepts all strings that end with exactly two `a`s. [2]

   $(\mathtt{a} + \mathtt{b})^*\mathtt{baa} + \mathtt{aa}$, or alternatively $(\mathtt{a} \mid \mathtt{b})^*\mathtt{baa} \mid \mathtt{aa}$.

6. Convert the regular expression $(\mathtt{a+b})^2\mathtt{b}^*\mathtt{a}^*(\mathtt{a+b})^2\mathtt{b}^*$ 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 acceptible.

(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 abreviate 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 transtion 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

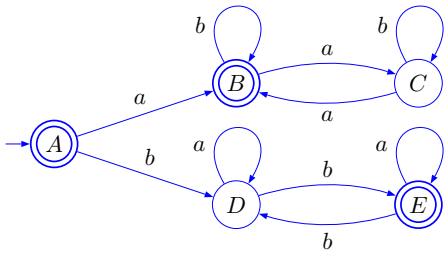
8. (a) Describe the production rules of *context-free grammars*. [2]

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 $L - \{\varepsilon\}$.



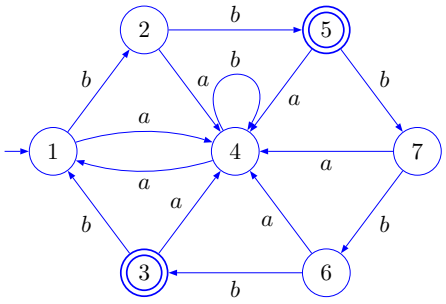
[3]

$$
\begin{aligned}
A &\rightarrow aB \mid bD \mid a \\
B &\rightarrow aC \mid bB \mid b \\
C &\rightarrow bC \mid aB \mid a \\
D &\rightarrow aD \mid bE \mid b \\
E &\rightarrow aD \mid bE \mid a
\end{aligned}
$$

(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:



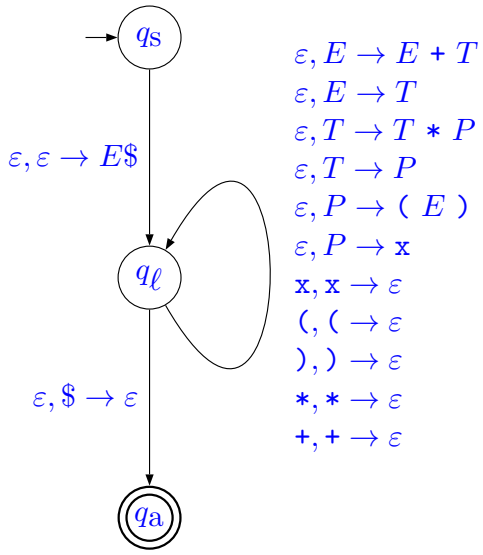
First, mark in the table on the right all the states that are not equivalent.

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]



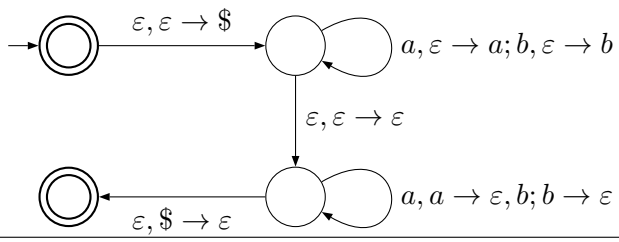
The context-free grammar is

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

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

(a) The grammar with state S, and rules $S \to aSa \mid bSb \mid \varepsilon$ 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 $s = \texttt{vuwuxyu}$ is generated by the grammar G, i.e., $v \in L(G)$, and that the string $t = \texttt{uwxuyuv}$ is *not* generated by the grammar. and the grammar G is given by the following rules.

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

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

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

0							
	v						
V	1						
		u					
∅	U	2					
			w				
∅	∅	W	3				
				u			
∅	U	S	U	4			
					x		
U	Y	∅	Y	X	5		
						y	
∅	∅	∅	∅	∅	Y	6	
							u
U	∅	∅	∅	∅	S	U	7

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

0							
	u						
U	1						
		w					
∅	W	2					
			x				
∅	∅	X	3				
				u			
∅	∅	∅	U	4			
					y		
∅	∅	∅	∅	Y	5		
						u	
∅	∅	∅	U	S	U	6	
							v
∅	∅	∅	∅	∅	∅	V	7

Now write out a detailed *rightmost* derivation of $U \overset{*}{\Rightarrow} s$. (2) [10]

$U \Rightarrow US \Rightarrow UYU \Rightarrow UY\texttt{u} \Rightarrow U\texttt{yu} \Rightarrow VY\texttt{yu} \Rightarrow VUX\texttt{yu} \Rightarrow VU\texttt{xyu} \Rightarrow VUS\texttt{xyu} \Rightarrow VUWU\texttt{xyu}$
$\Rightarrow VUW\texttt{uxyu} \Rightarrow VUw\texttt{uxyu} \Rightarrow V\texttt{uwuxyu} \Rightarrow \texttt{vuwuxyu} = s.$
If you guessed that `w` stands for `+`, `t` stands for `*`, `v` stands for `(`, and `x` stands for `)`, then the correctness of the v and incorrectness of w become obvious.

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

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

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

1. $q_101\#01$
2. $\mathtt{x}q_21\#01$
3. $\mathtt{x1}q_2\#01$
4. $\mathtt{x1}\#q_401$
5. $\mathtt{x1}q_6\#\mathtt{x1}$
6. $\mathtt{x}q_71\#\mathtt{x1}$
7. $q_7\mathtt{x1}\#\mathtt{x1}$
8. $\mathtt{x}q_11\#\mathtt{x1}$
9. $\mathtt{xx}q_3\#\mathtt{x1}$
10. $\mathtt{xx}\#q_5\mathtt{x1}$
11. $\mathtt{xx}\#\mathtt{x}q_51$
12. $\mathtt{xx}\#q_6\mathtt{xx}$
13. $\mathtt{xx}q_6\#\mathtt{xx}$
14. $\mathtt{x}q_7\mathtt{x}\#\mathtt{xx}$
15. $\mathtt{xx}q_1\#\mathtt{xx}$
16. $\mathtt{xx}\#q_8\mathtt{xx}$
17. $\mathtt{xx}\#\mathtt{x}q_8\mathtt{x}$
18. $\mathtt{xx}\#\mathtt{xx}q_8$

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

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

1. $q_10\#01$
2. $\mathtt{x}q_2\#01$
3. $\mathtt{x}\#q_401$
4. $\mathtt{x}q_6\#\mathtt{x1}$
5. $q_7\mathtt{x}\#\mathtt{x1}$
6. $\mathtt{x}q_1\#\mathtt{x1}$
7. $\mathtt{x}\#q_8\mathtt{x1}$
8. $\mathtt{x}\#\mathtt{x}q_81$

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

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

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 $0, 1, -1, 2, -2, 3, -3, \ldots$. 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 polinomials 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 coefficent 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 fiding a root the decider *rejects* otherwise it will *accepts*.

Total [120]