PAPER CODE NO.	EXAMINER : Paul W. Goldberg
COMP218	DEPARTMENT : Computer Science Tel. No. 0151-7954259

UNIVERSITY OF LIVERPOOL

Second Semester Examinations 2011/12

Decision, Computation and Language

TIME ALLOWED : Two Hours

INSTRUCTIONS TO CANDIDATES

Answer **ALL** questions from Section A and **TWO** from Section B.

If you attempt to answer more questions than the required number of questions (in any section), the marks awarded for the excess questions answered will be discarded (starting with your lowest mark).

UNIVERSITY OF LIVERPOOL

Section A

1. **a.** Write down a regular expression for the set of words over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ that begin and end with $\mathtt{a}$, and a regular expression for the set of words that contains at least one occurrence of each of $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. [5 marks]

b. Consider the following nondeterministic finite automaton $\mathcal{A} = (Q, A, \phi, i, T)$ where the alphabet $A = \{\mathtt{a}, \mathtt{b}\}$, states $Q = \{i, r, s, t\}$, accepting states $T = \{s, t\}$ and transition function ϕ defined by

$$\begin{array}{rr} \phi(i, \mathtt{a}) = r & \phi(r, \mathtt{b}) = s \\ \phi(s, \mathtt{a}) = \{r, t\} & \phi(t, \mathtt{b}) = t \end{array}$$

i. Draw a diagram of $\mathcal{A}$ and write down a regular expression for the language it accepts. [6 marks]

ii. Find a string in the language that has two accepting paths; write down the accepting paths. [4 marks]

2. Consider the set of words over the alphabet $A = \{\mathtt{a}, \mathtt{b}\}$ that are palindromes that consist of two equal-length strings of $\mathtt{a}$'s with a string of $\mathtt{b}$'s in the middle, or else two equal-length strings of $\mathtt{b}$'s with a string of $\mathtt{a}$'s in the middle. For example, strings such as $\mathtt{aaabbaaa}$, $\mathtt{baaaab}$, $\mathtt{aaaa}$ (where the last one is allowed given that some of the strings may be empty).

a. Write down a *context-free grammar* for this language. [6 marks]

b. Write down derivations for the strings $\mathtt{bbabb}$, $\mathtt{aaaa}$ using your grammar. [6 marks]

c. Is your grammar ambiguous? Justify your answer. [3 marks]

3. Consider the following context-free grammar:

$$\begin{array}{rcl} S & \longrightarrow & \mathtt{a}S\mathtt{b} \mid \mathtt{c}A \\ A & \longrightarrow & \mathtt{a} \mid \mathtt{a}A \mid \epsilon \end{array}$$

a. Which rules of this grammar are *not* in Greibach normal form? Write down an equivalent grammar that is in Greibach normal form. [10 marks]

b. Explain how a Greibach normal form grammar can be converted into an equivalent pushdown automaton. [5 marks]

4. **a.** Explain how to construct a Turing machine that accepts the set of strings over the binary alphabet $\{\mathtt{0}, \mathtt{1}\}$ for which the number of $\mathtt{1}$'s is greater than the number of $\mathtt{0}$'s. (A complete set of transitions is not needed; a general description is sufficient.) [10 marks]

b. Is the above language regular or context-free? Briefly justify your answer. [5 marks]

UNIVERSITY OF LIVERPOOL

Section B

Answer **two** questions in this section.

1. **a.** We know that a standard DFA has one initial state and may have more than one accepting state. Suppose that I define an "extended DFA" to be one that is allowed to have more than one initial state, with the rule that a string is accepted provided that there is an accepting path starting from at least one of the initial states. Prove that any language accepted by an "extended DFA" can be accepted by a standard DFA.

[7 marks]

b. Suppose we consider a restricted class of DFAs that have only one accepting state. Give an example of a regular language that is *not* accepted by such a restricted DFA, explaining why it is not accepted by any such restricted DFA. [3 marks]

c. Suppose that R, S and T are regular expressions. Give a detailed proof of the equivalence: $(R \cup S)^* = (R^* \cup S^*)^*$. [10 marks]

2. **a.** Write down the FIRST and FOLLOW sets of the variables of the grammar below, and explain why it is not LL(1) parsable.

$$\begin{array}{lcl} S & \longrightarrow & \mathtt{c}S \\ S & \longrightarrow & \mathtt{c}A \\ A & \longrightarrow & \mathtt{c}B \\ A & \longrightarrow & \mathtt{a} \\ B & \longrightarrow & \mathtt{b} \end{array}$$

[10 marks]

b. Given an equivalent grammar that is LL(1) parsable, and write down a parse table for the new grammar. [10 marks]

3. **a.** It was claimed in lectures that it is undecidable to check whether two context-free grammars are equivalent. Write down a general explanation of what this means, that would be useful for someone who knows about context-free grammars but not about undecidability. [5 marks]

b. It is also the case that it is undecidable to check whether a given CFG can generate every word over its alphabet. Explain how one can use this fact to deduce the fact that testing equivalence of CFGs is undecidable. [5 marks]

c. Is it undecidable to test whether two grammars that are in Chomsky normal form, are equivalent? Justify your answer. [5 marks]

d. Define "recursively enumerable", and explain why, for any context-free grammar, the set of all words it can generate is recursively enumerable. [5 marks]