PAPER CODE NO. **COMP209**	EXAMINER : Dr. Paul W. Goldberg DEPARTMENT : Computer Science Tel. No. 54259

THE UNIVERSITY
of LIVERPOOL

JANUARY 2007 EXAMINATIONS

Bachelor of Science : Year 2
No qualification aimed for : Year 1

Decision, Computation and Language

TIME ALLOWED : 2 Hours

INSTRUCTIONS TO CANDIDATES

Answer **all** questions in Section 1 and **two** questions from Section 2. Section 1 accounts for 60% of credit and Section 2 accounts for the remaining 40%.

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

THE UNIVERSITY
of LIVERPOOL

Section 1

Answer **all** questions in this section.

1.

a. For each of the following pairs of regular expressions over alphabet $\{\mathtt{a}, \mathtt{b}\}$, say whether they represent the same language, and if they do not, give an example of a word that belongs to one language but not the other.

   i. $\{\mathtt{ab}, \mathtt{aab}\}(\{\mathtt{ab}, \mathtt{aab}\})^*$ and $(\{\mathtt{ab}, \mathtt{aab}\})^*\mathtt{aab}$ [2 marks]

   ii. $(\{\mathtt{ab}, \mathtt{aab}\})^*$ and $(\{\mathtt{ab}, (\mathtt{aab})^*\})^*$ [2 marks]

   iii. $(\{\mathtt{b}, \mathtt{ba}\})^* \cup (\{\mathtt{ba}, \mathtt{baa}\})^*$ and $(\{\mathtt{b}, \mathtt{ba}, \mathtt{baa}\})^*$ [2 marks]

b. Give an example of a language that is context-free but is not regular. Write down a context-free grammar for the language. [5 marks]

c. Consider the following regular grammar, with start symbol S and additional variables A and B.

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

Find a regular expression that represents the language generated by the above grammar. (You may find it helpful to simplify the grammar via back substitution of variables.)

[4 marks]

2.

a. Draw a diagram of a deterministic finite automaton that accepts the language (over alphabet $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$) given by the regular expression $\mathtt{c}^*(\mathtt{a}^* \cup \mathtt{b}^*)$. [5 marks]

b. Explain how your DFA of part (a) can be altered to obtain a new DFA that accepts the complement of the language of the original DFA (that is, the set of all words not accepted by the original one). [5 marks]

c. Describe a general method for modifying a deterministic finite automaton M so that if L is the language accepted by M, the modified finite automaton (which may be non-deterministic) accepts L^*, the closure of L. [5 marks]

3. A language L is said to be prefix-closed if, given any word $w \in L$, all prefixes of w are also members of L. For example, any prefix-closed language that contains the word `cats` must also contain the words `cat`, `ca`, `c` and the empty word ϵ.

Consider the languages $L_1 = \{a^n b^m \ : \ n > m\}$, and $L_2 = \{b^m a^n \ : \ n > m\}$. (So, the first of these, L_1, is strings consisting of `a`'s followed by `b`'s, where there are more `a`'s than `b`'s. L_2 is `b`'s followed by `a`'s, where again there are more `a`'s than `b`'s.)

a. Which of these is prefix-closed and why? [5 marks]

b. Use the pumping lemma to prove that L_2 is not regular. [5 marks]

c. Is L_1 regular? Give a brief explanation for your answer. [5 marks]

4.

a. Explain what is meant by a *recursive* language, and what is meant by a *recursively enumerable* language. [5 marks]

b. Suppose that two languages L_1 and L_2 are accepted by Turing machines M_1 and M_2. Explain how M_1 and M_2 can be used to construct a Turing machine M that accepts words that belong to both L_1 and L_2. [10 marks]

Section 2

Answer **two** questions in this section.

5.

a. Give an unambiguous context-free grammar that can generate words of the form $\mathtt{a}^*\mathtt{b}^*\mathtt{c}^*$ (strings consisting of a sequence of `a`'s followed by a sequence of `b`'s followed by a sequence of `c`'s). Explain why your grammar is unambiguous. [5 marks]

b. Recall Chomsky Normal Form, in which all rules of a context-free grammar are of the form $S \longrightarrow \epsilon$, $X \longrightarrow YZ$, or $X \longrightarrow \mathtt{a}$, where S is the starting symbol, X, Y and Z may be any variable symbol, and a may be any alphabet symbol. Convert the following grammar to Chomsky Normal Form. You may use the general algorithm, or other transformations.

$$\begin{array}{lcl} S & \longrightarrow & \mathtt{abc}A \\ A & \longrightarrow & \mathtt{a}B\mathtt{a}B \\ B & \longrightarrow & C \mid \mathtt{b}CC \\ C & \longrightarrow & C\mathtt{c} \mid \epsilon \end{array}$$

[15 marks]

THE UNIVERSITY
of LIVERPOOL

6.

a. Define *pushdown automaton* (PDA), and explain what it means for a PDA to accept an input. [5 marks]

b. Describe a *deterministic* pushdown automaton that accepts the set of all palindromes over the alphabet $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ in which the only place the letter `c` occurs is at the centre of the word. (For example, words like `abcba`, `aabacabaa`.) [10 marks]

c. Explain why unrestricted palindromes cannot be recognised by deterministic pushdown automata (and one would have to use a nondeterministic PDA instead). [5 marks]

7.

a. Classify each of the following languages according to whether it is recursively enumerable, recursive, context-free or regular.

(i) words over the one-letter alphabet `a` whose length is a square number [2 marks]

(ii) words over the one-letter alphabet `a` whose length is either a multiple of 3 or a multiple of 4 [2 marks]

(iii) encodings of Turing machines that accept the empty string (An encoding of a Turing machine can be assumed to be any standard encoding that lists the states and transitions of the machine.) [2 marks]

b. Give a detailed description of a Turing machine M that performs a cyclic left shift of its input, assumed to be a string over the alphabet $\{\mathtt{0}, \mathtt{1}\}$. So, if the input is of length n, the machine should copy the first letter of the input to position n and copy each of the remaining letters one position to the left. M should halt after performing this operation. For an input of length n, how many transitions would be executed by your machine M, as a function of n? [14 marks]