PAPER CODE NO.	EXAMINER : Dr. Paul E. Dunne
COMP209	DEPARTMENT : Computer Science Tel. No. 43521

THE UNIVERSITY
of LIVERPOOL

JANUARY 2005 EXAMINATIONS

Bachelor of Science : Year 2

Decision, Computation and Language

TIME ALLOWED : 2 Hours

INSTRUCTIONS TO CANDIDATES

Answer **all** questions in Section 1 and **two** questions from Section 2.

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. Define what is meant by the following terms,

a. A **word** over an alphabet, Σ. **(3 marks)**

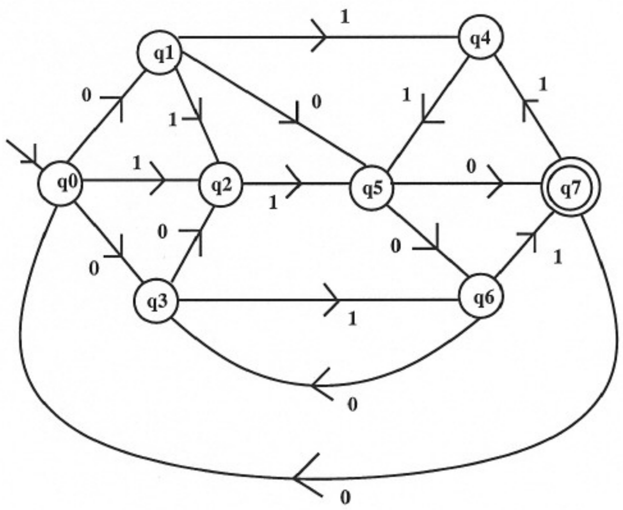
b. The language resulting from the **concatenation** of two **languages**. **(3 marks)**

c. The language **accepted by a non-deterministic finite automaton** $M = (Q, \Sigma, q_0, F, \delta)$. **(3 marks)**

d. A **formal grammar** $G = (V, T, S, P)$. **(3 marks)**

e. The Church-Turing Hypothesis. **(3 marks)**

2. For the **non-deterministic** finite automaton below,



a. Describe its **state transition function** – $\delta : Q \times \{0, 1\} \to \mathcal{P}(Q)$. **(8 marks)**

b. Give the **full computation tree** of this automaton when it is given the word **01101** as input, indicating whether this word is accepted or not. **(7 marks)**

3. For the **deterministic**, finite automaton $M = (Q, \{0,1\}, q_0, F, \delta)$ in which $Q = \{q_0, q_1, q_2, q_3, q_4\}$, $F = \{q_2, q_4\}$ and $\delta : Q \times \{0,1\} \to Q$ is given by

q	σ	$\delta(q,\sigma)$	q	σ	$\delta(q,\sigma)$
q_0	0	q_1	q_0	1	q_3
q_1	0	q_2	q_1	1	q_1
q_2	0	q_3	q_2	1	q_4
q_3	0	q_4	q_3	1	q_3
q_4	0	q_4	q_4	1	q_1

a. Construct a **right-linear grammar**, $G = \langle V, T, S, P \rangle$ such that G generates a word w if and only if w is accepted by M. **(9 marks)**

b. Give the sequence of productions that are applied in this grammar to generate the word **100110**. **(6 marks)**

4. Give examples of each of the following:

a. A **context–free** language that is **not** a **regular** language. **(5 marks)**

b. A **recursive** language that is **not** a **context–free** language. **(5 marks)**

c. A **recursively enumerable** language that is **not** a **recursive** language. **(5 marks)**

Full credit requires a formal definition of a language with the relevant properties (but not a proof).

THE UNIVERSITY
of LIVERPOOL

Section 2

Answer **two** questions in this section.

5. For the **context–free** grammar, $G = (V, \Sigma, S, P)$ with $V = \{A, B, C, S\}$; $\Sigma = \{a, b, c, d\}$; start symbol S and production rules, P, defined by

$$\begin{array}{lcl} S & \rightarrow & ASA \parallel BCC \parallel Cd \\ A & \rightarrow & BA \parallel SB \parallel bc \\ B & \rightarrow & CaB \parallel SC \parallel b \\ C & \rightarrow & AB \parallel SBCS \parallel c \end{array}$$

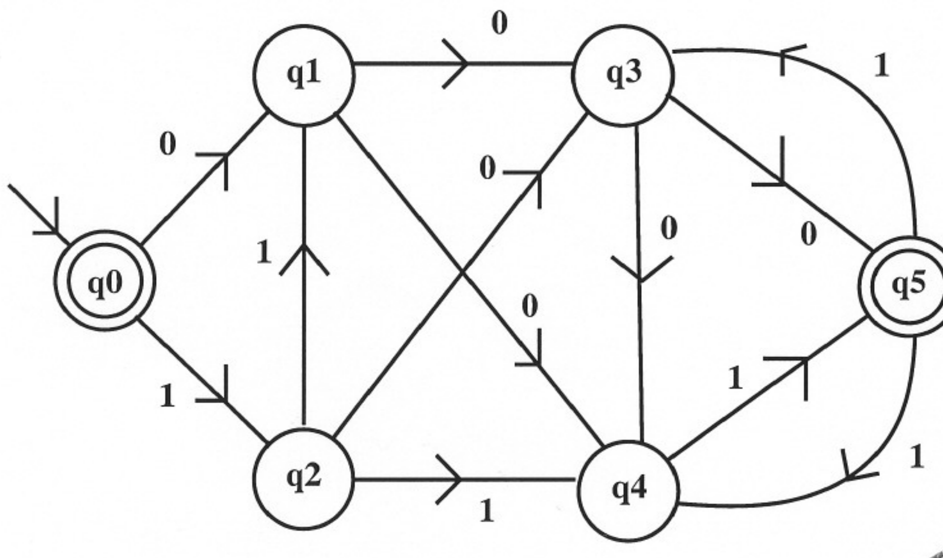
Maybe add "Justify your answer"?

a. Identify all of the production rules of G that do not meet the conditions of **Chomsky Normal Form**. (**6 marks**)

b. Carefully show how G may be modified to give a context–free grammar $G' = (V', \Sigma, S, P')$ such that G' generates exactly the same language as G and for which every production rule in P' is in Chomsky Normal Form. (**8 marks**)

c. The machine model corresponding to Context Free languages extends deterministic finite automata by allowing data to be stored on an unlimited capacity **stack**. Is the class of languages that can be recognised changed if, instead of a stack, a **double ended queue** is provided, i.e. a memory structure which permits access to its first and last element but not elements between? Briefly justify your answer. (**6 marks**)

6. For the **non-deterministic** finite automaton $M = (Q, \Sigma, q_0, F, \delta)$ below



no comma

a. Describe in detail how an equivalent **deterministic** finite automaton, is constructed from M. (**14 marks**)

b. Suppose $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$ is an alphabet of k symbols and $L \subseteq \Sigma^*$ is a language over Σ. Consider the language $bin(L)$ over the alphabet $\{0, 1\}$ defined as

$$bin(L) \quad = \quad \{\, 0^{i_1}10^{i_2}1 \cdots 10^{i_r} \; : \; \sigma_{i_1}\sigma_{i_2} \; \cdots \; \sigma_{i_r} \; \in \; L\}$$

If L is a **regular** language then is it the case that $bin(L)$ is also a regular language? Give a brief justification for your answer. (**6 marks**)

7.

a. **State** the **Pumping Lemma for Context–Free Languages**. (**5 marks**)

b. The language L_{GCD} over the alphabet $\{a, b\}$ is defined as

$L_{GCD} \; = \; \{\, a^n b^m \; : \; n, m \geq 1$ and the greatest common divisor of n and m is **greater than** $1\}$

e.g. the word *aaabb* is *not* in L_{GCD} but *aaaabb* is (since the greatest common divisor of 4 and 2 is 2).

Using the Pumping Lemma for **Regular Languages**, show that L_{GCD} is **not** a Regular language. (**9 marks**)

c. Proofs that some languages are undecidable often use constructions that assume particular Turing machine programs. It is, however, rarely the case that complete, formal descriptions of such Turing machine programs are presented, e.g. in terms of their state transition behaviour. Describe the rationale for regarding such detailed specifications as unnecessary in ensuring a proof is rigorous. (**6 marks**)