UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR08008 INFORMATICS 2A: PROCESSING FORMAL AND NATURAL LANGUAGES

Friday 14th December 2012

14:30 to 16:30

INSTRUCTIONS TO CANDIDATES

1. **Answer all five questions in Part A, and two out of three questions in Part B. Each question in Part A is worth 10% of the total exam mark; each question in Part B is worth 25%.**
2. **Use a single script book for all questions.**
3. **Calculators may be used in this exam.**

Convener: J Bradfield
External Examiner: A Preece

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

PART A
ANSWER ALL QUESTIONS IN PART A.

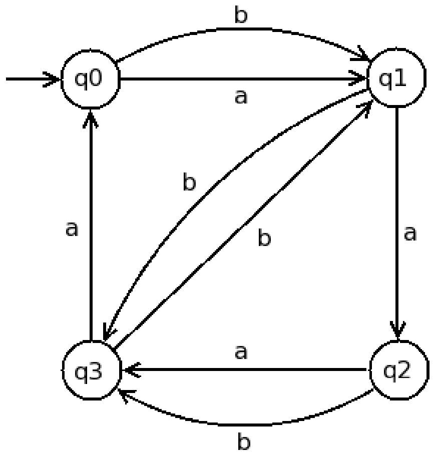
1. (a) List, in order of increasing expressive power, the levels of the *Chomsky hierarchy*. [*4 marks*]

   (b) Describe a class of (theoretically possible) English sentences whose structure *cannot* be captured by a grammar at the lowest level of the Chomsky hierarchy. Explain informally why it cannot. You may illustrate your class with sample sentences, though you need not give a formal grammar for it. [*5 marks*]

   (c) What is the lowest level of the Chomsky hierarchy that is believed to suffice for the description of syntactic structures in human languages generally? [*1 mark*]

2. This question concerns the operation of *minimization*, which constructs the smallest deterministic finite automaton (DFA) that recognises the same language as a given DFA.

   (a) Consider the picture below, which we shall use as a template to define DFAs, over the alphabet $\Sigma = \{a, b\}$, by specifying the set F of accepting states. (Note that the picture itself specifies the initial state as $q0$.)



   Draw the result of applying the minimization procedure to the DFA determined by each of the following choices for the set F of accepting states.

   i. $F = \{q0, q1, q2, q3\}$. [*2 marks*]

   ii. $F = \emptyset$, i.e., there are no accepting states. [*2 marks*]

   iii. $F = \{q1, q3\}$. [*4 marks*]

   (b) Briefly outline how minimization can be used to implement a *decision algorithm* to test whether two DFAs, M_1 and M_2, accept the same language. That is, the algorithm takes M_1 and M_2 as input, it outputs **yes** if $\mathcal{L}(M_1) = \mathcal{L}(M_2)$, and it outputs **no** if $\mathcal{L}(M_1) \neq \mathcal{L}(M_2)$. [*2 marks*]

3. Recall that the *Viterbi algorithm* identifies the most probable POS tagging for a given word sequence relative to a specified hidden Markov model. Use this algorithm to tag the word sequence

man bites dog

using transition and emission probabilities as follows:

	to N	to V
from start	0.6	0.4
from N	0.5	0.5
from V	0.8	0.2

Transitions

	man	bites	dog
N	0.5	0.2	0.3
V	0.4	0.4	0.2

Emissions

Show your working, and include explicit backtrace pointers in your Viterbi matrix. [*10 marks*]

4. Consider the following probabilistic context-free grammar for a simple language of questions:

S → Which NP VP ? (1.0)
NP → N (0.7) | A N (0.3)
VP → V N (0.9) | V like N (0.1)
N → orange (0.3) | flies (0.4) | bananas (0.3)
V → like (0.4) | flies (0.3) | throws (0.3)
A → orange (1.0)

(a) Draw all possible parse trees for the question

Which orange flies like bananas ?

and compute the probability for each parse tree. [*6 marks*]

(b) The above rules also generate some phrases that are ungrammatical in English, such as

Which bananas throws orange ?

Explain briefly how you would adapt the above grammar so that it generates only grammatically correct questions. You need not give full details of your solution, but should include two sample rules to illustrate your approach. [*4 marks*]

5. This question concerns *noncontracting grammars* as a method for defining context-sensitive languages.

   (a) What is the general format for productions in a noncontracting grammar over a set Σ of terminals and a set N of nonterminals? (You do not need to consider ϵ-productions in your answer. That is, you need only consider productions whose right-hand-side is nonempty.) [*2 marks*]

   (b) Consider the following noncontracting grammar, with terminals $\Sigma = \{\mathsf{0}, \mathsf{exp}, =\}$; nonterminals S (the start symbol), A and As; and productions

$$\begin{array}{rcl} \mathsf{S} & \longrightarrow & \mathsf{exp} = \mathsf{0}\ \mathsf{As} \\ \mathsf{As} & \longrightarrow & \mathsf{A} \mid \mathsf{A}\ \mathsf{As} \\ \mathsf{0}\ \mathsf{A} & \longrightarrow & \mathsf{A}\ \mathsf{0}\ \mathsf{0} \\ = \mathsf{A} & \longrightarrow & \mathsf{0} = \end{array}$$

   Give a full derivation of:

$$\mathsf{exp}\ \mathsf{0}\ \mathsf{0} = \mathsf{0}\ \mathsf{0}\ \mathsf{0}\ \mathsf{0}$$

   [*6 marks*]

   (c) Give a precise mathematical description of the language generated by the grammar in part (b). [*2 marks*]

PART B
ANSWER TWO QUESTIONS FROM PART B.

6. This question concerns two languages over the alphabet $\Sigma = \{1, –1\}$ (note that this is an alphabet with just two symbols: 1 and –1). The two symbols are interpreted, in the natural way, as the numbers 1 and -1, in order to define the languages, which are:

$$\begin{aligned} L_1 &= \{x \in \Sigma^* \mid \text{the sum of the numbers in } x \text{ is divisible by 3}\} \\ L_2 &= \{x \in \Sigma^* \mid \text{the sum of the numbers in } x \text{ is 0}\}\ . \end{aligned}$$

Thus, for example, the first two words below are in both L_1 and L_2, whereas the third and fourth are in L_1 but not in L_2.

ϵ 1 1–1 1 –1 –1 1 1 –1 1 1 –1 1 –1 –1 –1 –1 1

(a) The language L_1 is regular. Draw a deterministic finite automaton (DFA) that accepts L_1. [*4 marks*]

(b) Using your DFA, write out a system of simultaneous equations describing the language L_1, and solve these equations using Arden's Rule to produce a regular expression for L_1. [*10 marks*]

(c) The language L_2 is not regular. Prove this using the Pumping Lemma. [*6 marks*]

(d) The language L_2 is context free. Show this by constructing a pushdown automaton (PDA) that accepts L_2. (Hint: One approach is to use stacks of the form $1^n \perp$ and $(–1)^n \perp$ to record that the sum of the input string read so far is n and $-n$ respectively.) [*5 marks*]

7. In English, there are two common ways of presenting a list of items. One is to insert *and* between all adjacent items. The other is to insert a comma between adjacent items, except for the last two where *and* is inserted:

cows and pigs and goats and sheep
cows , *pigs* , *goats and sheep*

The following context-free grammar generates lists of both these kinds. The start symbol is L. (Here the non-terminal CAL, for example, stands for 'comma-and list'.)

$$\begin{array}{rcl} \mathsf{L} & \rightarrow & \mathsf{AL} \mid \mathsf{CAL} \\ \mathsf{AL} & \rightarrow & \mathsf{I} \mid \mathsf{I}\ \textit{and}\ \mathsf{AL} \\ \mathsf{CAL} & \rightarrow & \mathsf{CL}\ \textit{and}\ \mathsf{I} \\ \mathsf{CL} & \rightarrow & \mathsf{I} \mid \mathsf{I}\ ,\ \mathsf{CL} \\ \mathsf{I} & \rightarrow & \textit{cows} \mid \textit{pigs} \mid \textit{goats} \mid \textit{sheep} \mid \cdots \end{array}$$

(a) Even though our grammar is not in Chomsky Normal Form, it is possible to construct CYK parse charts for it. Draw up and fill out a CYK parse chart for the following phrase as a 5 × 5 matrix:

cows , *goats and sheep*

Take care to include all possible entries in the chart, not just those that contribute to some overall parse of the phrase. (You need not include pointers or other information to show how non-terminals are broken into its immediate sub-constituents.) [*6 marks*]

(b) Is the above context-free grammar *ambiguous*? Justify your answer. [*2 marks*]

For the remainder of this question, we may discard the lexical rules for I, and simply treat I as a terminal symbol along with , and *and*.

(c) Design an LL(1) grammar that is equivalent to the one above. There is more than one way to do this, but your solution should make use of a non-terminal CTail for which the generating rules are as follows:

$$\mathsf{CTail} \rightarrow \epsilon \mid ,\ \mathsf{I}\ \mathsf{CTail}$$

You may also introduce other new non-terminals if they are helpful. [*5 marks*]

(d) Write down the First and Follow sets for each of the non-terminals in your LL(1) grammar. You need not show your working. [*4 marks*]

(e) Draw up the LL(1) parse table for your grammar. [*5 marks*]

(f) Briefly discuss whether LL(1) grammars are in general an appropriate technology for natural language processing. [*3 marks*]

8. In this question, we will work out a semantics of a language that describes configurations of solid objects of various shapes and colours, some of which may be touching other objects. The grammar for our language is as follows:

$$\begin{array}{rl} \mathsf{S} \rightarrow & \text{There is a } \mathsf{NP} \\ \mathsf{NP} \rightarrow & \mathsf{N} \;\mid\; \mathsf{ANP} \;\mid\; \mathsf{ANP} \text{ that } \mathsf{VP} \\ \mathsf{ANP} \rightarrow & \mathsf{AP}\ \mathsf{N} \\ \mathsf{AP} \rightarrow & \mathsf{A} \;\mid\; \mathsf{A} \text{ or } \mathsf{A} \\ \mathsf{VP} \rightarrow & \text{touches every } \mathsf{NP} \;\mid\; \text{touches some } \mathsf{NP} \\ \mathsf{N} \rightarrow & \text{sphere} \;\mid\; \text{cube} \\ \mathsf{A} \rightarrow & \text{red} \;\mid\; \text{blue} \;\mid\; \text{green} \end{array}$$

For example, this generates the sentence

There is a red cube that touches every blue or green sphere

Our goal is to interpret this language in a first order logic in which variables range over objects. Our logic is equipped with unary predicates *sphere*(x), *cube*(x), *red*(x), *blue*(x), *green*(x), and a binary predicate *touches*(x,y).

(a) Supply a suitable semantic rule for each of the above grammar rules, in order to generate an appropriate lambda expression for every sentence of the language. For example, the semantics for the first grammar rule should be

$$\mathsf{S} \rightarrow \text{There is a } \mathsf{NP} \qquad \{\ \exists \text{x}.\ \mathsf{NP.Sem}(\text{x})\ \}$$

You may annotate the second rule for AP as '$\mathsf{AP} \rightarrow \mathsf{A}_1$ or A_2' to enable you to distinguish the two occurrences of A.

For all phrase categories apart from S, phrases should be interpreted as lambda expressions of type $< o, t >$, where o is the type of objects and t the type of truth values; no higher types are necessary. The rules involving 'touches' will require the most attention. [*12 marks*]

(b) Write down the formula of first order logic that your semantics gives for the sentence:

There is a red cube that touches every blue or green sphere

(This formula will be obtained by β-reducing a certain lambda expression given by your semantic rules, but you are not required to exhibit the lambda expression or the β-reductions explicitly.) [*5 marks*]

QUESTION CONTINUES ON NEXT PAGE

QUESTION CONTINUED FROM PREVIOUS PAGE

(c) As a step towards making our grammar more modular, we decide to replace the productions for VP by the following productions involving the new non-terminal QNP for *quantified* noun phrases:

$$\begin{aligned} \mathsf{VP} \rightarrow &\quad \text{touches } \mathsf{QNP} \\ \mathsf{QNP} \rightarrow &\quad \text{every } \mathsf{NP} \ \mid \ \text{some } \mathsf{NP} \end{aligned}$$

Provide semantic rules for these three productions. Phrases of category QNP should be interpreted as lambda expressions of type $<< o, t >, t >$. [*4 marks*]

(d) Write down the raw lambda expression associated by your new semantics with the phrase

touches some cube

Also write down the normal form to which this reduces (you need not show the individual β-reduction steps explicitly). [*4 marks*]