UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFORMATICS 2A: PROCESSING FORMAL AND NATURAL LANGUAGES

Thursday 15th December 2011

09:30 to 11:30

Convener: J Bradfield
External Examiner: A Preece

INSTRUCTIONS TO CANDIDATES

1. **Answer Parts A and B. The multiple choice questions in Part A are worth 50% in total and are each worth the same amount. Mark one answer only for each question — multiple answers will score 0. Marks will not be deducted for incorrect answers. Part B contains THREE questions. Answer any TWO. Each is worth 25%.**
2. **Use the special mark sheet for Part A. Use a separate script book for each of the two questions from Part B that you answer.**

CALCULATORS MAY BE USED IN THIS EXAMINATION.

PART A
ANSWER ALL QUESTIONS IN PART A. Use the special mark sheet.

1. Which of the following strings does *not* match the regular expression written in machine syntax as `(0*1?2+)*`

   (a) ϵ

   (b) `02`

   (c) `12`

   (d) `01201`

   (e) `12222`

2. All but one of the following regular expressions (in mathematical notation) define the same language. Which one does not define the same language as the other four?

   (a) $((a+b)c^*)^*$

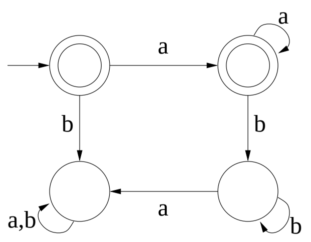
   (b) $(ac+bc)^*$

   (c) $(ac+(bc)^*)^*$

   (d) $\epsilon+(a+b)(ca+cb)^*c$

   (e) $(\epsilon+(b+a)c)^*$

3. Suppose we apply *minimization* to the following DFA over $\{a,b\}$:



   Which of the following correctly describes the resulting DFA M?

   (a) M has just a single state.

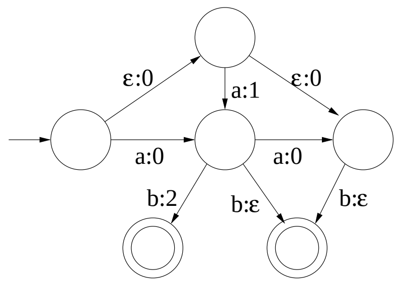
   (b) M has 2 states, 1 of which is accepting.

   (c) M has 3 states, 1 of which is accepting.

   (d) M has 3 states, 2 of which are accepting.

   (e) M has 4 states, 2 of which are accepting.

4. Consider the following non-deterministic finite state transducer with input alphabet $\{a, b, c\}$ and output alphabet $\{0, 1, 2\}$:



Which of the following is *not* a possible output string that could arise from processing the input string ab

(a) 0
(b) 00
(c) 01
(d) 02
(e) 012

5. Suppose a computer language has lexical classes Ident and Num defined by the following regular expressions in machine syntax:

Ident = `[a-z]+[0-9]?`　　Num = `0|([1-9][0-9]*)`

How would the input string `item90123` be split into lexical tokens according to the *principle of longest match*?

(a) `item90123`
(b) `item  90123`
(c) `item9  0123`
(d) `item9  0  123`
(e) None of the above.

6. Which one of the following context-free grammars is *unambiguous*? (Note that $a, b, c, (,), +$ are terminals, S, X, Y are nonterminals, and the start symbol in each case is S.)

   (a) $S \to aX \mid Yc$, $X \to bc$, $Y \to ab$

   (b) $S \to \epsilon \mid () \mid (S)$

   (c) $S \to \epsilon \mid (S) \mid S\,S$

   (d) $S \to \epsilon \mid (S) \mid S$

   (e) $S \to a \mid (S) \mid S + S$

7. Which of the following strings over $\{a, b\}$ is *not* accepted by the LL(1) parsing algorithm with the following parse table? The start symbol is S.

	a	b	c	$\$$
S	Tc	ϵ	Tc	ϵ
T	aSb		ϵ	

   (a) ϵ

   (b) ab

   (c) abc

   (d) $acbc$

   (e) c

8. Consider the following context-free grammar (where terminals are in lowercase, nonterminals are in uppercase, and the start symbol is A).

$$A \to CAB \mid a \mid \epsilon \qquad B \to bA \qquad C \to cA \mid \epsilon$$

   Which of the following sets is $\mathsf{First}(A)$?

   (a) $\{a, \epsilon\}$

   (b) $\{a, c\}$

   (c) $\{a, c, \epsilon\}$

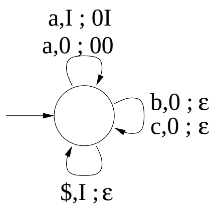
   (d) $\{a, b, c\}$

   (e) $\{a, b, c, \epsilon\}$

9. Which one of the following statements about LL(1) grammars is *true*?

   (a) Every unambiguous grammar is LL(1).
   (b) Every grammar in Chomsky normal form is LL(1).
   (c) Every regular grammar is LL(1).
   (d) In an LL(1) grammar, all productions must be left recursive.
   (e) In the worst case, the LL(1) parsing algorithm may require space proportional to the length of the input string.

10. Consider the following non-deterministic pushdown automaton with input alphabet $\{a, b, c\}$ and stack alphabet $\{I, 0\}$, where I is the initial stack symbol:



   Which of the following strings is *not* accepted on empty stack by this automaton?

   (a) ϵ
   (b) *aabc*
   (c) *abba*
   (d) *acab*
   (e) *aababb*

11. Consider two possible *semantics* for the language of regular expressions: Semantics A, which associates an NFA to each expression by means of a standard construction; and Semantics B, which associates to each expression the language it defines.

   Which one of the following statements about these semantics is *false*?

   (a) Both of these semantics may be defined *compositionally.*
   (b) Different regular expressions may receive the same interpretation under Semantics B.
   (c) Semantics B may be obtained from Semantics A by means of a standard construction.
   (d) Semantics A is more readily *executable* than Semantics B.
   (e) Semantics A is *more abstract* than Semantics B.

12. Which of the following statements about Turing machines is *false*?

   (a) For every context-sensitive language L, there is a Turing machine that accepts precisely the strings of L.

   (b) For any grammar G with set of terminals Σ, there is a Turing machine that accepts precisely the strings in Σ^* that *cannot* be derived from G.

   (c) There is a Turing machine which, given encodings of two DFAs over the same alphabet Σ, can tell whether or not they define the same language.

   (d) There is a Turing machine A which can simulate the behaviour of any given Turing machine B on any given finite input.

   (e) Any mathematical operation on natural numbers that can be implemented in Java can also be implemented by some Turing machine.

13. What will the following Python code print?

```
str = 'purple alice-b@google.com monkey dishwasher'
  match = re.search('([\w.-]+)@([\w.-]+)', str)
  if match:
    print match.group()
```

   (a) `google.com monkey dishwasher`

   (b) `google.com`

   (c) `alice-b`

   (d) `purple alice-b@google.com monkey dishwasher`

   (e) `alice-b@google.com`

14. What will the following Python code print?

```
list = ['students', 'love', 'inf2a']
  list.append('not')
  list.insert(0, 'the')
  print list
  print list.index('love')
```

   (a) `[0, 1, 2, 3, 4, 5]` and `love`

   (b) `['the', 'students', 'love', 'inf2a', 'not']` and 2

   (c) `['the', 'students', 'love']` and `['inf2a']`

   (d) it will print nothing

   (e) `['love', 'inf2a', 'not']` and `['not']`

15. What is dynamic programming?
    (a) A depth-first predictive parsing algorithm.
    (b) A method for solving complex problems by breaking them down into simpler subproblems.
    (c) A class of memory-efficient data structures in Python.
    (d) A method for brute force enumeration of all possible solutions.
    (e) A method for creating a frequency table.

16. In the following sentence, what is the correct part of speech (POS) sequence for the words *by some master photographer* (using the Penn Treebank tag set)?

    That/DT cold/JJ empty/JJ sky/NN was/VBD full/JJ of/IN fire/NN and/CC light/NN. It/PP seemed/VBD almost/RB a/DT magnification/NN of/IN the/DT Galaxy/NN itself/PP, of/IN the/DT Milky/NNP Way/NP blown/VBN up/RP by/?? some/?? master/?? photographer/??.

    (a) IN DT NN NN
    (b) IN JJ NN NNS
    (c) IN DT JJ NNP
    (d) IN DT JJ NN
    (e) PRP$ DT JJ NNS

17. Which of the following statements does *not* follow from assuming Zipf's law for natural languages?
    (a) The frequency of any word is inversely proportional to its rank.
    (b) There is a small number of very common words.
    (c) There is a small number of open and closed class words.
    (d) There is a large number of words that are infrequent.
    (e) There is a small-medium number of middle frequency words.

18. Consider the following probabilistic context-free grammar:

$$\begin{array}{rcll} S & \rightarrow & N \;\; VP & (1.0) \\ VP & \rightarrow & IV & (0.8) \\ VP & \rightarrow & TV \;\; N & (0.2) \\ N & \rightarrow & \text{clocks} & (0.7) \\ N & \rightarrow & \text{tables} & (0.3) \\ IV & \rightarrow & \text{tick} & (0.7) \\ IV & \rightarrow & \text{fly} & (0.3) \\ TV & \rightarrow & \text{hate} & (0.7) \\ TV & \rightarrow & \text{chase} & (0.3) \end{array}$$

Which of the following sentences is assigned the *highest* probability by this grammar?

(a) clocks tick
(b) tables fly
(c) clocks fly
(d) tables chase clocks
(e) tables hate clocks

19. Suppose that the predicate $H(x, y)$ means "x hates y". Which of the following is **not** a possible representation of the meaning of *Everybody hates somebody*?

(a) $\forall x.\, \exists y.\, H(x, y)$
(b) $(\lambda P.\, \forall x.\, \exists y.\, P(x, y))(\lambda x \lambda y.\, H(x, y))$
(c) $(\lambda P.\, \forall y.\, \exists x.\, P(y, x))(\lambda x \lambda y.\, H(x, y))$
(d) $(\lambda P.\, \forall x.\, \exists y.\, P(y, x))(\lambda x \lambda y.\, H(y, x))$
(e) $(\lambda P.\, \forall x.\, \exists y.\, P(x, y))(\lambda x \lambda y.\, H(y, x))$

20. Which of the following sets of productions is *not* in Chomsky normal form?

$$\begin{aligned}
G1: \quad & S \to AB \\
& A \to AB|a \\
& B \to Ba|b \\
G2: \quad & S \to AB \\
& A \to AB|a \\
& B \to BA|b \\
G3: \quad & S \to A|B \\
& A \to AB|a \\
& B \to BA|b
\end{aligned}$$

(a) G1

(b) G2

(c) G3

(d) G1 and G3

(e) G2 and G3

PART B
ANSWER TWO QUESTIONS FROM PART B. Use a separate script book for each question.

1. A general *grammar* consists of a set Σ of terminal symbols, a set N of nonterminals, a start symbol $S \in N$, and a set P of *productions* of the form $\alpha \to \beta$, where $\alpha, \beta \in (N \cup \Sigma)^*$. By placing suitable restrictions on the kinds of productions permitted, we may obtain classes of grammars that correspond to the various levels in the *Chomsky hierarchy* of language families.

   (a) List the four levels of the Chomsky hierarchy in order of increasing expressivity. For each level, name some class of machines or automata that gives rise to the language family in question via the notion of 'accepting computation'. [*8 marks*]

   (b) For each level in the Chomsky hierarchy, specify a suitable restriction on the form of productions $\alpha \to \beta$ so that the resulting grammars give rise to precisely the language family in question. [*5 marks*]

   (c) For each of the following languages, identify the *lowest* level of the Chomsky hierarchy in which the language resides. (You need not prove here that it does not reside in any lower level.) In each case, if this level is one of the two lowest levels of the hierarchy, give a grammar of the appropriate form to show that the language does indeed inhabit this level.

      i. The language L_1 over the alphabet $\{(,), :\}$ whose strings consist of two well-matched bracket sequences separated by a colon, e.g. $(()) : ()()$.

      ii. The language L_2 over $\{a, b, c\}$ consisting of strings $a^n b^n c^n$ for $n \geq 0$.

      iii. The language L_3 over $\{a, b\}$ consisting of strings in which a and b alternate (that is, strings with no substrings aa or bb).

      iv. The language L_4 over $\{\bullet, +, =\}$ consisting of strings $\bullet^n + \bullet^m = \bullet^{n+m}$ where $n, m \geq 0$. (For example, $\bullet\bullet + \bullet = \bullet\bullet\bullet$ is a string of L_4.) [*8 marks*]

   (d) Give a proof that the language L_4 from part (c) does *not* inhabit the bottom level of the Chomsky hierarchy. [*4 marks*]

2. Consider the following words together and their parts of speech. PN stands for proper noun, V for verb, N for noun, and ADJ for adjective. Some words are unambiguous (e.g., *John*), whereas others can be attested with more than one part of speech (e.g., *orange*, *dates*, *likes*).

John	PN
likes	V
orange	N, ADJ
dates	V, N

In what follows you will use a Hidden Markov model tagger and the Viterbi algorithm to annotate the sentence *John likes orange dates.* In order to do this, you are also given two tables, the emission (or observation) probability table and the transition probability table.

	John	likes	orange	dates
PN	1	0	0	0
V	0	0.6	0	0.4
N	0	0.4	0.3	0.6
ADJ	0	0	0.7	0

Table 1: Emission probabilities

	PN	V	N	ADJ
<s>	0.4	0.4	0.2	0
PN	0.3	0.3	0.3	0.1
V	0.4	0.3	0.2	0.1
N	0.1	0.6	0.2	0.1
ADJ	0.2	0.1	0.4	0.3

Table 2: Transition probabilities

(a) Describe in your own words the Viterbi algorithm for HMM part of speech tagging. You can show the algorithm in pseudo-code, but it is not required. [*4 marks*]

(b) What is the equation you use to update the value of each cell in the probability matrix for the Viterbi algorithm? [*4 marks*]

(c) Create the full probability matrix for the Viterbi algorithm for the sentence *John likes orange dates* and fill in the cells. (You may leave multiplication expressions unevaluated if their value is not required in order for you to proceed.) What is the most probable part of speech sequence for the sentence? Assume that the probability at the start state is 1. [*12 marks*]

Question 2 continues on page 11

(d) The HMM tagging algorithm requires a dictionary that lists the possible parts-of-speech of every word. But even the largest dictionary will still not contain every possible word (e.g., proper names and acronyms are created very often, and new common nouns and verbs enter the language at a surprising rate). Is this a problem for the tagger? Can you think of a method for guessing the tag of an unknown word in the HMM modeling framework?

[*5 marks*]

3. Consider the following grammar G1:

R1	S	$\rightarrow$	NP VP
R2	S	$\rightarrow$	VP
R3	NP	$\rightarrow$	Det NP
R4	NP	$\rightarrow$	PN N
R5	NP	$\rightarrow$	PN
R6	VP	$\rightarrow$	V NP
R7	Det	$\rightarrow$	the
R8	N	$\rightarrow$	cook | toast
R9	VP	$\rightarrow$	cook
R10	V	$\rightarrow$	cook
R11	PN	$\rightarrow$	French

(a) Is grammar G1 in Chomsky Normal Form? Explain your answer and put the grammar into CNF before attempting parts (b)–(d). [*3 marks*]

(b) Give the final chart generated by the CYK algorithm when parsing the phrase *cook the French toast* with G1. Remember that the final chart contains all edges added during the parsing process. Represent your chart as a 4×4 grid. Take care to include *all* possible entries in the chart, not just those that contribute to some overall parse of the phrase.

How many possible parse trees are there for the whole phrase? Draw them all. [*10 marks*]

(c) Explain how you can use the CYK algorithm to parse probabilistic context-free grammars. Give a modification of the algorithm that uses the chart to keep track of the probabilities of partial analyses. Specify how partial analysis probabilities are computed. [*6 marks*]

(d) Use the probabilistic version of CYK to parse the phrase *cook the French toast* with G1. Assume that the rules in G1 are equiprobable, i.e., if there are n rules with the same left-hand side, then the probability of each rule is $1/n$. You may leave multiplication expressions unevaluated (e.g. you may write 0.5×0.5 rather than 0.25). [*6 marks*]