UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFORMATICS 2A: PROCESSING FORMAL AND NATURAL LANGUAGES

Wednesday 15 th December 2010

14:30 to 16:30

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.**

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

Notation: In this section of the paper we use the following notation:

- $\#_c(x)$ stands for the number of c symbols in the string x.
- x^R is the *reverse* of the string x, so if $x = a_1 \dots a_n$, $x^R = a_n \dots a_1$

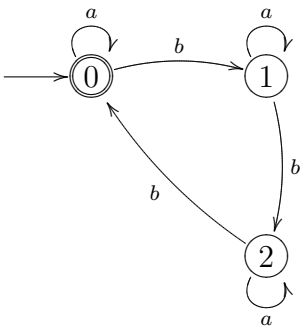
1. Someone asserts that the language $L = \{ww \mid w \in \{a, b\}^*\}$ is recognisable by a finite state machine with k states. You are in the process of demonstrating this is false using the pumping lemma. What would be a good choice of string to consider in using the pumping lemma to prove the assertion is false?

   (a) $aaaaaa$
   (b) $aaabbbbaabb$
   (c) $a^{2k}b$
   (d) $a^k b a^k b$
   (e) None of the above.

2. Which of the following is the dependency set for a string $x = a_1 \dots a_{2n}$ of length $2n$ in the language $L = \{ww \mid w \in \{a, b\}^*\}$?

   (a) $\{(i, 2n - i + 1) \mid 1 \leq i \leq n\}$
   (b) $\{(i, n - i + 1) \mid 1 \leq i \leq n\}$
   (c) $\{(i, n + i) \mid 1 \leq i \leq n\}$
   (d) $\emptyset$
   (e) None of the above.

3. What is the language recognised by the following FSA?



   (a) $\{x \in \{a,b\}^* \mid \#_a(x)$ and $\#_b(x)$ are both even$\}$
   (b) $\{x \in \{a,b\}^* \mid \#_a(x)$ is even$\}$
   (c) $\{x \in \{a,b\}^* \mid \#_a(x)$ is divisible by three$\}$
   (d) $\{x \in \{a,b\}^* \mid \#_b(x)$ is divisible by three$\}$
   (e) none of the above

4. Which of the following strings is a member of the language described by the regular expression $(a^*ba^*ba^*ba^*)^*$

   (a) *bbbb*
   (b) *bbaaabb*
   (c) *bbaaabbbabb*
   (d) *bbabbbab*
   (e) None of the above.

5. Someone has asserted that the following two regular expressions describe the same language: $R_1 = ((ab^*a) + (ba^*b))^*$ and $R_2 = ((ab^*a) + b^*)^*$. Which of the following strings is contained in one of the languages but not in the other?

   (a) *ababab*
   (b) *bbbbbb*
   (c) *abba*
   (d) *bbabba*
   (e) None of the above.

6. Which of the following context-free grammar productions describes the language which is a subset of $\{a\}^*$ in which all strings contain an odd number of a symbols *ambiguously*. In all cases, the start symbol is S and the alphabet is $\{a\}$.

   (a) $S \to a \mid aSa$

   (b) $S \to aaS \mid a$

   (c) $S \to Saa \mid a$

   (d) $S \to aA \mid Aa \mid a \quad A \to aS$

   (e) None of the above.

7. Which of the following descriptions best fits the language $L = \{a^n b^m c^n d^m \mid n, m \geq 0\}$?

   (a) L is a regular language

   (b) L is a context-free language that is not regular

   (c) L is a context-sensitive language that is not context-free

   (d) L is not a context-sensitive language

   (e) None of the above.

8. Consider the following context-free language: $L_1 = \{a^m b^n c^n \mid n, m \geq 0\}$. Which of the following choices of language L_2 is context-free and ensures that $L_1 \cap L_2$ is *not* a context-free language.

   (a) $L_2 = \{a^k b^{2k} c^m \mid k \geq 0 \text{ and } m \geq 0\}$

   (b) $L_2 = \{(abc)^k \mid k \geq 0\}$

   (c) $L_2 = \{a^k b^m c^k \mid k \geq 0 \text{ and } m \geq 0\}$

   (d) $L_2 = \{a^k b^{2k} c^{2k} \mid k \geq 0\}$

   (e) None of the above.

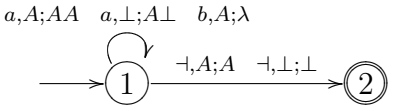
9. Consider the following context-free grammar:

$$G = (\{S, A, B\}, \{a, b, c, \dashv\}, \{S \to A \dashv \;\; A \to cA \mid BAa \mid b \;\; B \to b \mid \varepsilon\}, S)$$

Which of the following sets is $\text{First}_1(A)$?

(a) $\{a\}$
(b) $\{a, b\}$
(c) $\{a, b, \varepsilon\}$
(d) $\{a, b, c, \varepsilon\}$
(e) None of the above

10. What is the language recognised by the following PDA P? The stack alphabet of P is $\{A, B, \bot\}$ where $\bot$ is the initial stack symbol. The alphabet of P is $\{a, b, \dashv\}$ where $\dashv$ is used to mark the end of the input. Here, $\#_c(x)$ stands for the number of c symbols in x.



(a) $\{x \in \{a, b\}^* \dashv \mid \#_a(w) \geq \#_b(w)$ for every prefix w of $x\}$
(b) $\{x \in \{a, b\}^* \dashv \mid \#_a(w) > \#_b(w)$ for every prefix w of $x\}$
(c) $\{x \in \{a, b\}^* \dashv \mid \#_b(w) = 2\#_a(w)$ for every prefix w of $x\}$
(d) $\{x \in \{a, b\}^* \dashv \mid \#_b(w) \geq \#_a(w)$ for every prefix w of $x\}$
(e) none of the above

11. Which one of the following statements about word classes in natural languages is *false*?

(a) As a language develops, open classes acquire new words more frequently than closed classes do.
(b) Among languages worldwide, there is more variation in the inventory of open classes than of closed ones.
(c) Prepositions in English form a closed class.
(d) Closed classes often consist of relatively short words with some grammatical function.
(e) An ambiguous word in English *may* belong to both an open and a closed class.

12. Which type of part-of-speech tagger suffers from the disadvantage that it needs to be trained on a very large data set in order to work effectively?

    (a) A default tagger
    (b) A regular expression tagger
    (c) A unigram tagger
    (d) A bigram tagger
    (e) A rule-based tagger

13. The following regular expression is proposed for the purpose of recognizing *adjectives* in English:

    ```
    ^.*(a|e|i|o|u).*(ish|ic|al|ous|ble)$
    ```

    Which of the following words is *not* admitted by this regular expression?

    (a) uffish
    (b) hymnic
    (c) ancestral
    (d) pious
    (e) table

14. Which of the following sets of rules does *not* give rise to infinitely long derivations with start symbol S?

    (a) $S \to \epsilon, \quad S \to aSb$
    (b) $S \to aT, \quad T \to bS$
    (c) $S \to aT, \quad T \to ba, \quad U \to bU$
    (d) $S \to S$
    (e) $S \to aU, \quad T \to bU, \quad U \to cT$

15. *Linear indexed grammars* are intermediate in expressive power between

    (a) regular and LL(1) grammars
    (b) LL(1) and context free grammars
    (c) context free and context sensitive grammars
    (d) context sensitive and general (Type 0) grammars
    (e) none of the above

16. Which of the following statements about parsing algorithms is *incorrect*?

    (a) Recursive descent parsing is top-down and depth-first.

    (b) LL(1) parsing can be applied to any context-free grammar in Chomsky normal form.

    (c) The CYK algorithm is a bottom-up chart parsing algorithm.

    (d) The Earley algorithm is a bottom-up chart parsing algorithm.

    (e) The Earley algorithm uses top-down prediction to avoid building unnecessary structure.

17. Consider the following probabilistic context-free grammar:

S	$\rightarrow$	$N\ \ VP$	(1.0)
VP	$\rightarrow$	IV	(0.2)
VP	$\rightarrow$	$TV\ \ N$	(0.8)
N	$\rightarrow$	mice	(0.5)
N	$\rightarrow$	owls	(0.3)
N	$\rightarrow$	badgers	(0.2)
IV	$\rightarrow$	sleep	(0.7)
IV	$\rightarrow$	fly	(0.3)
TV	$\rightarrow$	like	(0.7)
TV	$\rightarrow$	hunt	(0.3)

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

    (a) badgers sleep

    (b) owls fly

    (c) mice fly

    (d) owls hunt mice

    (e) owls like badgers

18. What is the correct meaning representation for *If everyone takes part, everyone will be happy*?

    (a) $(\exists x.\text{TakesPart}(x)) \Rightarrow (\forall x.\text{WillBeHappy}(x))$

    (b) $(\exists x.\text{TakesPart}(x)) \Rightarrow (\exists x.\text{WillBeHappy}(x))$

    (c) $(\exists x.\text{TakesPart}(x)) \Rightarrow (\exists x.\text{WillBeHappy}(x))$

    (d) $\forall x.(\text{TakesPart}(x) \Rightarrow \text{WillBeHappy}(x))$

    (e) $(\forall x.\text{TakesPart}(x)) \Rightarrow (\forall x.\text{WillBeHappy}(x))$

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

    (a) $\forall x.\, \exists y.\, L(x, y)$

    (b) $(\lambda P.\, \forall x.\, \exists y.\, P(x, y))(\lambda x \lambda y.\, L(x, y))$

    (c) $(\lambda P.\, \forall x.\, \exists y.\, P(x, y))(\lambda x \lambda y.\, L(y, x))$

    (d) $(\lambda P.\, \forall y.\, \exists x.\, P(y, x))(\lambda x \lambda y.\, L(x, y))$

    (e) $(\lambda P.\, \forall x.\, \exists y.\, P(y, x))(\lambda x \lambda y.\, L(y, x))$

20. Which of the following statements about formal languages is *false*?

    (a) Every recursive language is recursively enumerable.

    (b) The complement of a recursive language is recursive.

    (c) The complement of a recursively enumerable language is recursively enumerable.

    (d) The union of two recursive languages is recursive.

    (e) The union of two recursively enumerable languages is recursively enumerable.

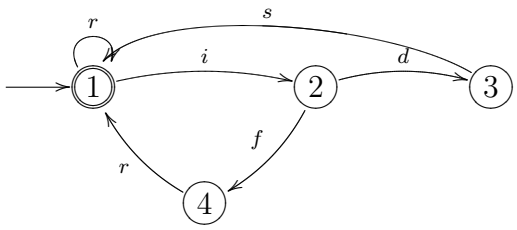
PART B
ANSWER TWO QUESTIONS FROM PART B. Use a separate script book for each question.

1. In this question you are asked to model a simple control system that is intended to ignite a heating system safely.

   (a) The ignition system can do the folllowing actions:

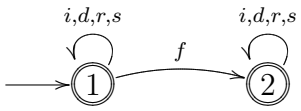
   - i – attempt to ignite the heater
   - d – detect that the heater has ignited correctly
   - f – fail to ignite the heater
   - r – reset after failure - the system awaits a reset before attempting to ignite again
   - s – switch off after the correct temperature is achieved

   The machine M_1 that models the ignition control system is:



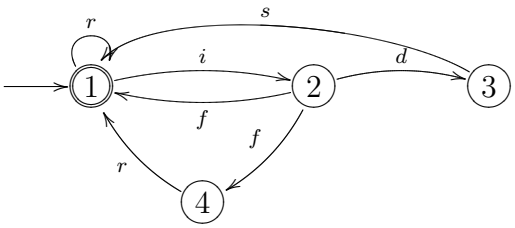
   The designer of the system wants to check it is correct and thinks that having a regular expression for the language $L(M_1)$ might be helpful. By writing down an equation for each state and solving them using the technique used in Kleene's theorem, find a regular expression for the language recognised by M_1. [*6 marks*]

   (b) After some experience in the field the designer decides that if the system fails to ignite twice then it should stop working to await maintenance. The machine that allows any action when one or fewer fails have been observed is M_2:



   Combine M_2 with the original model, M_1, of the ignition system using the intersection operation for Finite State Machines to construct a new version of the machine that stops working once it has seen two ignition failures. [*5 marks*]

(c) After further experience in the field it becomes evident that the system sometimes does not await a reset. The designer's model M_3 of this behaviour is the following:



Unfortunately M_3 is a *nondeterministic* finite automaton. Use the standard construction to find an equivalent *deterministic* finite automaton. [*6 marks*]

(d) The designer is now worried that perhaps the control system has other failures. To help monitor the behaviour of the system you are required to build a machine that checks to see that for every attempted ignition the system either fails or detects a correct ignition. The designer asks you to build a machine that recognizes:

$$L_4 = \{w \in \{i, d, s, f, r\}^* \mid \#_i(w) = \#_d(w) + \#_f(w)\}$$

The designer claims to have a finite state machine M_5 such that $L(M_5) = L_4$ and M_5 has k states. Do you disagree with the designer's claim? If you do disagree, provide notes on how you would go about convincing the designer the claim is false. If you agree with the designer provide a convincing argument the machine can be constructed. [*4 marks*]

(e) The designer decides to build a monitor machine M_6 that will recognise L_4. Can you construct a machine M_6 such that:

$$L(M_6) = L_4$$

[*4 marks*]

2. Consider the following grammar for noun phrases, in which the start symbol is NP, and the lowercase English words are terminals:

$$
\begin{array}{rcl}
\mathsf{NP} & \rightarrow & \mathsf{Det}\ \ \mathsf{Nom} \\
\mathsf{Nom} & \rightarrow & \mathsf{N} \mid \mathsf{AP}\ \ \mathsf{Nom} \\
\mathsf{AP} & \rightarrow & \mathsf{Adj} \mid \mathsf{AdvD}\ \ \mathsf{Adj} \\
\mathsf{N} & \rightarrow & \text{book} \mid \text{orange} \\
\mathsf{Adj} & \rightarrow & \text{heavy} \mid \text{orange} \\
\mathsf{Det} & \rightarrow & \text{a} \\
\mathsf{AdvD} & \rightarrow & \text{very}
\end{array}
$$

Here the symbols N, Adj, Det, AdvD represent various parts of speech: respectively, *nouns*, *adjectives*, *determiners* and *adverbs of degree*. Note in particular the part-of-speech ambiguity of 'orange'.

In this question, we shall apply several parsing algorithms to the noun phrase:

a very heavy orange book

(a) Draw up and fill out a complete CYK parsing chart for this phrase as a 5x5 grid. 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 each non-terminal is broken into its immediate sub-constituents.)

How many possible parse trees are there for the whole phrase? Draw them all. [*8 marks*]

(b) Recall from the lectures that a *bottom-up active parsing strategy* for a phrase of length n works as follows (we summarize it here in terms of building up a graph whose arcs are labelled with *dotted rules*).

i. Starting from a row of $n+1$ nodes, use the *bottom-up initialization rule* to add an arc for every word in the phrase.

ii. Apply the *bottom-up predict rule* wherever possible.

iii. Apply the *fundamental rule* of active chart parsers wherever possible.

iv. Repeat steps 2 and 3 until no new edges are added.

Construct the labelled graph obtained by applying this process to the phrase 'a very heavy orange book' with the above grammar. This time, you should only include arcs and labelled that contribute to some overall parse, drawing on your experience from part (a). To reduce clutter, you should omit all *self-loop* arcs. You may also label the same arc with as many dotted rules as necessary. [*8 marks*]

(c) Suppose now we delete the rule '$\mathsf{N} \rightarrow$ orange' from the grammar, thus resolving the POS-ambiguity in favour of Adj. Somewhat unusually for a Natural Language example, the remaining rules then constitute an *LL(1) grammar*. Draw up the LL(1) parse table for this grammar. (Formally, this can be done by computing First sets, but you may find it easier simply to fill out the table by inspection of the grammar. Note that Follow sets are not relevant since no non-terminal can expand to the empty string.) [*6 marks*]

(d) Now suppose we reinstate the rule '$\mathsf{N} \rightarrow$ orange', and again attempt to parse the phrase 'a very heavy orange book' using LL(1) parsing. Carefully explain what exactly goes wrong, and at what stage in the process it does so. [*3 marks*]

3. In this question we shall consider both the syntax and semantics of a language for making statements about particular named people. Our starting point is the language of *simple sentences* (SS) as defined by the following grammar. Note that the verbs in question are *intransitive*, *transitive* or *ditransitive*, according to whether they take 0, 1 or 2 arguments (not counting their subject). We also include a class of *common nouns* (CN) for later use.

SS	→	Name VP
VP	→	IV | TV Name | DV Name to Name
Name	→	John | Mary | Peter | Susan
IV	→	slept | walked
TV	→	met | saw
DV	→	introduced | sent
CN	→	child | man | woman

(a) For each of the three verb types, write down an example of a sentence generated by the above grammar containing a verb of that type. [*3 marks*]

(b) In English, a common noun may be modified by a *relative clause*. The following are examples of noun phrases involving relative clauses (the relative clause is underlined in each case).

(a) a man <u>who slept</u>
(b) a woman <u>who saw Peter</u>
(c) a man <u>who introduced Mary to Susan</u>
(d) a woman <u>Susan met</u>
(e) a child <u>Mary sent John to</u>

The relative clauses here are of three kinds. In (a)-(c), it is the *subject* of the clause that has been relativized (<u>a man</u> slept, etc.). In (d), the *direct object* has been relativized (Susan met <u>a woman</u>). In (e), the *indirect object* has been relativized (Mary sent John to <u>a child</u>).

Write a set of context-free rules which, in combination with those given above, yield a grammar for a phrase type NP of noun phrases. Here, a *noun phrase* should consist of a common noun preceded by the determiner 'a' and *optionally* followed by a single relative clause. Your rules should generate all the examples listed above, and others with the same structure. It should not generate phrases that are obviously ungrammatical in English. To cover the above phrases, you will only need a single rule for each of the three kinds of relative clause. (Note: your grammar is *not* required to generate the phrase 'the man Mary sent to Susan'.) [*6 marks*]

We may now obtain a grammar for *complex sentences* (CS) by adding the single rule:

$$\mathsf{CS} \;\rightarrow\; \mathsf{Name}\ \text{is}\ \mathsf{NP}$$

We next build up a compositional semantics for our language using predicate logic and lambda expressions. We attach a semantic valuation function to the grammar rules for simple sentences as follows (the clauses for the remaining terminals are similar). Four of the semantic clauses in this definition have been intentionally left blank.

Name	→	John	{ john }
IV	→	slept	{ λx. slept(x) }
TV	→	saw	
DV	→	sent	
VP	→	IV	{ IV.Sem }
VP	→	TV Name	
VP	→	DV Name_1 to Name_2	
SS	→	Name VP	{ VP.Sem (Name.Sem) }

(c) Complete the definition by attaching a suitable semantics to the remaining four rules. You may assume the logic contains predicates saw(x,y) and sent(x,y,z), meaning respectively 'x saw y' and 'x sent y to z'. [*4 marks*]

(d) For the sentence with a ditransitive verb that you gave in part (a), write down the raw lambda expression assigned to it by the above definition. Then show how this lambda expression can be reduced, one β-step at a time, to a much simpler formula. [*4 marks*]

(e) Your final task is to extend the above semantics to noun phrases as defined by your rules. The semantics of a noun phrase (NP) should be a unary predicate — informally, one which is true of exactly those individuals who fit the description given by the phrase. Write out the rules you gave in part (b), attaching suitable semantic valuations in the style above. You may assume the logic contains unary predicates such as man(x). [*8 marks*]

It is possible to verify your answer to part (e) in the following way (though you are not required to do so for full credit). The semantics for complex sentences may be completed as follows:

$$\mathsf{CS} \;\rightarrow\; \mathsf{Name}\ \text{is}\ \mathsf{NP} \quad \{\ \mathsf{NP}.\text{Sem}\ (\mathsf{Name}.\text{Sem})\ \}$$

You may now compute the semantics for a sentence such as 'John is a man whom Susan met' and check that it β-reduces to the formula man(john) $\wedge$ met(susan,john).