

UNIVERSITY OF EDINBURGH
COLLEGE OF SCIENCE AND ENGINEERING
SCHOOL OF INFORMATICS

**INFORMATICS 2A: PROCESSING FORMAL AND NATURAL
LANGUAGES**

Thursday 25th August 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.**

PART A

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

1. What is a correct first order logic representation of the sentence:

Every student likes some course that he or she is taking.

- (a) $\forall X. \exists Y. \text{student}(X) \wedge \text{course}(Y) \wedge \text{takes}(X, Y) \Rightarrow \text{likes}(X, Y)$
 - (b) $\forall X. \forall Y. \text{student}(X) \wedge \text{course}(Y) \wedge \text{takes}(X, Y) \Rightarrow \text{likes}(X, Y)$
 - (c) $\exists Y. \forall X. \text{student}(X) \wedge \text{course}(Y) \wedge \text{takes}(X, Y) \Rightarrow \text{likes}(X, Y)$
 - (d) $\forall X. \text{student}(X) \Rightarrow \exists Y. \text{course}(Y) \wedge \text{takes}(X, Y) \wedge \text{likes}(X, Y)$
 - (e) None of the above.
2. Which of the following characteristics does *not* apply to rule-based part-of-speech tagging?
- (a) A rule-based tagger requires a list of possible POS tags for each word.
 - (b) A rule-based tagger requires a large number of disambiguation rules if it is to determine POS tags effectively.
 - (c) The disambiguation rules typically make reference to words or POS tags occurring in the neighbourhood of the word to be tagged.
 - (d) The disambiguation rules have to be written manually.
 - (e) Rule-based tagging can be combined with a probabilistic approach.
3. Which of the following properties holds for probabilistic context-free grammars?
- (a) The sum of the probabilities of all rules in the grammar has to be one.
 - (b) The sum of the probabilities of all rules with the same right-hand side has to be one.
 - (c) The probability of a given parse of a sentence is the sum of the probabilities of all the rules used to derive this parse.
 - (d) The probability assigned to a sentence is the sum of the probabilities of its possible parses.
 - (e) The probability assigned to a sentence is the product of the probabilities of its possible parses.

4. Which of the following lambda expressions has the same truth value as

$$(\lambda P.\lambda Q.\exists x.(P(x) \wedge Q(x)))(\text{dog})(\lambda z.\text{sneeze}(z))$$

- (a) $(\lambda P.\lambda Q.\exists x.(P(x) \wedge Q(x)))(\text{sneeze}(\text{dog}))$
- (b) $\text{sneeze}(\text{dog})$
- (c) $\exists x.(\text{dog}(x) \wedge \text{sneeze}(x))$
- (d) $\exists x.(\text{dog}(x) \wedge (\lambda z.\text{sneeze}(z)))$
- (e) None of the above.

5. Which of the following pairs of rules involves indirect recursion?

- (a) $\text{NP} \rightarrow \text{Det Adj N}, \quad \text{NP} \rightarrow \text{Pron}$
- (b) $\text{VP} \rightarrow \text{V NP}, \quad \text{NP} \rightarrow \text{Det N that VP}$
- (c) $\text{N}' \rightarrow \text{N N}', \quad \text{NP} \rightarrow \text{Det N}'$
- (d) $\text{S} \rightarrow \text{AdvP S}, \quad \text{AdvP} \rightarrow \text{Adj Adv}$
- (e) $\text{VP} \rightarrow \text{V NP}, \quad \text{NP} \rightarrow \text{Det N PP}$

6. Which one of the following grammars is LL(1)?

- (a) $\text{S} \rightarrow \text{TU}, \quad \text{T} \rightarrow \epsilon \mid \text{aT}, \quad \text{U} \rightarrow \epsilon \mid \text{bS}$
- (b) $\text{S} \rightarrow \text{Ta} \mid \text{Tb}, \quad \text{T} \rightarrow \epsilon \mid \text{c}$
- (c) $\text{S} \rightarrow \epsilon \mid \text{ST}, \quad \text{T} \rightarrow \text{a} \mid \text{b}$
- (d) $\text{S} \rightarrow \epsilon \mid \text{TU}, \quad \text{T} \rightarrow \epsilon \mid \text{a}, \quad \text{U} \rightarrow \epsilon \mid \text{b}$
- (e) $\text{S} \rightarrow \text{Ta} \mid \text{Ub}, \quad \text{T} \rightarrow \text{c}, \quad \text{U} \rightarrow \text{c}$

7. Which of the following strings *cannot* be derived from the symbol S using the rules $\text{S} \rightarrow \text{SS} \mid \text{aaa} \mid \text{aaaaa}$?

- (a) aaaaa
- (b) aaaaaa
- (c) aaaaaaa
- (d) aaaaaaaaa
- (e) aaaaaaaaaa

8. Which of the following describes the language $\{a^n b^n c^n \mid n \geq 0\}$?
- (a) Regular.
 - (b) Context-free but not regular.
 - (c) Context-sensitive but not context-free.
 - (d) General recursive but not context-sensitive.
 - (e) None of the above.
9. Which one of the following statements about languages and grammars is *false*?
- (a) Some natural languages involve mildly context-sensitive features.
 - (b) Agreement phenomena in natural languages can sometimes be captured by context-free grammars at the expense of an explosion in the number of rules.
 - (c) Typing constraints in programming languages such as Haskell and Java can be readily captured by context-free grammars.
 - (d) Nested clause structures in English (as in ‘This is the rat that ate the malt that lay in the house that Jack built’) can be readily captured by context-free grammars.
 - (e) Nested block structures in programming languages cannot be readily captured by regular grammars.
10. Recall that a Turing machine T can be represented or ‘coded’ by an integer m . Let us write ‘the m th Turing machine’ to mean the Turing machine coded by m . Which of the following sets is *not* recursively enumerable?
- (a) The set of m such that the m th Turing machine halts on the input 0.
 - (b) The set of m such that the m th Turing machine does not halt on the input 0.
 - (c) The set of m such that the m th Turing machine halts on the input m .
 - (d) The set of m such that the m th Turing machine applied to the input 0 yields the output 0.
 - (e) The set of n such that all Turing machines halt on the input n .

11. Which of the following grammars is unsuitable as the basis for a recursive descent parser? (In each case A is the start symbol for the grammar.)
- (a) $A \rightarrow aA \mid bA \mid \varepsilon$
 - (b) $A \rightarrow Bc \mid a \quad B \rightarrow Ab \mid \varepsilon$
 - (c) $A \rightarrow aAc \mid bAc \mid \varepsilon$
 - (d) $A \rightarrow Bc \mid a \quad B \rightarrow bA \mid \varepsilon$
 - (e) None of the above
12. If we consider a Nondeterministic Finite State Machine M with n states, and convert it to a Deterministic Finite State Machine M' using the standard construction, what is the maximum number of states required by M' ?
- (a) n
 - (b) n^2
 - (c) $2n$
 - (d) 2^n
 - (e) $\log_2 n$
13. For a Context-Free Grammar G , which of the following statements is *false*?
- (a) Corresponding to each derivation in G there is exactly one parse tree.
 - (b) Corresponding to each parse tree in G there is exactly one derivation.
 - (c) Corresponding to each leftmost derivation in G there is exactly one parse tree.
 - (d) Corresponding to each rightmost derivation in G there is exactly one parse tree.
 - (e) Corresponding to each parse tree in G there is exactly one leftmost derivation.
14. Which of the following sets is a dependency set for a string of length $2n$ in the language $L = \{xx \mid x \in \{a, b\}^*\}$?
- (a) $\{(i, 2n - i + 1) \mid 1 \leq i \leq n\}$
 - (b) $\{(i, n + i) \mid 1 \leq i \leq n\}$
 - (c) $\{(i, 2n - i) \mid 1 \leq i \leq n\}$
 - (d) $\{(n - i, 2n - i) \mid 1 \leq i \leq n\}$
 - (e) None of the above

15. Consider a Context-Free Grammar with the following rules. Which of the following statements is true of the grammar?

$$A \longrightarrow aAa \mid aAb \mid b$$

- (a) Unambiguous, locally unambiguous and $LL(k)$ for some k
 - (b) Unambiguous and locally ambiguous
 - (c) Ambiguous
 - (d) Unambiguous, locally unambiguous but not $LL(k)$ for any k
 - (e) None of the above
16. Consider a Context-Free Grammar with the following rules. Which of the following sets is $\text{First}_1(A)$?

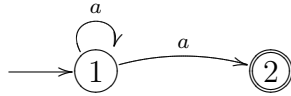
$$\begin{aligned} S &\rightarrow A \dashv \\ A &\rightarrow bAc \mid dAc \mid B \\ B &\rightarrow \varepsilon \mid aB \end{aligned}$$

- (a) $\{b, d\}$
 - (b) $\{b, d, c\}$
 - (c) $\{b, d, a\}$
 - (d) $\{b, d, a, \varepsilon\}$
 - (e) None of the above
17. Consider the following two Context-Free Grammars, G_1 and G_2 with the following sets of rules (S_1 is the top symbol for G_1 and S_2 is the top symbol for G_2). Which of the following languages is $L(G_1) \cap L(G_2)$?

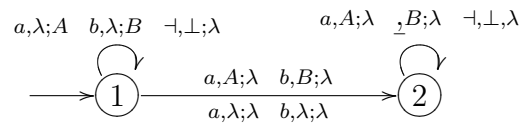
$$\begin{aligned} S_1 &\rightarrow cS_1 \mid aS_1b \mid bS_1a \mid S_1S_1 \mid \varepsilon \\ S_2 &\rightarrow bS_2 \mid aS_2c \mid cS_2a \mid S_2S_2 \mid \varepsilon \end{aligned}$$

- (a) $\{a^n b^n c^n \mid n \geq 0\}$
- (b) $\{b^n a^n c^n \mid n \geq 0\} \cup \{c^n a^n b^n \mid n \geq 0\}$
- (c) $\{w \in \{a, b, c\}^* \mid \#_a(w) = \#_b(w) = \#_c(w)\}$ where $\#_a(w)$ stands for the number of a s in w .
- (d) \emptyset
- (e) None of the above

18. Consider the following NFA M and say what language is recognised by constructing the machine that recognises the *complement* of $L(M)$ in $\{a\}^*$.



- (a) \emptyset
 (b) $\{a\}^*$
 (c) $\{a\}$
 (d) $\{\varepsilon\}$
 (e) None of the above
19. What is the language recognised by the following pushdown automaton? The machine accepts with an empty stack and the initial symbol on the stack is \perp . In addition, we are using λ for the empty string of stack symbols and \dashv is the end of input symbol.



- (a) \emptyset
 (b) $\{w \dashv \mid w \in \{a, b\}^*, w = w^R\}$ where w^R is the *reverse* of the string w .
 (c) $\{w \dashv \mid \#_a(w) = \#_b(w)\}$
 (d) $\{a^n b^n \dashv \mid n \geq 0\}$
 (e) None of the above
20. Which of the following phrases most closely capture what it means for a grammar to be *weakly adequate* for a Natural Language?
- (a) The grammar generates some of the strings in the Natural Language
 (b) The grammar generates all and only the strings in the Natural Language
 (c) The grammar generates all and only the strings in the Natural Language but assigns the wrong structure to some of the sentences.
 (d) The grammar generates some of the strings in the Natural Language and assigns the correct structure to those strings.
 (e) None of the above

PART B

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

1. A home delivery food chain proposes to let customers order meals online by typing any phrase admitted by the following context-free grammar, in which the start symbol is NP, and the lowercase English words are terminals:

$$\begin{aligned} \text{NP} &\rightarrow \text{N} \mid \text{Adj NP} \mid \text{NP and NP} \mid \text{NP in NP sauce} \\ \text{N} &\rightarrow \text{chicken} \mid \text{mushroom} \mid \text{pumpkin} \mid \text{orange} \\ \text{Adj} &\rightarrow \text{fresh} \mid \text{roasted} \mid \text{organic} \end{aligned}$$

The phrase is then parsed, and the cost of the meal systematically computed from the parse tree.

- (a) Using any reasonable notation, give a compositional definition of some possible *cost function* which, given a parse tree for a phrase in this language, returns a price in British currency. Your cost function should have the following characteristics:

- A sauce made from X costs half the price of a main portion of X.
- Fresh produce costs 50% extra.

[7 marks]

- (b) One problem with this proposal is that the above grammar is ambiguous. Consider for example the phrase:

fresh chicken and mushroom in pumpkin sauce

How many parse trees are there for this phrase? Draw *two* trees that are assigned a different cost by the function you defined in part (a), and compute the cost for each of them.

[6 marks]

- (c) One approach to addressing the issue of ambiguity is to attach *probabilities* to the various rules in the grammar in order to capture the idea that certain parse trees are more likely than others. Either assign probabilities to the rules with left hand side NP in such a way that one parse tree for the above phrase becomes more probable than all the others, or explain clearly why this is not possible in this case.

[3 marks]

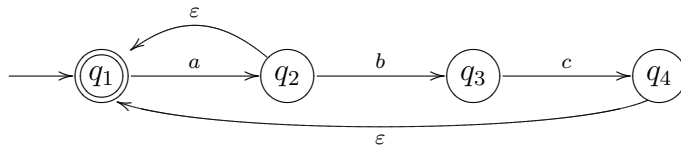
- (d) Another way to address the issue of ambiguity is to replace the grammar by an unambiguous one that generates exactly the same sentences. Write down an LL(1) grammar that is equivalent to the one above and give its parse table. For the purpose of this part, you may treat the symbols N and Adj as if they were terminals and ignore the rules that rewrite these to English words.

[7 marks]

- (e) State one reason why the solution suggested in part (d) is *not* a good solution in the context of the proposed application.

[2 marks]

2. Consider the following NFA M . M is a model of a simple system that repeatedly carries out the actions a followed by b , followed by c . Unfortunately, it also has a bug that means sometimes the machine “forgets” it has carried out an a action.



- (a) Use the techniques of Kleene’s theorem to convert this NFA to a Regular Expression. [6 marks]
- (b) The technician trying to fix the bug does not understand non-deterministic machines. Use the standard construction to convert M to a deterministic machine. [6 marks]
- (c) The technician’s boss is fed up with how long it is taking to fix the bug. She argues that the bug doesn’t happen very often and the system that uses this machine as a component works fine with an occasional repeated a . She proposes monitoring this by making a machine that accepts the language $L = \{x\$w \mid x \in \{a^*bc\}^*, w = a^{\#_a(x)}\}$. Strings in L comprise a string x in $\{a^*bc\}^*$ followed by a dollar symbol followed by a string of a s whose length is equal to the number of appearances of a in x . The technician thinks that this language could be recognised by a Finite State Automaton.
- i. Do you think the technician is right? [1 mark]
 - ii. Write reasonably detailed notes to support your view. [3 marks]
 - iii. Construct a machine that recognises the language L . [3 marks]
- (d) What kind of grammar would you use to specify the language L ? Provide a brief outline of how you would expect the grammar to generate the language and provide a grammar for the language. [6 marks]

3. (a) Explain what it is for a context-free grammar to be *ambiguous*. What are the consequences of this for the efficiency of any parsing process for such languages? Ignore issues of lexical ambiguity: Assume each word belongs to only one class. [3 marks]
- (b) Consider the following probabilistic context-free grammar. Where probabilities are not assigned explicitly, you can assume all rules for that nonterminal are equally probable. The rules of the grammar are (the top symbol is S):

$$\begin{aligned}
 S &\rightarrow AS [0.1] \mid AS' [0.9] \\
 S' &\rightarrow S'B \mid B \\
 A &\rightarrow b \mid c \\
 B &\rightarrow a \mid b
 \end{aligned}$$

Using the above grammar, provide *two* derivation trees that are assigned different probabilities by the grammar for the following sentence in the language:

cbba [4 marks]

- (c) Calculate the probability for both of your derivation trees and indicate the most likely structure for the sentence. Ignore probabilities for productions with A or B on the left hand side. These probabilities are not supplied. [4 marks]
- (d) Redesign the above grammar by changing the probabilities on the productions so that the choice of more likely derivation is reversed. Recalculate the probabilities to demonstrate this is the case. [6 marks]
- (e) What is the language generated by the grammar? Justify your answer. [4 marks]
- (f) Devise a new, unambiguous, grammar that generates the same language as the original grammar. [4 marks]