UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFORMATICS 2A: PROCESSING FORMAL AND NATURAL LANGUAGES

December 16, 2009

09:30 to 11:30

Convener: J Bradfield
External Examiner: A Preece

INSTRUCTIONS TO CANDIDATES

1. **Candidates in the third or later year of study for the degrees of MA(General), BA(Relig Stud), BD, BCom, BSc(Social Science), BSc (Science) and BEng should put a tick ($\surd$) in the box on the front cover of the script book.**
2. **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 multiple choice exam answers. Part B contains THREE questions. Answer any TWO. Each is worth 25%.**
3. **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.**

Write as legibly as possible.
CALCULATORS ARE NOT PERMITTED.

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

1. Consider the language $L = \{x \in \{a,b\}^* \mid \#_a(x) = \#_b(x)^2\}$. You are using the pumping lemma to show L is *not* regular. In the proof by contradiction you have assumed that L is recognised by a k state FSM. What would be a good choice of string to consider in using the pumping lemma to prove that L is not regular?

   (a) $aaaabb$
   (b) $aaaabbbb$
   (c) $a^{k^2}b^k$
   (d) a^kb^k
   (e) None of the above.

2. Consider the following GNF context-free grammar G:

   $$G = (\{S, L, R\}, \{\texttt{a}, \texttt{;}, \texttt{[}, \texttt{]}\}, \{S \to \texttt{[}L \mid \texttt{a}, \quad L \to \texttt{a}R \mid \texttt{[}LR, \quad R \to \texttt{]} \mid \ \texttt{;}L\}, S)$$

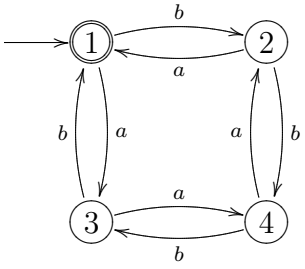
   How many different parse trees are there for the string `[[a],a]`?

   (a) 0
   (b) 1
   (c) 2
   (d) 3
   (e) More than three

3. Which of the following is the dependency set for a string $x = a_1 \dots a_{2n}$ 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) $\{(i, n - i + 1) \mid 1 \leq i \leq n\}$
   (c) $\{(i, n + i) \mid 1 \leq i \leq n\}$
   (d) $\emptyset$
   (e) None of the above.

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



   (a) $\{x \in \{a,b\}^* \mid \#_a(x)$ and $\#_b(x)$ are both even$\}$
   (b) $\{x \in \{a,b\}^* \mid \#_a(x)$ and $\#_b(x)$ are both odd$\}$
   (c) $\{x \in \{a,b\}^* \mid \#_b(x) - \#_a(x)$ is divisible by 4$\}$
   (d) $\{x \in \{a,b\}^* \mid \#_b(x)$ is divisible by 4$\}$
   (e) none of the above

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

   (a) *bbaa*
   (b) *aabbba*
   (c) *aaaabb*
   (d) *babbaa*
   (e) None of the above.

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

   (a) *bbbbbb*
   (b) *bab*
   (c) *bbb*
   (d) *babbbb*
   (e) None of the above.

7. 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 *unambiguously*. In all cases, the start symbol is S and the alphabet is $\{a\}$.

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

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

   (c) $S \to aA \mid Ba \quad A \to aaA \mid \varepsilon \quad B \to Baa \mid \varepsilon$

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

   (e) None of the above.

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

   (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.

9. Consider the following context-free language: $L_1 = \{x \in \{a, b, c\}^* \mid \#_a(x) = \#_b(x)\}$. 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) $L_2 = \{a^{2k} b^{2k} c^{2k} \mid k \geq 0\}$

   (e) None of the above.

10. Consider the following context-free grammar G:

$$G = (\{S, A, B, C\}, \{a, b, c, \dashv\}, \{S \to BC \dashv, A \to AA \mid a \mid \varepsilon, B \to BAB \mid b \mid \varepsilon, C \to cC \mid c\}, S)$$

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

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

11. The language used in speaking the integers up to a billion – eg.

23	twenty three	2356789	two million three hundred and fifty six
235	two hundred and thirty five	23567	twenty three thousand five hundred and sixty seven

can be described by a grammar that is

(a) regular but not context-free
(b) context-free but not regular
(c) regular and context-free
(d) context-sensitive but not context-free
(e) neither regular nor context-free

12. Which of the following statements is true of a **parse tree**?

(a) It is an order-independent representation of the set of all CF derivations of a given string.
(b) It is an order-independent representation of the set of equivalent context-free (CF) derivations of a given string.
(c) It is an order-independent representation of the set of equivalent CF or context-sensitive (CS) derivations of a given string.
(d) It is an order-sensitive representation of the set of equivalent CS derivations of a given string.

13. On the basis of language complexity, which of the following is **not** a possible human language?

    (a) a palindromic language with strings of arbitrary length

    (b) a language whose strings consist of words in any order (ie, word order doesn't matter)

    (c) a copying language whose strings contain three copies of an arbitrary substring

    (d) a lexically-restricted language with only 100 words

    (e) a language whose words lack a part-of-speech class.

14. Zipf's law states that:

    (a) Most words in any corpus of real texts will be nouns.

    (b) There are an infinite number of possible words in any human language.

    (c) Half the words in a language by type will account for half the tokens found in any corpus of real texts.

    (d) The rank of a word in a corpus of real texts is inversely proportional to its frequency of occurrence in the corpus.

    (e) No corpus of real texts will contain all the words in a language.

15. What type of Part-of-Speech (PoS) tagger does not need a separate strategy for handling unknown words?

    (a) a Rule-based tagger.

    (b) a Default tagger.

    (c) A Unigram tagger.

    (d) A Regular-expression tagger.

16. Which of the following accurately specifies a constraint that a parser puts on the kind of grammar that it can use in parsing?

    (a) A shift-reduce parser requires a grammar to be in Greibach Normal Form.

    (b) A CKY parser requires grammar rules that rewrite to only one or two symbols on their RHS.

    (c) A recursive-descent parser requires a grammar to have no left recursive rules.

    (d) An LL(1) parser requires a grammar to be in Chomsky Normal Form.

17. 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 the parse of a sentence is the sum of the probabilities of all the rules used to derive this parse.

    (d) The probability of a sentence is the sum of the probabilities of all its parses.

    (e) The probability of a sentence is the product of the probabilities of all its parses.

18. What is the probability that the following probabilistic context-free grammar assigns to the sentence *Peter lost the telescope*?

    S → NP VP (1.0)
    VP → V NP (0.9)
    VP → V NP PP (0.1)
    NP → Det N (0.5)
    NP → PN (0.5)
    PP → P NP (1.0)
    PN → Peter (0.5)
    PN → Mary (0.5)
    V → lost (1.0)
    P → with (1.0)
    Det → the (1.0)
    N → telescope (0.1)
    N → house (0.3)
    N → man (0.6)

    (a) 0.45

    (b) 0.225

    (c) 0.01125

    (d) 0.025

    (e) 0.0125

19. What is the correct meaning representation for *Every student that sits Inf2A is smart*?

    (a) $\forall x.(student(x) \wedge sit(x, Inf2A)) \Rightarrow is\_smart(x)$

    (b) $\exists x.(student(x) \wedge sit(x, Inf2A) \wedge is\_smart(x))$

    (c) $\forall x.(student(x) \Rightarrow sit((x, Inf2A) \wedge is\_smart(x)))$

    (d) $\exists x.(student(x) \wedge sit(x, Inf2A)) \Rightarrow is\_smart(x)$

    (e) $\forall x.(student(x) \wedge sit(x, Inf2A)) \wedge is\_smart(x)$

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

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

(a) $(\lambda P \lambda Q.\exists x.(P(x) \wedge Q(x)))(sneeze(dog))$

(b) sneeze(dog)

(c) $(\lambda P.\exists x.(sneeze(x) \wedge P(x)))(dog)$

(d) $\exists x.(dog(x) \wedge sneeze(x))$

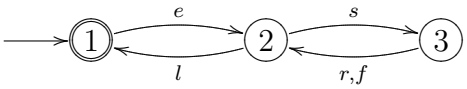
Part B
ANSWER TWO QUESTIONS FROM PART B

1. In this question you are asked to model a simple voting machine where only one person is allowed into the voting booth at a time and people can vote or fail to vote when they use the machine.

   (a) The initial version of the voting machine can do the following actions:

   - e – enter the voting booth
   - l – leave the voting booth
   - s – sign on to the voting machine
   - r – register a vote
   - f – fail to register a vote

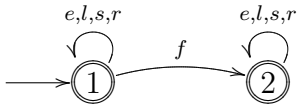
   The machine M_1 that models the voting booth is:



   The designer of the voting booth wants to check the gate 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%]

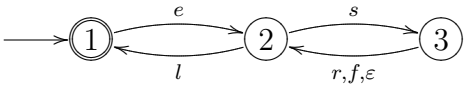
   (b) The voting booth designer observes that many users are repeatedly failing to vote and decides that voters should only be allowed one failure. The machine that allows any action when one or fewer fails have been observed is M_2:



   Combine M_2 with the original model, M_1, of the voting booth using the intersection construction for Finite State Machines to construct a new version of the voting booth that stops working once the user has failed to register a vote twice.

   [5%]

(c) The designer discovers that there is a flaw in the design of the machine and occasionally it fails to register the user's vote or failure to register a vote. The designer's model M_3 is the following:



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

[6%]

(d) The designer now wants to enlarge the market for the voting booth M_1 by adding additional checks to the booth. A friend has pointed out to the designer that once someone enters the booth they could vote several times (if they had additional voter registrations "borrowed" from friends). The designer thinks of a quick fix and claims to have a finite state machine M_5 such that it accepts a subset of the strings accepted by M_1 with the additional constraint that the number of e's accepted by the machine is never smaller than the number of r's. 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%]

(e) Finally the designer decides to build a monitor machine M_6 that will just check if the number of es is always at least as great as the number of rs so at no point can anyone have registered two votes. Can you construct the PDA M_6 such that:

$$L(M_6) = \{x \in \{e, l, s, r, f\}^* \mid \text{for every prefix } p \text{ of } x : \#_e(p) \geq \#_r(p)\}$$

[4%]

2. In English, when the subject and object of a clause are the same — that is, someone or something is doing something to itself — then a *reflexive pronoun* such as "myself", "yourself", "herself", etc. should be used in object position — eg.

   (a) Fred scratched himself
   (b) a dog scratched itself
   (c) it scratched itself

   While *personal pronouns* can appear in object position in English – eg,

   (d) Fred scratched it

   reflexive pronouns can never appear in subject position:

   (e) *itself scratched a dog
   (f) *himself shaved himself
   (g) *myself ate a banana.

   While English makes other use of reflexive pronouns – for example, in *emphatic constructions* like "I myself would never do that", these uses are not relevant to the current exercise.

   Start with the following grammar. (Later, for parts (c) and (d), we will attach semantic *valuation functions* to its rules.)

S → NP VP	PRO → **it**
NP → DET N	N → **dog**
NP → NPR	V → **scratched**
NP → PRO	DET → **a**
VP → V NP	NPR → **fred**

   (a) Using the part-of-speech label **PrRflx** for *reflexive pronoun*, modify this grammar so that it can generate sentences like a–d above, but not sentences like e–g. (In modifying the grammar, you can **add**, **remove**, or **change** rules.) Make sure that you include both grammatical rules (like those in the left-hand column above) and lexical rules (like those in the right-hand column).

   [5%]

(b) Using your grammar, show the chart produced by the CYK algorithm analysing the sentence

a dog scratched itself

You can represent the chart as either a matrix or a graph.

[5%]

(c) We want the interpretation of VPs of the form "scratched himself", "scratched itself", "scratched themselves", etc. to be:

λ x . scratched(x)(x)

where **scratched(x)** is taken to be a functor that takes **x** as its argument. This can also be written as **scratched(x,x)**, but you'll find it easier to deal with in the form given above.

Here we have attached simple semantic valuation functions to some of the grammar rules given above:

S[sem=<?np(?vp)>] → NP[sem=?np] VP[sem=?vp]
NP[sem=<?det(?n)>] → DET[sem=?det] N[sem=?n]
NP[sem=?npr] → NPR[sem=?npr]
VP[sem=<?obj(?v)>] → V[sem=?v] NP[sem=?obj]

N[sem=dog] → **dog**
NPR[sem=λP . P(fred)] → **fred**
DET[sem=λQ λP . $\exists$ x . (Q(x) & P(x))] → **a**
V[sem=λy λx . scratched(y)(x)]→ **scratched**

Attach a semantic valuation function to each rule you have **added** or **changed** in part (a), so as to produce the above interpretation for sentences of the form "scratched itself", "scratched himself", etc. Give your rules and their attached semantic valuation functions.

[5%]

(d) Give the parse tree for the sentence *fred scratched himself*, showing the *beta-reduced* form of the valuation function at each non-terminal node in the tree, up to the root node, S.

[5%]

(e) Some verbs occur relatively frequently with reflexives, while others do not. For example, in the *Wall Street Journal Corpus*, 10 of the 11 occurrences (≈91%) of the verb *"divest"* occur with a reflexive object (eg, *"divest itself"*), while only 3 of the 21 occurrences (≈14%) of the verb *"portray"* occur with a reflexive object (eg, *"portray itself"*), and none of the 29 occurrences (0%) of the verb *"resist"* do so.

Can you capture this kind of frequency difference in a standard probabilistic context-free grammar (PCFG)? If so, how? If not, why not?

Could you capture this kind of frequency difference in a lexicalised PCFG? If so, how? If not, why not?

[5%]

3. Consider the grammar rules for simple sentences (some categories e.g. Determiner have been omitted to reduce the number of rules):

$$
\begin{aligned}
\text{S} &\rightarrow \text{NP VP} \\
\text{NP} &\rightarrow \text{N} \mid \text{N PP} \\
\text{VP} &\rightarrow \text{V NP} \mid \text{VP PP} \\
\text{PP} &\rightarrow \text{Prep NP}
\end{aligned}
$$

$$
\begin{aligned}
\text{N} &\rightarrow \textbf{Paul} \mid \textbf{Mary} \mid \textbf{town} \\
\text{V} &\rightarrow \textbf{met} \mid \textbf{found} \\
\text{Prep} &\rightarrow \textbf{in} \mid \textbf{by}
\end{aligned}
$$

(a) Is the sentence: **Paul met Mary in town** globally ambiguous? Is there a part of the sentence that is locally ambiguous, that does not lead to global ambiguity? What are the possible parses of the sentence?

[4%]

(b) In this part you will create a chart for the first two words of the sentence **Paul met Mary in town** using the bottom-up active parsing strategy. You should do the following:

i. Create the dotted rule set for the grammar.

[3%]

ii. Construct the chart for the first two words of the sentence **Paul met Mary in town**. You should take care to detail how you initialize the chart and the rules you use to add further edges to the chart.

[5%]

In an attempt to simplify the grammar a Programming Language designer pro-

duces a revised version:

$$\begin{aligned}
\text{S} &\rightarrow \text{NP VP} \\
\text{NP} &\rightarrow \text{N PP} \\
\text{VP} &\rightarrow \text{V NP} \\
\text{PP} &\rightarrow \varepsilon \mid \text{Prep N PP}
\end{aligned}$$

$$\begin{aligned}
\text{N} &\rightarrow \textbf{Paul} \mid \textbf{Mary} \mid \textbf{Town} \\
\text{V} &\rightarrow \textbf{met} \mid \textbf{found} \\
\text{Prep} &\rightarrow \textbf{in} \mid \textbf{by}
\end{aligned}$$

(c) If the grammar is ambiguous, demonstrate this. Otherwise provide a brief justification for why the grammar is unambiguous.

[1%]

(d) Consider the parse tree(s) you obtain for the sentence **Paul met Mary in town**. Do they correspond with your intuitions?

[1%]

(e) Augment the revised grammar with the new production $S' \rightarrow S \vdash$ and then calculate $\text{First}_1(A)$ for each of the nonterminals $\text{S}, \text{NP}, \text{VP}, \text{PP}$. (When you calculate First, use the categories $\text{N}, \text{V}, \text{Prep}$ as the terminal symbols of the grammar rather than the spcific words.)

[3%]

(f) Using this augmented grammar, calculate $\text{Follow}_1 A$ for each of the nonterminals $\text{S}, \text{NP}, \text{VP}, \text{PR}, \text{PP}$. (Again, when you calculate Follow, use the categories $\text{N}, \text{V}, \text{Prep}$ as the terminal symbols of the grammar, rather than the specific words.)

[4%]

(g) Construct the $LL(1)$ parse table for this grammar.

[4%]

Specimen Answers

Part A

1. c – because (a) and (b) are not parameterised by k and so cannot be a good choice, choice (d) is not in the language. Choice (c) is in the language and is parameterised by k.

2. b because the grammar G is in GNF and is deterministic so there is a unique leftmost derivation for any string in the language.

3. c is the correct answer, (a) is incorrect because it matches the first character with the last character, second with second last, ..., (b) is incorrect because it does not reference the second half of the string, (c) matches the definition, (d) is incorrect because there clearly are dependencies.

4. c - the machine counts the number of bs mod 4 and occurrences of a are subtracted from the total.

5. c - any occurrence of bb cannot be followed by anything - this eliminates (a), (b) and (d) and (c) is in the language.

6. d - because for the string to be a member of R_1, $abbbb$ must be a member of $((ab)^*(bb)^*)^*$ this is not the case.

7. d - the grammar uses right linear rules so it generates a unique derivation for each string and generates only strings of odd length.

8. b - this is context-free because it is just the concatenation of $a^n b^n$ with $b^m c^m$ so this is context-free because these two languages are CF and CFLs are closed under concatenation.

9. c - (a) has intersection c^*, (b) has L_2 as intersection, (c) has $a^k b^k c^k$ as intersection, (d) is not a CFL

10. a - answer (b) omits the a symbol that arises from the rule $A \rightarrow AA$, answer (c) includes $\dashv$ that cannot follow A, answer (d) is wrong because it includes ε

11. a

12. b

13. c

14. d

15. b

16. c
17. d
18. c
19. a
20. d

Part B

1. (a) The equations derivable from the machine are:

$$\begin{aligned} R_1 &= eR_2 + \varepsilon \\ R_2 &= lR_1 + sR_3 \\ R_3 &= (f + r)R_2 \end{aligned}$$

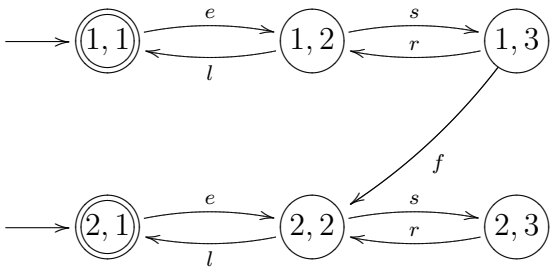
Allocate three marks for the equations.

Solving the equations:

$$\begin{aligned} R_2 &= lR_1 + sR_3 \\ R_2 &= lR_1 + s(f + r)R_2 \\ R_2 &= (s(f + r))^* lR_1 \\ R_1 &= e(s(f + r))^* lR_1 + \varepsilon \\ R_1 &= (e(s(f + r))^* l)^* \end{aligned}$$

Allocate three marks for solving the equations, include 1 mark for recalling that $R = AR + B$ has solution A^*B.

(b) The intersection construction yeilds the machine:



Allocate 2 marks for getting the stateset correct, including the final states. Allocate a further two marks for the non-f transitions and one mark for the f transition.

(c) In this section we use the subset construction to construct the new machine. The stateset of the new machine comprises subsets of the states of the non-deterministic machine. The new transition function is:

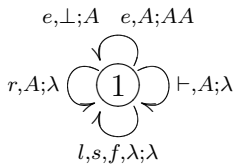
State	e	l	s	r	f
1	2				
2		1	3,2		
3, 2	1	3,2	2	2	

Award two marks for using subsets of the stateset and a further 4 for the transition function.

(d)
- I disagree with the designer. *Award one mark*
- Use the pumping lemma for regular languages to demonstrate the designer's claim is false:
  - Consider the string $x = (esrsrl)^k(el)^{k+1} \in L(M_5)$.
  - By the pumping lemma there are strings $uvw = x$ with the length of uv less than k and $v \neq \varepsilon$ such that $uw \in L(M_5)$
  - uv must be a prefix of $(esrsrl)^k$
  - There are three cases:
    i. $v = s\alpha r$ for some $\alpha \in \{e, s, r, l\}^*$ in this case the number of rs in v is larger than the number of es in v and so for an appropriate choice of i, uv^iw contains more rs than es.
    ii. $v = e\alpha l$ for some $\alpha \in \{e, s, r, l\}^*$ similar to the previous case.
    iii. otherwise uv^iw is not accepted by M_5 for some suitable choice of i.

Award up to three marks for something that captures this line of argument (it need not be formal).

(e) The PDA is (state 1 is the initial state, I have omitted the in arrow on state 1 for legibility. The notation $l, s, f, \lambda; \lambda$ stands for three transitions $l, \lambda; \lambda, \ldots$). The machine accepts by empty stack and we have used $\vdash$ to indicate end of input:



Allocate 1 mark for geting the PDA notation correct, two marks for the r and e transitions and a further one mark for the other transitions.

2. In English, when the subject and object of a clause are the same — that is, someone or something is doing something to itself — then a *reflexive pronoun* such as "myself", "yourself", "herself", etc. should be used in object position — eg.

   (a) a dog scratched itself
   (b) Fred shaved himself
   (c) I amused myself

   Reflexive pronouns never appear in subject position in English

   (d) *itself scratched a dog
   (e) *himself shaved himself
   (f) *myself ate a banana.

   While English uses reflexive pronouns for other purposes as well – for example, in *emphatic constructions* like "I myself would never do that", they are not relevant to the current exercise.

   Start with the following grammar. (Later, for parts (c) and (d), we will attach valuation functions to the rules.)

S → NP VP	Pro → it | I
NP → Det N	N → dog
NP → NPR	V → scratched
NP → Pro	Det → a
VP → V NP	NPR → **Fred**

   (a) Using the part-of-speech label **PrRflx** for *reflexive pronoun*, modify this grammar so that it can generate sentences like a–c above, but not sentences like d–f.

   Additional lexical rules:

   PrRflx → itself | himself | herself | ...

   Additional grammatical rules:

   VP → V PrRflx

   A rule of the form (NP → PrRflx) will end up generating sentences like d–f, so is wrong. Score 2 pts for the lexical rule, 3 pts for the grammar rule.

   [5%]

   (b) Using your grammar, show the chart produced by the CYK algorithm analysing the sentence *a dog scratched itself*.

   The arcs/entries are as follows:

(0,1) Det
(1,2) N
(0,2) NP
(2,3) V
(3,4) PrRflx
(2,4) VP
(0,4) S

Score 2 points for the diagonal, then 1 point each for the other three arcs/entries. [5%]

(c) We want the interpretation of VPs like "scratched itself", "scratched himself", etc. to be

λ x . scratched(x)(x)

where **scratched(x)** is taken to be a functor that takes **x** as its argument. This can also be written as **scratched(x,x)**, but you'll find it easier to deal with in the form given above.

Here we have attached simple semantic valuation functions to some of the grammar rules given above:

S[sem=<?np(?vp)>] → NP[sem=?np] VP[sem=?vp]
NP[sem=<?det(?n)>] → DET[sem=?det] N[sem=?n]
NP[sem=?npr] → NPR[sem=?npr]
VP[sem=<?obj(?v)>] → V[sem=?v] NP[sem=?obj]

N[sem=dog] → **dog**
NPR[sem=λP . P(fred)] → **fred** DET[sem=λQ λP . $\exists$ x . (Q(x) & P(x))]
→ **a**
V[sem=λy λx . scratched(y)(x)]→ **scratched**

Attach a semantic valuation function to each rule you have **added** or **changed** in part (a) so as to produce this kind of interpretation for "scratched itself", "scratched himself", etc.

[5%]

Lexical rules:

PrRflx[sem=λP λz . P(z)(z)] → itself | himself | herself | ...

Add grammar rules:

VP[sem=<?rflex(?v)>] → V[sem=?v] PrRflx[sem=?rflex]

(d) Give the parse tree for the sentence *fred scratched himself*, showing the *beta-reduced* form of the valuation function at each non-terminal node in the tree,

up to the root node, S.

[5%]

answer: Tree reflects S → NP VP → NPR VP → NPR V PrRflx

S : beta-reduced valuation function **scratched(fred,fred)** or **scratched(fred)(fred)**
NP: valuation function λ**P . P(fred)**
VP: beta-reduced valuation function λ **z . scratched(z)(z)**
V: valuation function λ **y** λ **x . scratched(y)(x)**
PrRflx: valuation function λ**P** λ**z . P(z)(z)**

(e) Some verbs occur relatively frequently with reflexives, while others do not. For example, in the *Wall Street Journal Corpus*, of the 11 occurances of the verb *"divest"*, 10 occur with a reflexive object (eg, *"divest itself"*), while only 3 of the 21 occurances of the verb *"portray"* occur with a reflexive object (eg, *"portray itsef"*), and none of the 29 occurances of the verb *"resist"* do so.

Could you capture this kind of difference in frequency in a standard probabilistic context-free grammar (PCFG)? If so, how? If not, why not?

Answer: The difference cannot be captured in a standard PCFG with rules such as VP → V PrRflx, unless there is a separate rule for each verb V. This is half-way to a lexicalised PCFG.

Could you capture this kind of difference in frequency in a lexicalised PCFG? If so, how? If not, why not?

Answer: By lexicalising the head (ie, V in this case), we can get rules of the form VP(divest) → V(divest) PrRflx, VP(resist) → V(resist) PrRflx, whose probability can be specific to each different V.

[5%]

3. (a) Yes the grammar is globally ambiguous. *Allocate 1 mark.* There is no local ambiguity that does not give rise to global ambiguity. *Allocate 1 mark.* There are two possible parses – the essential difference is whether the prepositional phrase is attached to the noun or the verb:
   - **Paul [met [Mary in town]]**: This uses the rule: VP → V NP.
   - **Paul [met [Mary] [in town]]**: This uses the rule: VP → VP PP.

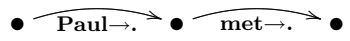
   Allocate 1 mark for each parse.

   (b) i. The dotted rule set introduces a dot at each possible position in the rules to indicate what has been parsed and what is predicted to parse for that rule.

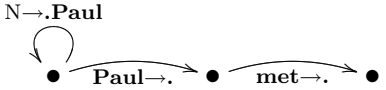
$$
\begin{aligned}
\text{S} &\rightarrow \textbf{.}\ \text{NP VP}\\
\text{S} &\rightarrow \text{NP}\ \textbf{.}\ \text{VP}\\
\text{S} &\rightarrow \text{NP VP}\ \textbf{.}\\
&\ldots\\
\text{N} &\rightarrow \textbf{. Paul}\\
&\ldots\\
\text{N} &\rightarrow \textbf{Paul .}\\
&\ldots\\
\textbf{Paul} &\rightarrow \textbf{.}
\end{aligned}
$$

   Allocate 1 mark for each type of rule.

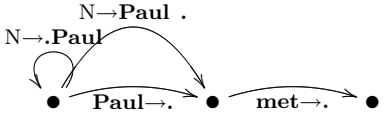
   ii. *Allocate 1 mark for the initialisation rule, 2 marks for the bottom-up predict rule and its application, 2 marks for the fundamental rule and its application.*
   - This is the initialised chart:



   - The bottom-up predict rule will add edges like this:



   - The fundamental rule will add edges like this:



   (c) The revised grammar is unambiguous because the prepositional phrase can only be associated with the noun not with the verb. This is the only source of ambiguity in the language, so the omission of these rules results in an unambiguous grammar. *Allocate one mark for the answer and justification.*

(d) The parse is not intuitive since the PP is associated with **Mary** rather than with the verb. *Allocate one mark*

(e) Construct the table that approximates the first sets:

	0	1	2
S	$\emptyset$	$\emptyset$	N
NP	$\emptyset$	N	N
VP	$\emptyset$	V	V
PP	$\emptyset$	ε, Prep	ε, Prep

The only nonterminal we need to consider is PP:

$\text{Follow}_1(PP) = \text{Follow}_1(PP) \cup \text{Follow}_1(NP) = \text{First}_1(\text{VP}) \cup \text{Follow}_1(\text{VP}) = \{\text{V}, \vdash\}$

(f) The parse table is:

	V	N	Prep	$\vdash$
S		NP VP		
NP		N PP		
VP	V NP			
PP	ε		Prep N PP	ε

Allocate two marks to the PP row and two marks to the other rows.