UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFORMATICS 2A: PROCESSING FORMAL AND NATURAL LANGUAGES

December 9, 2008

14:30 to 16:30

Convener: M O'Boyle
External Examiner: R Irving

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.

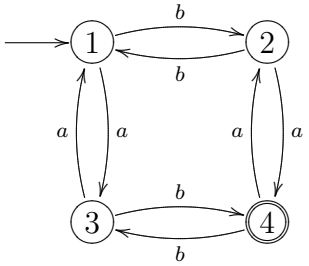
1. Someone asserts that the language $L = \{x \in \{a, b\}^* \mid x = x^R\}$ is recognisable by a finite state machine with k states. You are in the process of demonstrating this is false using the pumping lemma. What would be a good choice of string to consider in using the pumping lemma to prove the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

   (a) $aaabbbbaaa$
   (b) $aaaaaaabbbbbbb$
   (c) $a^k b^k$
   (d) $a^k b a^k$
   (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 $L = \{x \in \{a, b\}^* \mid x = x^R\}$?

   (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. Given that $\#_a(x)$ stands for the number of a symbols in string x, 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) $\{x \in \{a, b\}^* \mid \#_b(x)$ is even$\}$
   (d) $\{x \in \{a, b\}^* \mid \#_b(x)$ is odd$\}$
   (e) none of the above

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

   (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: $R_1 = (b(ab^*a)^*b)^*$ and $R_2 = (b + ((ba)^*a))^*$. Which of the following strings is contained in one of the languages but not in the other?

   (a) *baab*

   (b) *baaaab*

   (c) *bbbaa*

   (d) *baabbaab*

   (e) None of the above.

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

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

   (b) $S \rightarrow aaS \mid \varepsilon$

   (c) $S \rightarrow SS \mid aa \mid \varepsilon$

   (d) $S \rightarrow \varepsilon \mid aA \mid Ba \mid \varepsilon \quad A \rightarrow aaA \mid a \quad B \rightarrow Baa \mid a$

   (e) None of the above.

7. As before, $\#_a(x)$ stands for the number of a symbols in string x. Which of the following descriptions best fits the language $L = \{x \in \{a, b, c\}^* \mid \#_a(x) = \#_b(x) + \#_c(x)\}$?

   (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: $L_1 = \{x \in \{a,b,c\}^* \mid \#_a(x) = 2\#_b(x)\}$. 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) $L_2 = \{(abc)^{2k} \mid k \geq 0\}$
   (c) $L_2 = \{a^k b^m c^{2k} \mid k \geq 0 \text{ and } m \geq 0\}$
   (d) $L_2 = \{a^k b^{2k} c^{2k} \mid k \geq 0\}$
   (e) None of the above.

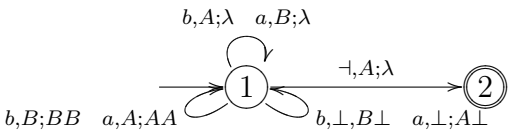
9. Consider the following context-free grammar:

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

   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 $\{A, B, \perp\}$ where $\perp$ is the initial stack symbol. The alphabet of P is $\{a, b, \dashv\}$ where $\dashv$ is used to mark the end of the input. As before, $\#_c(x)$ stands for the number of c symbols in x.



   (a) $\{x \in \{a,b\}^* \dashv \mid \#_a(x) \text{ and } \#_b(x) \text{ are both even}\}$
   (b) $\{x \in \{a,b\}^* \dashv \mid \#_a(x) > \#_b(x)\}$
   (c) $\{x \in \{a,b\}^* \dashv \mid \#_b(x) = 2\#_a(x)\}$
   (d) $\{x \in \{a,b\}^* \dashv \mid \#_b(x) \geq \#_a(x)\}$
   (e) none of the above

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

    (a) It is an order-independent representation of the set of equivalent context-free (CF) derivations of a given string.

    (b) It is an order-independent representation of the set of all 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.

12. On theoretical grounds (ie, on the basis of language complexity), which of the following is **not** a possible human language?

    (a) a language with palindromic strings of arbitrary length

    (b) a language that allows strings of arbitrary words from the language in any possible order (ie, a language in which order doesn't matter)

    (c) a language in which strings contain three exact copies of an arbitrary substring

    (d) a language with only 50 words

    (e) a language that lacks Parts of Speech

13. What is the size of the Well-Formed Substring Table (WFST) for a string of length **N**, where the WFST can store pointers to immediate constituents as well as non-terminal symbols and the spans they cover?

    (a) N

    (b) N^2

    (c) $(N+1)^2$

    (d) N^3

14. The distribution of which of the following phenomena is **not** Zipfian?

    (a) digits in a corpus of integers.

    (b) words in a corpus of English text.

    (c) words in a corpus of Korean text.

    (d) prepositions in a corpus of English text.

    (e) books sold by Amazon (UK).

15. Which of the following lambda expressions does **not** reduce to

    Love(John, Mary)

    (a) λx.Love(John, x)(Mary)

    (b) λx.Love(x, Mary)(John)

    (c) λx.λy.Love(x,y)(John)(Mary)

    (d) λu.λv.Love(v,u)(John)(Mary)

    (e) λP.λw.λz.P(z,w)(John)(Mary)λu.λv.Love(v,u)

16. Which of the following is *not* a valid derivation of the grammar G_1 with the following productions?

$$\begin{array}{rlcl} G_1: & S & \rightarrow & AB \mid BA \mid ABA \\ & A & \rightarrow & AA \mid a \\ & B & \rightarrow & BB \mid b \end{array}$$

    (a) $S \Rightarrow AB \Rightarrow AAB \Rightarrow AABB \Rightarrow aABB \Rightarrow aaBB \Rightarrow aabB \Rightarrow aabb$

    (b) $S \Rightarrow AB \Rightarrow ABB \Rightarrow ABb \Rightarrow AABb \Rightarrow aABb \Rightarrow aaBb \Rightarrow aabb$

    (c) $S \Rightarrow ABA \Rightarrow ABa \Rightarrow aBa \Rightarrow aba$

    (d) $S \Rightarrow AB \Rightarrow AAB \Rightarrow AABA \Rightarrow AaBA \Rightarrow aaBA \Rightarrow aabA \Rightarrow aaba$

    (e) $S \Rightarrow ABA \Rightarrow AABA \Rightarrow AaBA \Rightarrow aaBA \Rightarrow aabA \Rightarrow aaba$

17. Which of the following terms is *not* a plausible formulation of a probabilistic POS tagger?

    (a) $t_i = \arg\max_j P(t_j|t_{i-1}, w_{i-1}, w_i)$

    (b) $t_i = \arg\max_j P(t_j|t_{i-1}, w_i)$

    (c) $t_i = \arg\max_j P(t_j|t_{i-2}, t_{i-1}, w_i)$

    (d) $t_i = \arg\max_j P(t_j|t_{i-1}, w_i, t_{i+1})$

    (e) $t_i = \arg\max_j P(t_j|t_{i-1}, w_{i-1})$

18. Which of the following rules contains indirect recursion?

    (a) NP → Det Adj N
        NP → Pron

    (b) VP → V NP
        NP → Det N that VP

    (c) N' → N N'
        NP → Det N'

    (d) S → AdvP S
        AdvP → Adj Adv

    (e) VP → V NP
        NP → Det N PP

19. 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.

20. Which of the following first order predicate logic formulas is a correct representation of *every ambitious politician took a large bribe*?

    (a) $\forall x.\text{politician}(x) \wedge \text{ambitious}(x) \wedge \exists y.\text{bribe}(y) \wedge \text{large}(y) \wedge \text{took}(x, y)$

    (b) $\forall x.\text{politician}(x) \wedge \text{ambitious}(x) \Rightarrow \exists y.\text{bribe}(y) \Rightarrow \text{large}(y) \wedge \text{took}(x, y)$

    (c) $\forall x.\text{politician}(x) \wedge \text{ambitious}(x) \Rightarrow \exists y.\text{bribe}(y) \wedge \text{large}(y) \wedge \text{took}(x, y)$

    (d) $\exists x.\text{politician}(x) \wedge \text{ambitious}(x) \Rightarrow \forall y.\text{bribe}(y) \wedge \text{large}(y) \wedge \text{took}(x, y)$

    (e) $\exists x.\text{politician}(x) \wedge \text{ambitious}(x) \wedge \forall y.\text{bribe}(y) \wedge \text{large}(y) \Rightarrow \text{took}(x, y)$

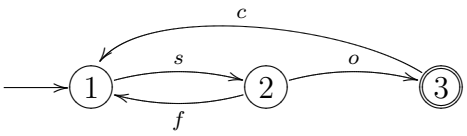
Part B
ANSWER TWO QUESTIONS FROM PART B

1. In this question you are asked to model a simple swipe gate, like the ones at the entrance to the Informatics Forum, that fails from time to time.

   (a) The swipe card system can do the following actions:

   - s – read a swipe card
   - f – fail after reading the card
   - o – open the gate
   - c – close the gate

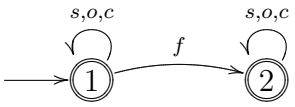
   The machine M_1 that models the gate is:



   The designer of the gate 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%]

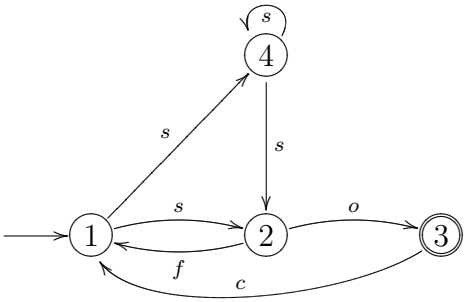
   (b) Using reliability and maintenance data for the gate, the designer decides that if the gate has two swipe failures then it should stop working to await maintenance. The machine that allows any action when at most one fail has been observed is M_2:



   Combine M_2 with the original model, M_1, of the gate using the intersection operation for Finite State Machines to construct a new version of the machine that stops working once it has seen two swipe failures.

   [5%]

   (c) The designer continues to experiment with different models of the swipe gate and tries a model where the gate can ignore swipes as well as explicitly failing. The designer's model M_3 is the following:



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

[6%]

(d) The designer now wants to enlarge the market for the gate M_1 by designing a variant for more secure settings. The new version of the gate has an additional action: e standing for "exit". The designer's idea is to include an exit gate that will let people out provided the number of times the exit button is pressed is smaller than or equal to the number of times the gate has opened. So the language recognised in the new model is (here $\|$ is the interleave operator):

$$L_4 = \{x \in L(M_1) \parallel e^* \mid \text{for every prefix } p \text{ of } x : \#_o(p) \geq \#_e(p)\}$$

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 that the machine can be constructed.

[4%]

(e) The designer decides to build a monitor machine M_6 that will just check if the number of os is always at least as great as the number of es. Can you construct the PDA M_6 such that:

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

[4%]

2. **Transitive verbs** in English such as *love* and *contain* have a standard clause structure corresponding to the two simple grammar rules

   S → NP VP
   VP → V NP

   where the NP on the right-hand side of the sentence (S) rule is called the SUBJECT, and that on the right-hand side of the verb phrase (VP) rule is called the DIRECT OBJECT.

   Ditransitive verbs such as *give* and *throw* standardly expand VP according to the grammar rules

   VP → V NP PP (e.g. *give a flower to John*)
   VP → V NP NP (e.g. *give John a flower*)

   where *John* realises the INDIRECT OBJECT of the ditransitive verb and *a flower* realises its DIRECT OBJECT.

   In actual discourse, English allows **topicalisation** of a DIRECT OBJECT, which moves it to the front of the sentence, as in

   i. Edinburgh, John loves. London, he dislikes.
   ii. A rose, Mary gave John. A daffodil, she gave Fred.
   iii. A rose, Mary gave to John. A daffodil, she gave to Fred.

   (a) Add no more than two grammar rules to the above four, to recognise topicalised sentences such as (i.) involving **transitive verbs**.

   [5%]

   (b) If the grammar also contains the rules

   NP → NPR (e.g. a proper name)
   NP → PRO (e.g. a pronoun)
   NPR → *John* | *Mary* | *Edinburgh* | *London*
   PRO → *he* | *it*
   V → *loves*

   give a complete well-formed substring table (WFST) first for the sentence *John loves Edinburgh*, and then for the sentence *Edinburgh John loves* according to a grammar consisting of the above rules and the rule or rules that you've written for part (a).

   [5%]

(c) In the two rules for **ditransitive verbs** given at the start of the problem, the second rule (repeated here)

VP → V NP NP (e.g. *give John a flower*)

cannot be used if the second NP is realised as a pronoun such as *it*, as in

**Mary gave John it.*

How would you modify and/or extend the grammar in order to avoid accepting or generating sentences such as *Mary gave John it*? (You can refer to the two NP rules given above in part (b).)

[5%]

(d) Could sentences such as *Mary gave John it* be avoided (or given very low probability) in a different way in a **non-lexicalised** probabilistic context-free grammar (PCFG)? If so, give the relevant rules, along with a low probability. If not, explain. (You can refer to the two NP rules given above in part (b).)

[5%]

(e) Could such sentences be avoided (or given very low probability) in a different way in a **lexicalised PCFG** that allowed lexicalisation on parts of speech like NPR and PRO, as well as on specific words? If so, give the rules and their probabilities. If not, explain. (Again, you can refer to the two NP rules given above in part (b).)

[5%]

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

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

$$
\begin{aligned}
\text{N} &\rightarrow \textbf{Fred} \mid \textbf{Hades} \mid \textbf{Orpheus} \mid \textbf{Eurydice} \\
\text{V} &\rightarrow \textbf{saw} \mid \textbf{abducted} \mid \textbf{lost} \mid \textbf{met} \\
\text{Prep} &\rightarrow \textbf{in} \mid \textbf{from} \mid \textbf{to} \mid \textbf{by}
\end{aligned}
$$

(a) Is this grammar ambiguous? Illustrate your answer by considering all parses of the phrase: **Orpheus saw Eurydice in Hades**.

[3%]

(b) Is the language generated by this grammar infinite?

[1%]

(c) The grammar is in Chomsky Normal Form and so it is suitable for the CYK parsing algorithm. In your script book, draw a five-by-five table and use the CYK algorithm to parse the sentence **Orpheus saw Eurydice in Hades**.

[6%]

In an attempt to simplify the grammar a Computer Scientist produces a revised version:

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

$$
\begin{aligned}
\text{N} &\rightarrow \textbf{Fred} \mid \textbf{Hades} \mid \textbf{Orpheus} \mid \textbf{Eurydice} \\
\text{V} &\rightarrow \textbf{saw} \mid \textbf{abducted} \mid \textbf{lost} \mid \textbf{met} \\
\text{Prep} &\rightarrow \textbf{in} \mid \textbf{from} \mid \textbf{to} \mid \textbf{by}
\end{aligned}
$$

(d) Is the revised grammar ambiguous? If it is, demonstrate this, if not provide brief notes on why the grammar is unambiguous.

[2%]

(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 $\text{S}, \text{NP}, \text{VP}, \text{PR}, \text{PP}$.

[3%]

(f) Using this augmented grammar, calculate $\text{Follow}_1(A)$ for the nonterminal PP.

[4%]

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

[5%]

(h) Is the grammar $LL(1)$?

[1%]

Specimen Answers

Part A

1. d – because this string is in the language and is longer than k so we can apply the Pumping Lemma with this as our choice of string.

2. a – because if we change the first character in a string in the language we can only change the last character in the string to ensure the changed string is in the language and so on for the second and second to last character ...

3. b - if the number of b's read by the machine is odd we are in either state 2 or state 4, if the number of a's is odd we are either in state 3 or state 4 so if we are in state 4 (the final state) both are odd.

4. a – bb is in the language $b(ab^*a)^*b$ so $bbbb$ is in the language $(b(ab^*a)^*b)^*$

5. c - this is in the second RE but not in the first.

6. b – there is only one parse in which pairs of as are generated successively from S.

7. b - we know that we can construct a CFG that generates sentential forms where the number of As and Bs is equal, we then require that A is rewritten as an a while B can be rewritten either as a b or a c.

8. c – We can easily construct a CFG that generates L_2 in this case and intersecting this gives the language $\{a^k b^{2k} c^{2k} \mid 0 \leq k\}$

9. d – A can derive the empty string hence its inclusion and each of the other letters can start a string derived from A

10. b – the stack always comprises a list of As that count how many more as than bs we have seen, or a list of Bs that count how many more bs than as we have seen. In order to accept the end of input symbol we must have at least one A on the stack.

11. a

12. c

13. b

14. a

15. c

16. d

17. e

18. b

19. d

20. c

Part B

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

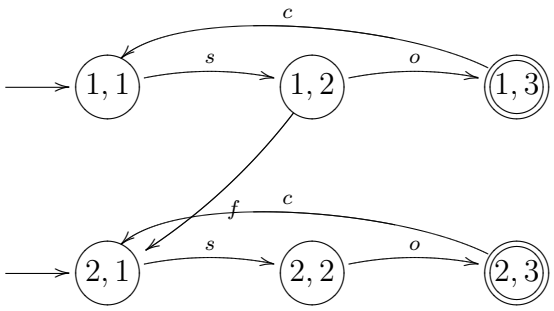
$$\begin{aligned} R_1 &= sR_2 \\ R_2 &= fR_1 + oR_3 \\ R_3 &= cR_1 + \varepsilon \end{aligned}$$

Allocate three marks for the equations.
Solving the equations:

$$\begin{aligned} R_2 &= fsR_2 + oR_3 \\ R_2 &= (fs)^* oR_3 \\ R_1 &= s(fs)^* oR_3 \\ R_3 &= cs(fs)^* oR_3 + \varepsilon \\ R_3 &= (cs(fs^*)o)^* \\ R_1 &= sfR_1 + soR_3 \\ R_1 &= sfR_1 + so(cs(fs^*)o)^* \\ R_1 &= (sf)^* so(cs(fs^*)o)^* \end{aligned}$$

Allocate three marks for solving the equations.

(b) The intersection construction yeilds the machine:



Allocate 2 marks for getting the stateset correct, including the final state. 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:

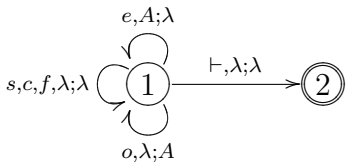
State	c	f	o	s
1				2,4
$2, 4$		1	3	2,4
3	1			

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 $x = (so)^k e^k \in L_4$.
  - By the pumping lemma there are strings $uvw = x$ with the length of uv less than k and $v \neq \varepsilon$ such that $uw \in L_4$
  - uv must be a prefix of $(so)^k$
  - There are two cases (a) $v = (so)^j$ for some $0 < j \leq k$ or (b) this is not the case. For case (b) uv is not a member of L_4, hence a contradiction or in case (a) we have $(so)^{k-j}e^k \in L_4$ and again this is a contradiction.

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 $s, c, f, \lambda; \lambda$ stands for three transitions $s, \lambda; \lambda$, $c, \lambda; \lambda$, $f, \lambda; \lambda$:



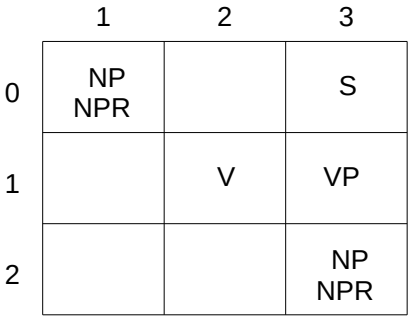
Allocate 1 mark for geting the PDA notation correct, two marks for the o and e transitions and a further one mark for the other transitions.

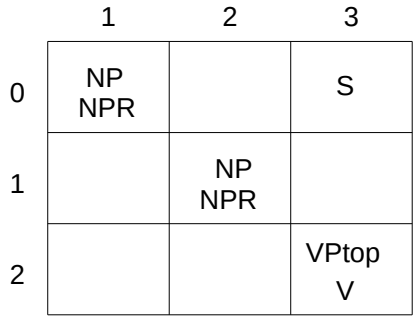
2. (a) **Answer:**

S → NP NP VP_top
VP_top → V

Marking guide: 2 marks each for each rule. 1 mark for recognizing that one rule each is needed for S and for a SPECIAL TYPE of VP.

(b) **Answer:**





Marking guide: The two WFSTs can be given EITHER as matrices OR as graphs. This part should be marked INDEPENDENTLY of part a. So if the student provides WFSTs that are correct with respect to the grammar rules they've given in answer to part a, mark them correct.

(c) **Answer:**

VP → V NP NPR

Alternatively,

VP → V NP NP_nopro
NP_nopro → NPR

Marking guide: The second answer isn't as good as the first, so deduct a point. The given grammar only provides two ways of realising an NP – as a proper name (NPR) or as a pronoun. So if the object NP can't be realised as a pronoun, then it has to be realised as an NPR.

(d) **Answer:** NO. You cannot avoid sentences like "Mary gave John it" simply by assigning different probabilities to the given rules for re-writing NP or to the given rules for di-transitives, as this would change the probability of **any** NP anywhere being re-written as a pronoun.

Marking guide: 2 marks for recognizing that the answer is **no**. 3 marks for a correct explanation.

(e) **Answer:** Yes. You can avoid these sentences with a lexicalised PCFG.

VP → V NP NP(PRO) Prob1
VP → V NP NP(NPR) Prob2

where Prob1 (the probability of the first rule, in which the head of the NP is a

pronoun) is much less than Prob2 (the probability of the second rule, in which the head of the NP is a proper name).

Marking guide: 1 mark for recognizing that the answer is yes. 2 marks for each rule.

3. (a) Yes the grammar is ambiguous. *Allocate 1 mark.* There are two parses. The essential difference is whether the prepositional phrase is attached to the noun or the verb:

   - **Orpheus** [**saw** [**Eurydice in Hades**]]: This uses the rule: VP → PR for VP.
   - **Orpheus** [**saw** [**Eurydice**] **in Hades**]: This uses the rule: VP → PR PP for VP.

   Allocate 1 mark for each parse.

   (b) Yes. *Allocate 1 mark*

   (c) The parse table is the following:

N,NP		S		S
	V	PR,VP		PR,VP
		N,NP		NP
			Prep	PP
				N,NP

   Allocate 1 mark for the setup and one mark for each correct diagonal entered.

   (d) 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 one for the justification.*

   (e) Use the following equations:

$$\begin{aligned}
\text{First}_1(\text{PP}) &= \text{First}_1(\varepsilon) \cup \text{First}_1(\text{Prep N PP}) \\
&= \{\varepsilon, \text{Prep}\} \\
\text{First}_1(\text{PR}) &= \text{First}_1(\text{V NP}) \\
&= \{\text{V}\} \\
\text{First}_1(\text{VP}) &= \text{First}_1(\text{PR}) \\
\text{First}_1(\text{NP}) &= \text{First}_1(\text{N PP}) \\
\text{First}_1(\text{S}) &= \text{First}_1(\text{NP})
\end{aligned}$$

   (f) The only nonterminal we need to consider is PP:

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

   (g) The parse table is:

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

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

(h) Yes the grammar is $LL(1)$ because no table entry comprises two or more productions.