2024 EXAMINATIONS

Lancaster University

Part II

COMPUTING AND COMMUNICATIONS

Available Time [2.5 Hours]

SCC.312 Languages and Compilation

---

*Candidates are asked to answer **THREE** questions from **FOUR**; each question is worth a total of 25 marks. Use a separate answer book for each question.*

Question 1

1.a List all of the possible types of grammar in the Chomsky hierarchy and for each one describe the general form of the production rules and any restrictions on them.

[6 marks]

1.b Consider the following grammar:

$S \rightarrow bA \mid aB$
$A \rightarrow a \mid aS \mid bAA$
$B \rightarrow b \mid bS \mid aBB$

Classify this grammar according to the Chomsky hierarchy. Illustrate by example and with reference to your answer for part (a) why this grammar is of a certain type and why it is not of the other types.

[3 marks]

1.c Show that the grammar in part (b) is ambiguous. Illustrate your answer by providing two possible parse trees for the sentence: bbaaba

[4 marks]

1.d Describe in words the format of sentences that are generated or recognised by the grammar in part (b).

[2 marks]

1.e Generate a non-deterministic push-down automata directly from this grammar.

[6 marks]

1.f Define the halting problem and briefly explain (maximum one paragraph) the implications of the halting problem for a Universal Turing Machine simulating a Turing Machine.

[4 marks]

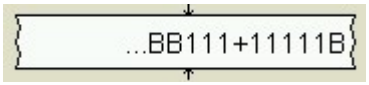
[Total 25 marks]

Question 2

a) Define a Universal Turing Machine (UTM) in terms of its relationship to an ordinary Turing Machine (TM). Your definition should include a description of what the input and output of a UTM is. In addition, for the process of coding a UTM, describe what is contained in each quintuple.

[6 marks]

b) The unary number system can be defined as follows: 1 is 1, 2 is 11, 3 is 111, 4 is 1111 and so on. This format can be used to keep a tally or count. Design a Turing machine that adds two unary numbers together. A possible input tape for a TM such as this, denoting that 3 and 5 should be added, is:



Your Turing machine should be able to process this and subsequently end up with a tape containing a string of eight 1s:

You can assume that neither number will be zero and the input tape will only consist of 1s and a single plus sign, so no error checking is required. The read/write head will start directly over the leftmost non-blank symbol. Your answer should include:

- a step-by-step description of the algorithm that you have used

[7 marks]

- a diagram of the Turing Machine itself

[12 marks]

[Total 25 marks]

Question 3

3.a Most programming languages are “whitespace agnostic” – state what this means, and describe how a lexical analyser for such a language handle whitespace?

[2 marks]

3.b Provide an example of a non-whitespace agnostic language. Suggest a way that the lexical analyser could handle this language.

[2 marks]

3.c.i Many modern compilers have a modular design, which separates each stage of the compiler from one another. Explain why this design can be advantageous.

[2 marks]

3.c.ii How might having an independent lexical analysis phase be problematic? You must include examples in your answer.

[4 marks]

3.d.i Using the supplied grammar, show each step of a bottom-up (LR) parser processing the supplied input string. Show each rule you use at each step, and you should complete your transforms with the only remaining non-terminal as ‘E’.

Grammar	*Input*
E --> E * B E --> E + B E --> B B --> 0 B --> 1	“1 + 0 * 1”

[4 marks]

3.d.ii Assume that the non-terminal ‘B’ is expanded to include any integer number. Why is this particular combination of parser and bottom-up strategy likely to result in incorrect calculations? You should include an example.

[3 marks]

3.d.iii Using the following grammar and input string (the same as 3.d.i), show each transform for a top-down (LL) parse. Start with 'E'.

Grammar	*Input*
E --> E * B E --> E + B E --> B B --> 0 B --> 1	"1 + 0 * 1"

[4 marks]

3.e State and provide an example for each form of type coercion

[4 marks]

[Total 25 marks]

Question 4

4.a.i Assuming a language where single letters and arithmetic operators are the terminals, draw the two possible parse trees for the following expression:

$$a + b \times c$$

[4 Marks]

4.a.ii Why is the situation in question 4.a.i problematic for the compiler?

[2 Marks]

4.b State what the 3 types of dynamic or static type checks are.

[3 Marks]

4.c Here is a grammar G, with terminals {b, y, e}, non-terminal {Y} where Y is the distinguished symbol.

Y -> b Y e
Y -> y

Build the Action and Goto tables for this grammar. Marks will be awarded as follows.

Step 1 :	build the item sets.	**[6 marks]**
Step 2 :	generate the state transition table	**[4 marks]**
Step 3 :	produce the Action and Goto Tables.	**[6 marks]**

[Total 25 marks]

--- End of Paper ---