2023 EXAMINATIONS

Lancaster University

Part II

COMPUTING AND COMMUNICATIONS – Online Assessment [2.5 hrs]

SCC.312 LANGUAGES AND COMPILATION

---

*Candidates are asked to answer **THREE** questions from **FOUR**; each question is worth a total of 20/25 [delete as applicable] marks.*

Lancaster University

Question 1

Question 1 consists of two types of questions:

- Multiple-choice questions that allow multiple selections. Some questions may have more than one correct answer, and you must select all of the correct answers to receive full credit.

- Practical JFLAP questions that must be solved using JFLAP 7.1, similar to how you used it to solve the languages coursework as well as the lab-work for Week 12.

For questions asking you to use JFLAP:

- Use JFLAP version 7.1, which you can download directly.
- Test and run your JFLAP machines before submitting them to ensure they work correctly.
- Mark the start and end states of your JFLAP machines correctly.
- Use small letters for all terminal symbols.
- There are no partial marks for JFLAP questions.
- Use the correct file naming format as suggested in each question.
- Using Moodle, drag and drop (or browse and select) your JFLAP (.jff) files as answers to the intended question.
- Hand-drawn machines are not accepted.
- Use "Save" or "Save As" to save your JFLAP machines as .jff files.

Question 1 continues on next page…

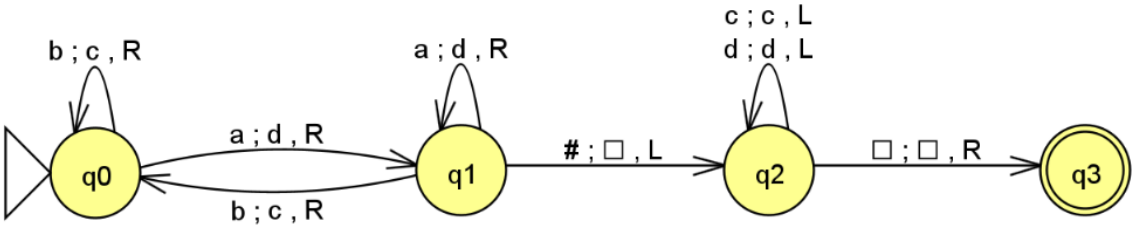
Question 1 continued.

1.a.
Which of the following statements about Chomsky's hierarchy are true?

a. A regular grammar can generate a regular language.
b. A context-free grammar can generate a context-sensitive language.
c. A type-0 grammar can generate any recursively enumerable language.
d. A type-2 grammar can generate any context-free language.
e. A type-3 grammar can generate any regular language.

[3 Marks]

1.b. Consider the following Turing Machine Transducer, select the statements that are true about this machine:



a. The machine Accepts the following input string: bbbaaa#
b. If the input string aaabbb# the transducer's output will be bbbaaa#
c. The machine Accepts the following input string: aaaddd#
d. If the input string bbbaaa# the transducer's output will be cccaaa#
e. If the input string bbbaaa# the transducer's output will be cccddd#

[3 marks]

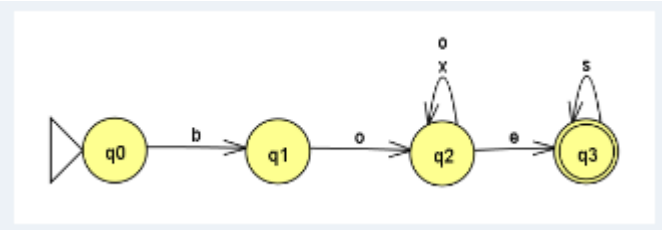
Question 1 continues on next page…

Question 1 continued.

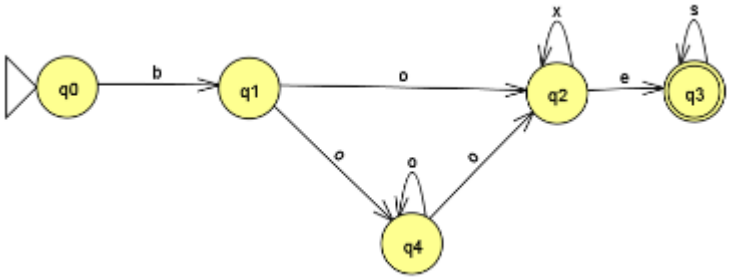
1.c. Select the Finite State Recognisers (FSR machine) that are equivalent to the following Regular Expression (RegEx)

bo+x*es*

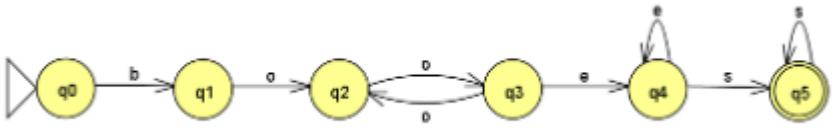
a.



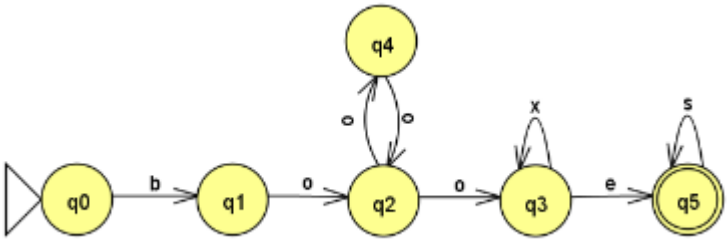
b.



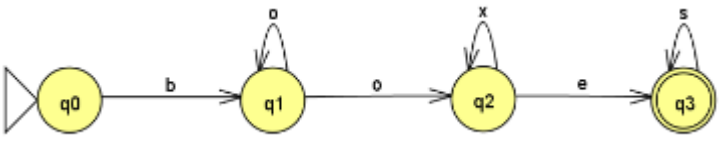
c.



d.



e.



[3 marks]

Question 1 continues on next page…

Question 1 continued.

1.d. Consider a context-free language L defined by a set of production rules. Which of the following statements about context-free languages are true?

a. A context-free language can generate an infinite number of distinct strings.
b. A context-free language can be generated by a regular grammar.
c. A context-free language can be ambiguous.
d. A context-free language can be generated by a context-sensitive grammar.
e. A context-free language can be recognized by a pushdown automaton.

[3 marks]

1.e. Create a Finite State Recogniser (FSR) using the following regular grammar production rules.
In your answer you should mark which nodes are the start (initial) and final (halt) ones. If your machine is missing a start or a final state, your answer will be considered invalid, and you get a zero mark. Pay extra care for spelling. A reminder that terminals are written in small letters. Both, deterministic or non-deterministic answers are accepted. **Use JFlap 7.1 and save the machine as StudentID.Q1e.jff (e.g. 1234567.Q1e.jff)**

- Start symbol = S
- Non-terminals = S, A, B, C, D, E, F, G, J, H
- Terminals = c, m, a, r, x, d, s
- λ = lambda where the machine ends

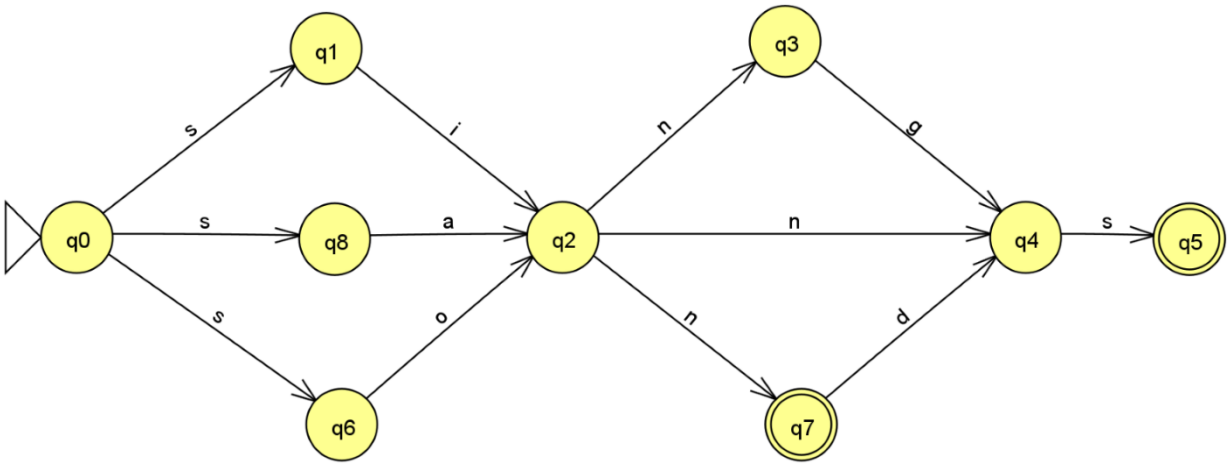
S --> cA | mB
A --> aA | aC
B --> rD | rB | aC
C --> xE
D --> xF
E --> xG
F --> dJ
G --> sH
J --> sH
H --> sH | sS | λ

[6 marks]

Question 1 continues on next page…

Question 1 continued.

1.f. Convert the following non-deterministic finite state recogniser into a deterministic one using the subset construction algorithm (all the terminals in the JFLAP figure below are lower case.
For clarification, those are: **s, a, o, i, n, g, d**.
In your answer you should mark which nodes are the start (initial) and final (halt) ones. If your machine is missing a start or a final state, your answer will be considered invalid, and you get a zero mark. **Use JFlap 7.1 and save the machine as StudentID.Q1a.jff (e.g. 1234567.Q1a.jff).**



[7 marks]

[Total 25 Mark]

Lancaster University

Question 2

Question 2 consists of two types of questions:

- drag-and-drop questions, where you will need to drag text into predefined boxes, either in between text or on images.
- Multiple-choice questions that allow multiple selections. Some questions may have more than one correct answer, and you must select all of the correct answers to receive full credit.
- Practical JFLAP questions that must be solved using JFLAP 7.1, similar to how you used it to solve the languages coursework as well as the lab-work for Week 12.

For questions asking you to use JFLAP:

- Use JFLAP version 7.1, which you can download directly.
- Test and run your JFLAP machines before submitting them to ensure they work correctly.
- Mark the start and end states of your JFLAP machines correctly.
- Use small letters for all terminal symbols.
- There are no partial marks for JFLAP questions.
- Use the correct file naming format as suggested in each question.
- Using Moodle, drag and drop (or browse and select) your JFLAP (.jff) files as answers to the intended question.
- Hand-drawn machines are not accepted.
- Use "Save" or "Save As" to save your JFLAP machines as .jff files.

Question 2 continues on next page…

Question 2 continued.

2.a. The Chomsky Hierarchy is a classification of formal grammars and languages, proposed by Noam Chomsky in 1956. It defines four types of grammars, which are increasingly more powerful in terms of the languages they can generate. Complete the Table below which shows the different Types of Grammar. Drag drop the choices into the correct cell.

Type	Grammar	Language Generated	Automaton
0	blank	blank	blank
1	blank	blank	blank
2	blank	blank	blank
3	blank	blank	blank

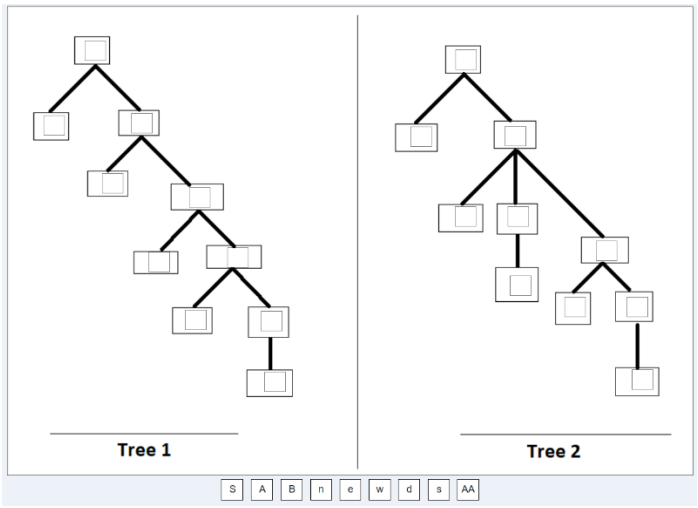
FiniteState Automaton ContextSensitive Languages Regular Grammar ContextSensitive Gramm ar ContextFree Languages ContextFree Grammar Linear Bounded Automaton All Languages Unr estricted Grammar Regular Languages Pushdown Automaton Turing Machine

[4 marks]

2.b. Consider the following Context Free Grammar (CFG):

S --> sA | nA
A --> eA | dB | eAA
B --> s | sS

Show that the grammar is ambiguous. Illustrate your answer by providing two possible parse trees for the sentence: 'needs'.
Drag drop and drop terminals and non-terminals to fit in the boxes provided on each tree.

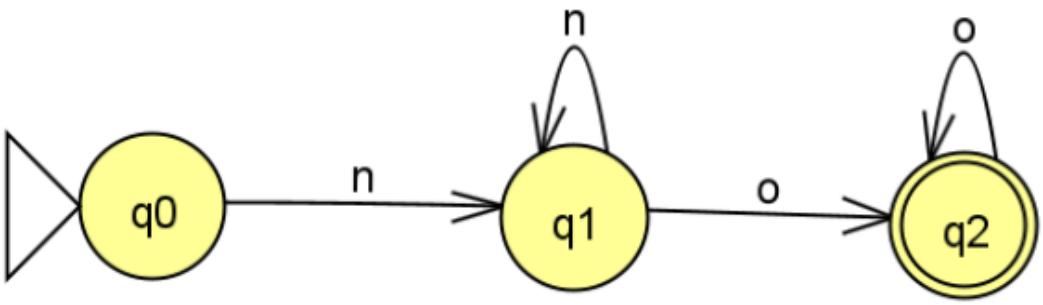


[4 marks]

Question 2 continues on next page…

Question 2 continued.

2.c. Consider the following Finite State Recogniser (FSR), select the **Set Definition** that better describes this machine.



a. $\{n^k o^k: k >= 0\}$
b. $\{n^k o^k: k >= 1\}$
c. $\{n^k o^w: k, w >= 1\}$
d. $\{n^k o^w: k, w >= 2\}$
e. $\{n^k o^w: k, w >= 0\}$

[2 marks]

2.d. Which of the following is an example of a regular language?

a. The language of all strings over the alphabet {a, b} that have an equal number of 'a's and 'b's, such as "ab", "aabb", "aaabbb", "ababab", etc.
b. The language of all strings over the alphabet {a, b} that contain a palindrome, such as "a", "aba", "abba", "abcba", etc.
c. The language of all strings over the alphabet {0, 1} that represent even numbers in binary notation, such as "0", "10", "110", "1000", "1110", etc.
d. The language of all strings over the alphabet {a, b} that start with an 'a' and end with a 'b', such as "ab", "aab", "abb", "aaab", "abab", etc.

[2 marks]

2.e. Which of the following is true about context-sensitive languages?

a. The left hand side rules are always equal to the right hand side rules
b. They can be generated by type-0 grammars
c. They can be recognized by pushdown automata (PDAs)
d. They can be generated by regular grammars
e. They can be generated by context-free grammars

[2 marks]

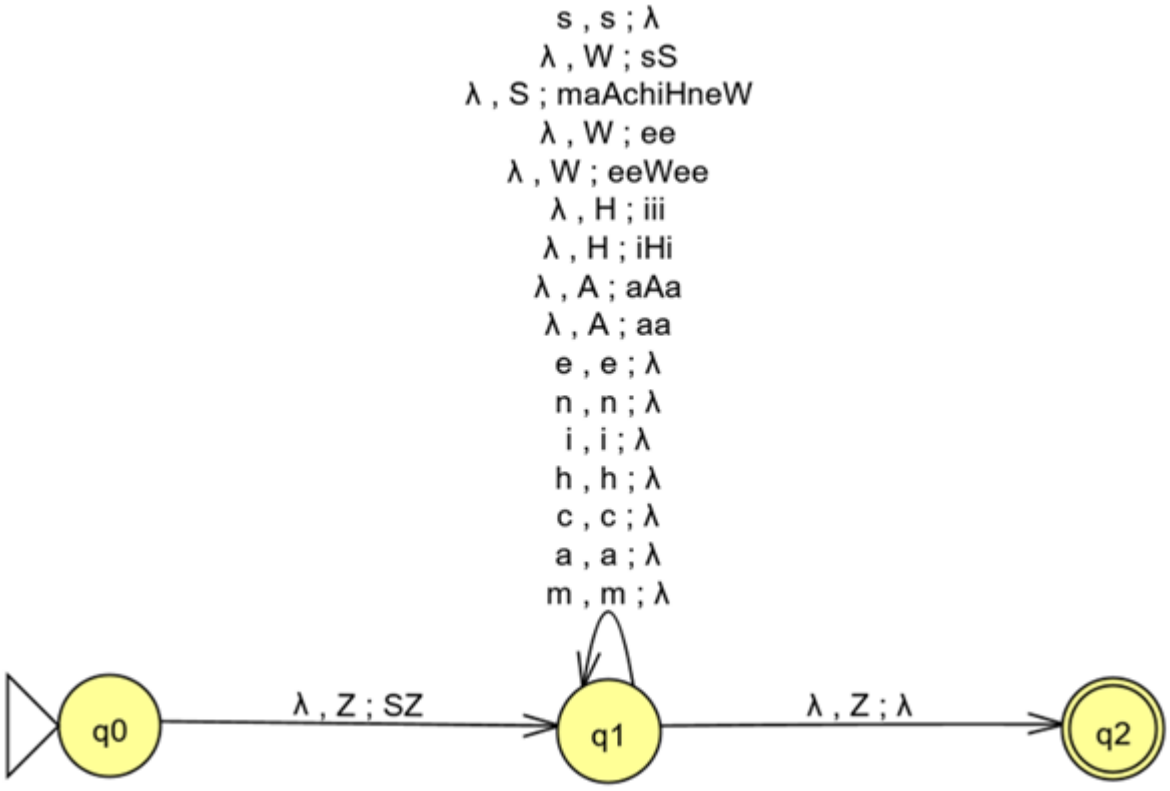
Question 2 continues on next page…

Question 2 continued.

2.f. Some context free grammars can be rewritten as regular grammars, and some are genuinely context-free. Convert the following context free grammar (Push Down Recogniser) into a regular grammar by creating a Deterministic Finite State Recogniser (DFSR) corresponding to this language (I provide the Context Free Grammar).

In your answer you should mark which nodes are the start (initial) and final (halt) ones. If your machine is missing a start or a final state, your answer will be considered invalid, and you get a zero mark. The answer must be a regular grammar or otherwise your machine is invalid and is marked as zero. **Use JFlap 7.1 and save the machine as StudentID.Q2f.jff (e.g. 1234567.Q2f.jff).**

S --> maAchiHneW
A --> aa | aAa
H --> iii | iHi
W --> ee | eeWee | sS



[8 marks]

Question 2 continues on next page…

Question 2 continued.

2.g. Consider the following Backus-Naur Form (BNF) grammar, which of the following strings are valid expressions in thelanguage generated by this grammar?
Note: the quotations " " are there to identify the terminals and are not part of the grammar.

<math>::= <number> | "(" <math> ")" | <math> <operation>
<operation>::= "+" <math> | "-" <math> | "*" <math> | "/" <math>
<number>::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" | <number><digit>
<digit>::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

a. ((((9))))))
b. (2+3)*(5-1)
c. 1+(2+(3+(4+(5+6))
d. 4+7)*3
e. 1000*2000/3000

[3 marks]

[Total 25 Marks]

Question 3

Consider the below augmented grammar, which has the non-terminals {J, C, N} and the terminals {a, b, c, d, e}, where J is the distinguished symbol.

S -> J
J -> a C
C -> b N
N -> c N d
N -> e

Construct the Action table to parse this grammar. You must use the technique for this which was demonstrated in the course.

Marks are awarded as follows:

i. Construct the item sets, showing each step **[16 marks]**

ii. Construct the state transition table **[3 marks]**

iii. Produce the Action table **[6 marks]**

[Total 25 Marks]

Question 4

Consider the grammar shown below

<program> = <statement>
<statement> -> ident = <expression> | if <expression> then <statement>
<expression> -> ident | <comparison>
<comparison> -> <expression> == <expression> | <expression> > <expression>

4.a. What is the FIRST set for the non-terminal <statement>

[2 marks]

4.b. What is the FOLLOW set for the non-terminal <comparison>

[2 marks]

4.c. Is this grammar LL(1)

[1 mark]

4.d. Why

[1 mark]

4.e. For a recursive descent parser, write the function to recognise <statement>. You can assume the existence of a helper function acceptTerminal(t) and a special variable nextToken containing the next token from the lexical analyser. The acceptTerminal(t) function, on a matching token, advances the nextToken variable to contain the subsequent token in the lexical analyser's token stream.

[11 marks]

4.f. Consider the following function:

```
int add(int a, int b)
        {
        int result = a + b
        return result
        }
```

What would a compiler's symbol table contain following successful parsing of this function?

[4 marks]

Question 4 continued on next page…

Question 4 continued.

4.g. Now consider the below function:

```
int div(int a, int b)
        {
        float result = a / b
        return result
        }
```

How would a compiler treat this function? What would the compilation result be, and how would this result be derived?

[4 marks]

[Total 25 Marks]

---End of Paper---