SCC.312 Languages and Compilation

Coursework Part 1: Languages

Assessment Title	Languages Coursework
Module Convener	Mo El-Haj
Distribution Date	3rd February 2023
Submission Deadline	Friday, 17 February 2023, 16:00
Contribution to overall module assessment	20%
Indicative student time working on assessment	4-6 hours
Assessment Type (individual or group)	individual

Main Objectives of the assessment

Formative assessment on the first five weeks-worth of learning outcomes in the languages-half of the module.

Description of the Assessment

A workbook of questions similar in style to those covered in weeks 12-13 non-assessed quizzes and workbook performed in the labs. A total of 6 questions.

Format of the Assessment

This is a practical assessment, you need to use **JFLAP 7.1 (direct download)** to answer the questions and provide your output files.

Submission Instructions

Coursework must be submitted electronically via the University's Moodle system. The required file formats for this coursework is **zip** file with your answers (.jff) included following the format: **studentID.zip** (e.g. **1234567.zip**).

Avoiding Plagiarism

Please ensure that you understand the meaning of plagiarism and the seriousness of the offence. Information on plagiarism can be found in the SCC Student Handbook.

Late Assessment

Please refer to the SCC Student Handbook for information on submitting late, penalties applied and procedures in the case of mitigating circumstances.

How to seek help:

Myself and the TAs will be around during the labs for questions. Please note that we will not answer questions such as 'is my solution correct'…etc. But either way, you are encouraged to come and talk to us if you feel stuck or if you are not sure about any point.

Please do not send your answers/files by email or Teams for me to check. Make sure you attend the labs to ask questions (there are 5 labs during the week and you are welcome to come to any of them to ask me or the TAs. The labs times and locations are on Moodle)

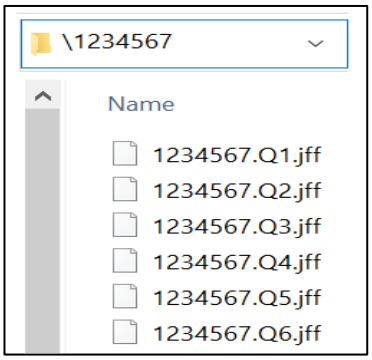
Important Notes (please read before answering the questions to avoid losing marks – always ask if in doubt):

- **IMPORTAN_1:** your JFLAP machine answers will be marked automatically as runnable machines using JFLAP. The process will also generate an auto-feedback. I will still give generic feedback during one of the lectures and highlight common errors and mistakes so you can avoid those in the future. You are more than welcome to approach me after the marks are out for more feedback.
- **IMPORTANT_2:** It is your responsibility to test and run your submitted JFLAP machines. Make sure they run and are generating the correct answer by accepting the appropriate strings. If your machine does not run, is stuck in a loop, generates errors, or accepts the wrong grammar/strings, the machine is marked as zero.
- **IMPORTANT_3:** Pay extra care for Start (Initial) and Final (Halt) states. You must mark them in your JFLAP answers or otherwise your machine(s) will not run correctly, and they will be considered invalid. If your machine is missing an initial or a final state, you get a zero mark for that question.
- **IMPORTANT_4:** Pay extra care for spelling and grammar. The machine may be considered invalid and could be marked as zero. If you make an honest spelling mistake (e.g. wrote a '0' instead of 'o'), come talk to me.
- **IMPORTANT_5:** Terminals are strictly in small letters. If you use capital letters for terminals, your machine will not run correctly, machines that do not run correctly are considered invalid and are marked with zero.
- **IMPORTANT_6:** There are no partial marks. Your machine should run correctly, otherwise it will be considered incorrect. This is to help you understand how important grammar is.
- **IMPORTANT_7:** You must provide your JFLAP (**.jff**) answers. An answer that is missing a **.jff** machine will not be marked.
- You must use **JFLAP version 7.1** to answer the questions (direct download).
- Use correct format. For example, if answering Q1, your .jff file should be named using your studentID as in **StudentID.Q1.jff** (e.g. **1234567.Q1.jff**) please do not use any other format to avoid losing marks.
- To save your machine use save or save as from the file menu in JFLAP so you can save it locally as .jff file.
- When you finish answering, test your machines make sure they run correctly then Zip all your answer machines into one zip file. The zip file name should be your **StudentID.zip** (e.g. 1234567.zip). No need for subfolders.
- You do not need to write or upload a report. The machines are auto-marked using JFLAP. Only **.jff** files are accepted in your **zip** file, any other formats will not be marked.

Check your submission structure:

Before you zip your answer files, please make sure this is how the structure of your folder looks like (here assuming your StudentID is 1234567): **[pay attention for spelling and spaces to avoid losing marks].**

It is your responsibility to check that you have submitted the correct answers.



When you are satisfied that your folder contains the correct .jff answers, you can then zip your files and upload the zip file to the Moodle submission point.

The questions start on the next page (6 questions in total) …

Q1: 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.Q1.jff (e.g. 1234567.Q1.jff)**

Start symbol = S
Non-terminals = S, A, B, C, D, E, F, G, H
Terminals = p, a, s, w, o, r, d
λ = lambda (similar to epsilon ε), this is where the machine terminates (no need to include epsilon or lambda in your JFLAP answer, it's to help you know which state is your final state).

S → pA
A → aA | sB
B → wC | sB
C → oD |
D → oG | oD
G → rF |
F → dE |
E → dE | sH
H → λ | sC

[2 marks]

Q2: Convert the following Regular Expression (RE) into a Finite State Recogniser (FSR).
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. Both, deterministic or non-deterministic answers are accepted. **Use JFlap 7.1 and save the machine as StudentID.Q2.jff (e.g. 1234567.Q2.jff). HINT:** *remember this is a regular expression, therefore + * and () have special meaning and are not terminals.*

la+ncaas+te+rr(fas+te+rr)*

[2 marks]

Q3: Make a finite state recogniser (FSR) for the following Backus–Naur Form (BNF) regular grammar, which shows the addition, subtraction, multiplication or division (+, -, *, /) of any single roman number (where **i**, **v**, **x, m**, **d**, **+**, **-**, *****, **/**) are terminal symbols). You can simplify the names of the non-terminals if you wish.
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. Both, deterministic or non-deterministic answers are accepted. **Use JFlap 7.1 and save the machine as StudentID.Q3.jff (e.g. 1234567.Q3.jff).**

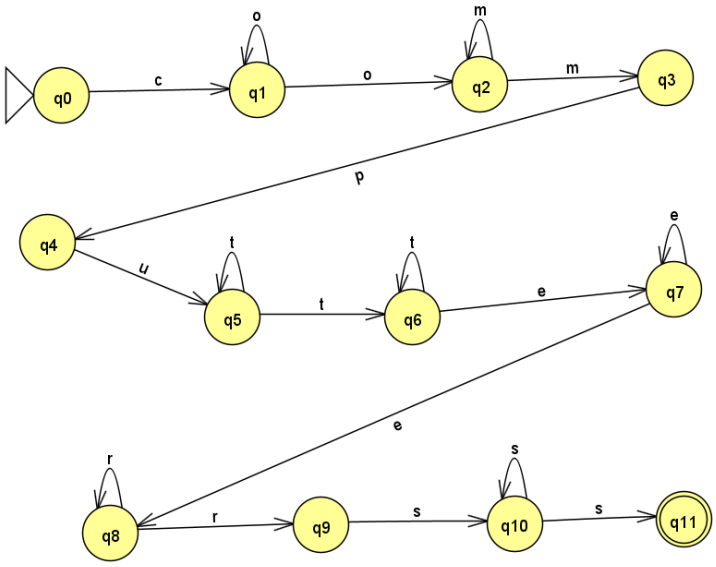
<math>::= **i** | **v** | **x** | **m** | **d** | **i**<operation> | **v**<operation> | **x**<operation> | **m**<operation> | **d**<operation>
<operation>::= **+**<math> | **-**<math> | *****<math> | **/**<math>

[2 marks]

Q4: Modify the finite state recogniser you created for **Q3** to allow the addition, subtraction, multiplication or division of any number of roman-number sequences rather than a single number.
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. Both, deterministic or non-deterministic answers are accepted. **Use JFlap 7.1 and save the machine as StudentID.Q4.jff (e.g. 1234567.Q4.jff).**

[2 marks]

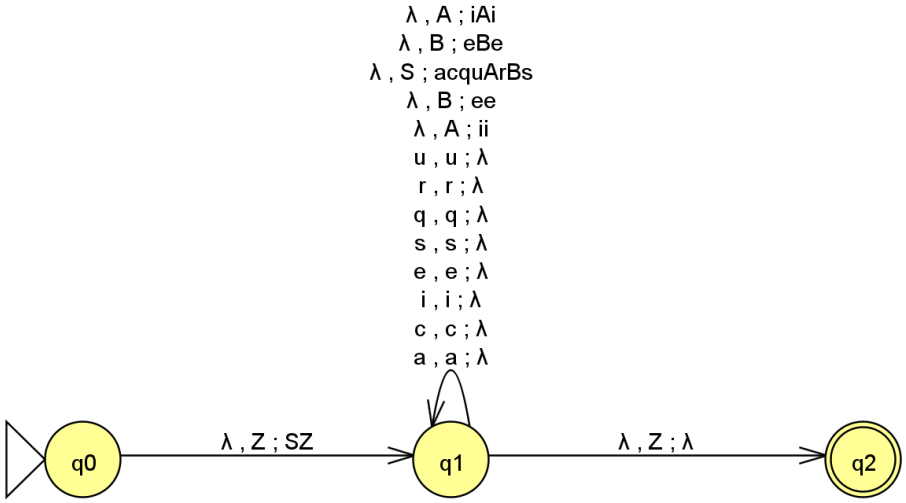
Q5: Convert the following non-deterministic finite state recogniser into a deterministic one using the subset construction algorithm (all the terminals in the JFLAP figure below are lower case.
For clarification, those are: c, o, m, p, u, t, e, r, s.
In your answer you should mark which nodes are the 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.Q5.jff (e.g. 1234567.Q5.jff).



[2 marks]

Q6: 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.

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.Q6.jff (e.g. 1234567.Q6.jff).



[2 marks]

[Coursework total marks 12]

Conversion to 20% (your-mark*20/12)

--- End of Coursework Paper ---