CSCI3136
Assignment 3

Instructor: Alex Brodsky

Due: 9:00am, Friday, June 7, 2018

1. **[30 marks]** Write a scanner for a language that contains the following kinds of tokens:

   operators: `! % & | * - / + = # < > @`
   delimiters: `( ) [ ] { } ,`
   keywords: `true false nil let letrec def set lambda if elseif else guard catch raise`
   integer literal: `[0-9][0-9]*`
   string literal: `"[^"]*"`
   symbol: `[a...zA-Z_][0-9a-zA-Z_]*` that is not a keyword

   where the latter three tokens are specified by regular expressions. For example, `123`, `0`, and `0123` are all integer literals; `"Hello World!"` and `""` are string literals; and `var`, `abc123`, and `_asYouLikeIt` are all identifiers. **Note:** for this assignment treat keywords as symbols.

   You may implement your scanner in any language that you wish, but it must be runnable on the `bluenose` server. The scanner **must** take input from `stdin` and output to `stdout`. The input to the scanner will be text (a program), Note, whitespace is for the most part ignored.

   For example, the following would be a possible input to your scanner.

Sample Input	Sample Output

`if a > 2{`
`  "Hello World?"`
`} elseif b >3 {`
`  "Ciao World?"`
`}else {`
`  "Goodbye World?"`
`}`
	`line 1 col 1 : if`
	`line 1 col 4 : a`
	`line 1 col 6 : >`
	`line 1 col 8 : 2`
	`line 1 col 9 : {`
	`line 2 col 3 : "Hello World?"`
	`line 3 col 1 : }`
	`line 3 col 3 : elseif`
	`line 3 col 10 : b`
	`line 3 col 12 : >`
	`line 3 col 13 : 3`
	`line 3 col 15 : {`
	`line 4 col 3 : "Ciao World?"`
	`line 5 col 1 : }`
	`line 5 col 2 : else`
	`line 5 col 7 : {`
	`line 6 col 3 : "Goodbye World?"`
	`line 7 col 1 : }`

Note the input is purposely formatted poorly to indicate that white-spaces are not guaranteed to separate every two tokens.

The output of your scanner should be the sequence of tokens, as each token is scanned. The output includes the line and column of the first character of each token. This is useful for generating error messages later on. Each token should be printed on a separate line with the following format:

`line` L `col` C `:` T

where L is the line number (starting at 1), C is the column of the first character of the token (starting at 1), and T is the token itself. This is illustrated in the previous example.

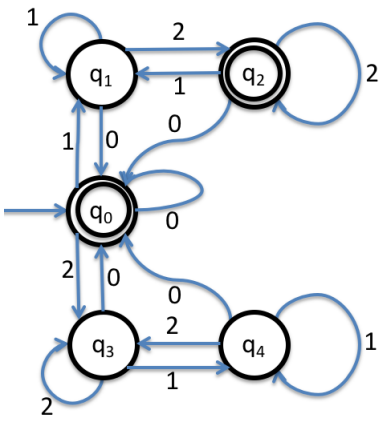
The sample solution takes about 140 lines of Python code. A set of test cases is provided for you to test your scanner. A test script `test.sh` is provided to run the tests. Since the choice of language is up to you, you must provide a standard script called `runme.sh` to run your scanner. For example the script for a Python solution looks like:

```
#!/bin/sh

# Command to run your program
python3 splat.py
```

To submit this part of the assignment please use Brightspace. Please submit both the written work and the program electronically. Remember, you need to include a script called `runme.sh` that will let the marker run your code. Note: A `makefile` to compile your program, if compiling is needed, would be greatly appreciated.

2. **[5 marks]** Minimize the following DFA M:



To minimize M we perform the equivalence class splitting algorithm. Initially there are two classes $C_0 = \{q_1, q_3, q_4\}$ and $C_1 = \{q_0, q_2\}$. We then look at each class and consider the transitions.

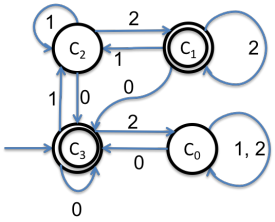
Iteration 1

State		0	1	2
C_0	q_1	q_0 (C_1)	q_1 (C_0)	q_2 (C_1)
	q_3	q_0 (C_1)	q_4 (C_0)	q_3 (C_0)
	q_4	q_0 (C_1)	q_4 (C_0)	q_3 (C_0)
C_1	q_0	q_0 (C_1)	q_1 (C_0)	q_3 (C_0)
	q_2	q_0 (C_1)	q_1 (C_0)	q_2 (C_1)

Iteration 2

State		0	1	2
C_2	q_1	q_0 (C_3)	q_1 (C_2)	q_2 (C_1)
C_0	q_3	q_0 (C_3)	q_4 (C_0)	q_3 (C_0)
	q_4	q_0 (C_3)	q_4 (C_0)	q_3 (C_0)
C_3	q_0	q_0 (C_3)	q_1 (C_2)	q_3 (C_0)
C_1	q_2	q_0 (C_3)	q_1 (C_2)	q_2 (C_1)

Observe that none of the transitions causes either of the classes to be split, so the resulting minimal DFA is of the form:



3. **[5 marks]** Is the language $L = \{\sigma \in \{a, b\}^* \mid |\sigma|_a - |\sigma|_b \textit{ is divisible by } 2^n, n \geq 1\}$ regular? Be sure to prove your answer.

   L is regular. Proof: Let $n = 1$. We can easily construct a 2-state DFA that keeps track of the difference between the number of a's and b's modulo 2, which is that is necessary to recognize this language.

4. **[10 marks]** Prove that the language $L = \{a^{2^n} | n \geq 0\}$ is not regular.

   Proof by contradiction. Assume L is regular and there is an n as specified by the Pumping Lemma.
   Let $\sigma = a^{2^n}$.
   According to the Pumping Lemma, $\sigma = \alpha\beta\gamma$, where $|\alpha\beta| \leq n$ and $|\beta| > 0$, and $\sigma' = \alpha\beta^k\gamma \in L$, for all $k \geq 0$.
   Now, if we set $k = 2$, then $2^n < |\sigma'| \leq 2^n + n < 2^n + 2^n = 2^{n+1}$ because $0 < |\beta| \leq n$ and $n < 2^n$ for all $n \geq 0$. Thus, $\sigma' \notin L$ because its length is not a power of 2.
   But, according to the Pumping Lemma, $\sigma' \in L$. Contradiction. Hence, L cannot be regular.

CSCI3136: Assignment 3

Summer 2019

Student Name	Login ID	Student Number	Student Signature

	Mark
Question 1	/30
Question 2	/5
Question 3	/5
Question 4	/10
Total	**/50**

Comments:

Assignments are due by 9:00am on the due date. Assignments *must* be submitted electronically via Brightspace. Please submit a PDF for the written work. You can do your work on paper and then scan in and submit the assignment.

Plagiarism in assignment answers will not be tolerated. By submitting their answers to this assignment, the authors named above declare that its content is their original work and that they did not use any sources for its preparation other than the class notes, the textbook, and ones explicitly acknowledged in the answers. Any suspected act of plagiarism will be reported to the Facultys Academic Integrity Officer and possibly to the Senate Discipline Committee. The penalty for academic dishonesty may range from failing the course to expulsion from the university, in accordance with Dalhousie Universitys regulations regarding academic integrity.