Test 1 – Regular Expressions

Test1 (Language Processors) will cover Regular Expressions and related topics, as introduced by the lecture notes of week 2: Language Processing and Lexical Analysis. Below we show the type of questions that you will be expected to answer. Some questions will be multiple choice,.

Regular Expressions

1) The following regular expression recognises certain strings consisting of the letters a, b and c:
   b^+ [a]? (c|ab)*

   - For the following 5 strings, indicate whether or not they are recognised by the above regular expression:

   accab, bbccca, bccabc, bbbba, baab

   - Show three more strings that are recognised by the above expression.

   - Show two more strings consisting of the letters a, b and c that are ***not*** recognised by the above regular expression.

2) A hexadecimal literal is a string that starts with prefix 0x, followed by a non-empty sequence of digits and uppercase/lowercase A,B,C,D,E. Examples: 0xFF, 0xa230, 0x0, 0x12. Write down a regular expression that defines hexadecimal literals.
3) Write a regular expression that defines the set of even numbers.
   Hint: Even numbers always end with 0,2,4,6 or 8.
4) Is it possible to specify the set of multiple of 3 numbers (0,3,6,9,1215,etc) using regular expressions?
5) Given alphabet a,b,c and d. Write down a regular expression that recognises strings that/where:
   a. Contain at least one character a.
   b. Contain at least two characters b.
   c. The first **a** must start before the first **d** (if any).
   d. Character c must always be immediately followed by d. For example "abbbcdad" is correct while "abbbcaad" is not.

Lexical Specifications

1) Given the following lexical specification:

Token	RegExp	JavaCC
BINARY	b (0 | 1)+	???????
TRUE	true	
ID	[a-zA-Z] ([a-zA-Z] | [0-9])*	
FALSE	false	
ERROR	~[SKIP]	

a. Assuming that whitespaces are skipped – the definition SKIP denotes all whitespace characters – and have precedence to the other token definitions, what tokens will be recognised by the lexical analyser given the following input?

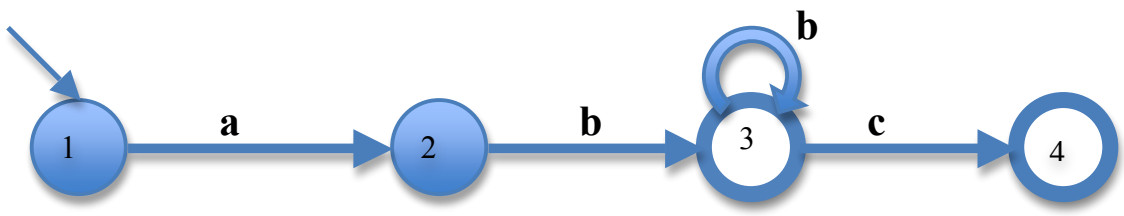
   Input: true b0 truefalse b121 false b1_2 b001X2

   Output: TRUE BINARY(b0) ID(truefalse) ID(b121) ID(false) BINARY(b1) ERROR(_) ERROR(2) ID(b001X2)

b. What do you have to do in order to make the lexical analyser recognise "false" as **FALSE**?
c. What would happen if the token **ERROR** is moved to the top of the list?
d. Replace the BINARY spec with # b (0 | 1)+. What tokens will be recognised for the input string: "b0 #b101x12 true#b"?
e. Write the JavaCC specification of the tokens defined above.

Finite Automata

1) What's the regular expression defined by the following DFA?



Answer: $ab^{+}(c)?$

a. Which are the start and final states?
   Answer: Start state = 1, Final states = 3,4.
b. Change the DFA so it defines the regular expression ab*c instead.
c. Show the sequence of states for the string "abbbc"
   Answer: 1-2-3-3-3-4

2) Write down the NFA that implements the following lexical specification:

Token	RegExp
TRUE	t | T
FALSE	f | F
ID	[a-z] ([a-z0-9])*
IDDOT	[a-z] ([a-z0-9])* ("." [a-z] ([a-z0-9])*)+

Don't forget to label final states with the corresponding token names.

a) How many final states can be reached for input string "T".
b) Show the sequence of states for string "t.x0.f1"