Formal Languages and Automata Theory
Second Assessment – November 2010

(**Duration: 45 minutes**)

LAST NAME(s) (Capital letters)			
FIRST NAME (Capital letters)			
NIA		DNI	

INSTRUCTIONS FOR THE EXAM

- Read these instructions carefully before starting the exam.
- Do not forget to write your name, NIA and DNI in every answer sheet.
- Pay attention to what it is asked in each question and/or problem, given that it is not the same: to explain, to list, to describe, to define, etc., always, sometimes, at least.
- **The duration of the exam (Test + problems) is 45 minutes.**

PROBLEM

1. An identifier in C language can be expressed as a letter whether followed by any number of letters and/or numbers or not.

   a) Write a regular expression to denote any valid identifier for this programming language.

   The regular expression that denotes every valid identifier is:

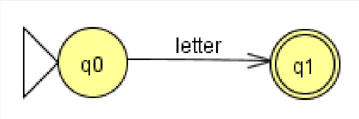
   letter (letter | digit)*

   They all must begin with a letter followed or not by any combination of letters and/or numbers

   b) Use subset construction to generate a NFA for the previous regular expression. Show the complete process and explain in detail.

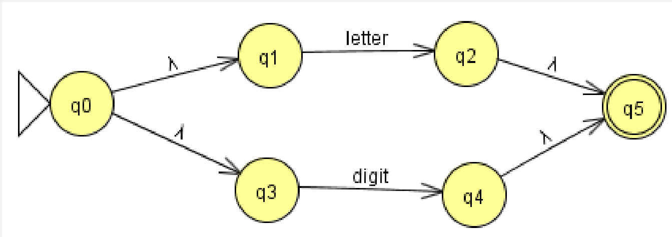
   Using subset construction, we can generate an equivalent NFA for the regular expression of the previous section:

   <u>letter</u>

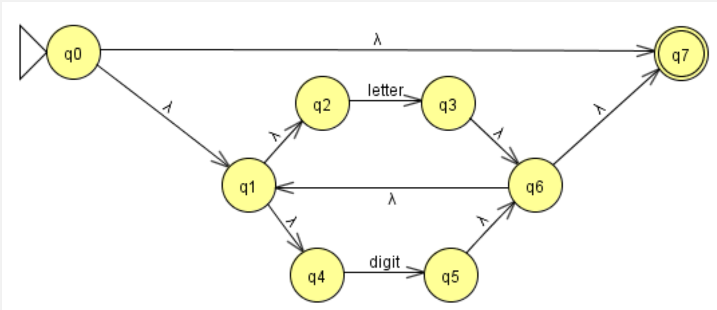


   <u>digit</u>

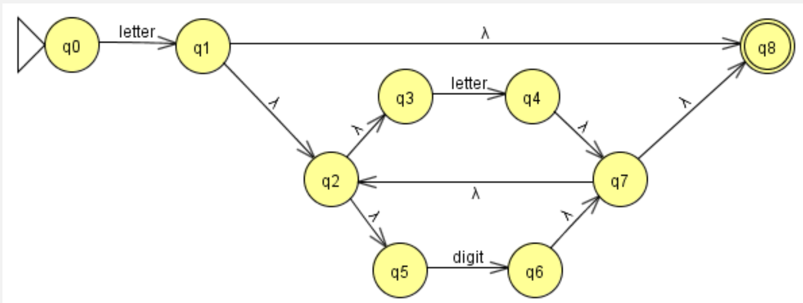
letter | digit



(letter | digit)*



letter(letter | digit)*



c) Obtain and represent an equivalent DFA applying the corresponding algorithm (NFA → DFA) and explaining the process in detail.

The initial state of the equivalent DFA can be calculated as follows:
A = λ-closure{q0} = {q0}

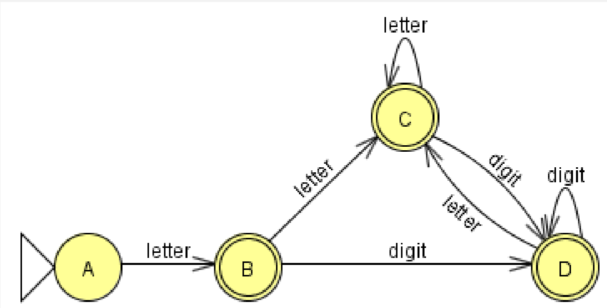
We have a transition from the initial state of the DFA to the state λ-closure{q1} with the symbol *letter*.
B = λ-closure{q1} = {q1, q2, q3, q5, q8}

Transitions from the state B
Letter: NFA q3 → q4
C = λ-closure{q4} = {q4, q7, q8, q2, q3, q5}
Digit: NFA NFA q5 → q6
D = λ-closure{q6} = {q6, q7, q8, q2, q3, q5}

Transitions from the state C:
Letter: NFA q3 → q4
λ-closure{q4} = C
Digit: NFA q5 → q6
λ-closure{q6} = D

Transitions from the state D:
Letter: NFA q3 → q4
λ-closure{q4} = C
Digit: NFA q5 → q6
λ-closure{q6} = D

Given that the final state q8 of the NFA is included in the states B, C and D of the DFA, they are also final states in the DFA.



d) Obtain an equivalent minimal DFA by calculating the equivalence relationships and quotient sets. Explain the process in detail.

Q/E0 is calculated by creating a class with final states and a second class with non-final states:

C_0 = {A}
C_1 = {B, C, D}

Q/E1

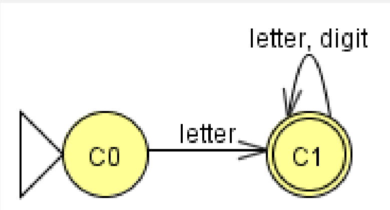
We take the class C_1, given that the class C_0 just contains one state.

f(B, letter) = C f(C, letter) = C f(D, letter) = C
f(B, digit) = D f(C, digit) = D f(D, digit) = D

Then, we can conclude that B, C, D are equivalent states.

Q / E1 = Q/E0 = Q/ E

The minimal DFA is as follows:



e) Obtain a regular grammar from the DFA by applying the corresponding algorithm (DFA → G3 grammar).

Just applying the corresponding algorithm, the equivalent type-3 grammar is:

$$G3 = (\{letter, digit\}, \{C_0, C_1\}, C_0, P)$$

where P includes:

C_0 ::= letter C_1 | letter
C_1 ::= letter C_1 | digit C_1 | letter | digit

f) Write the characteristic equations of the obtained minimal DFA, solve the system of equations and verify that the obtained regular expression generates the same language that the one defined in section a).

The characteristic equations for the minimal DFA are:

X0 = letter X1 + letter
X1 = letter X1 + digit X1 + letter + digit

We can rewrite the second equation as:

X1 = (letter + digit) X1 + (letter + digit)

Using the Arden solution:

X1 = $(letter + digit)^*$ (letter + digit)

then:

X0 = letter (letter + digit)* (letter + digit) + letter = letter $(letter + digit)^+$ + letter
X0 = letter $(letter | digit)^+$ | letter = letter $(letter | digit)^*$