Formal Languages and Automata Theory
Exercises Regular Languages
Unit 5 – Part 1

Authors:

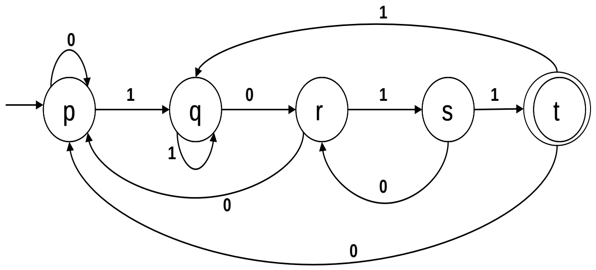
Araceli Sanchis de Miguel
Agapito Ledezma Espino
Jose A. Iglesias Martínez
Beatriz García Jiménez
Juan Manuel Alonso Weber

* Several exercises are based on the ones proposed in the following books:

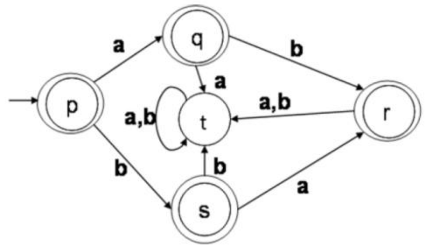
- Enrique Alfonseca Cubero, Manuel Alfonseca Cubero, Roberto Moriyón Salomón. *Teoría de autómatas y lenguajes formales.* McGraw-Hill (2007).
- Manuel Alfonseca, Justo Sancho, Miguel Martínez Orga. *Teoría de lenguajes, gramáticas y autómatas.* Publicaciones R.A.E.C. (1997).
- Pedro Isasi, Paloma Martínez y Daniel Borrajo. *Lenguajes, Gramáticas y Autómatas. Un enfoque práctico.* Addison-Wesley (1997).

Universidad Carlos III de Madrid
www.uc3m.es

1. Given the following automata, obtain the corresponding grammars.



(a)



(b)

2. Given the following grammar:

   G =({0,1}, {A,B,C}, A, P),
   P={A::=0B, A::= λ ,B::=1C , B::=1 , C::=0B)}
   Select the automaton that recognizes the language that is generated:

   a. AG=[{0,1}, {A,B,C,F}, f, A, {F}]
      f(A,0)=B, f(A, λ)= λ, f(C,0)=B, f(B,1)=C, f(B,1)= λ
   b. AG=[{0,1}, {A,B,C,F}, f, A, {F}]
      f(A,0)=B, f(A, λ)=F, f(C,0)=B, f(B,1)=C, f(B,1)=F
   c. AG=[{0,1}, {A,B,C,F}, f, A,{F}]
      f(A, B)=0, f(A,F)= λ, f(C,B)=0, f(B,C)=1, f(B,F)=1
   d. AG=[{0,1}, {A,B,C,F}, f, A, {F}]
      f(B,0)=A, f(F, λ)=A, f(B,0)=C, f(C,1)=B, f(F,1)=B

3. Obtain the Finite Automata corresponding to the following grammars:

a) G_1 = ({a,b}, {S,A,B}, S, P_1) P_1 = { S ::= aA | bA A ::= bB A ::= a B ::= aA | bA }	b) G_2 = ({a,b}, {S,A,B}, S, P_2) P_2 = { S ::= aA A ::= aA | bB A ::= a | b B ::= aA | bA }
c) G_3 = ({0,1}, {S,A,B}, S, P_3) P_3 = { S ::= 0A | 1A A ::= 0A | 1A A ::= 0B | 1B B ::= 0B | 1B B ::= 0 | 1 }	d) G_4 = ({0,1}, {S,D,E}, S, P_4) P_4 = { S ::= 1D | 0S | 0 | λ D ::= 0E | 1S | 1 E ::= 1E | 0D }

4. Obtain the minimum DFA equivalent to each one of the following grammars describing the intermediate steps: G → NFA→ DFA → minimal DFA.

a) G_A = ({a,b,c}, {S,A,B}, S, P_A) P_A = { S ::= aA | bB | c A ::= aB | b | cA B ::= a | bA | c }	b) G_B = ({a,b,c}, {S,B,C,E}, S, P_B) P_B = { S ::= a | aS | aB | cC C ::= c B ::= bE | b E ::= bB | b }
c) G_C = ({a,b,c}, {S,A,B,C,D}, S, P_C) P_C = { S ::= aA A ::= aA | bB | a B ::= bB | bC | b C ::= bC | cD | bB D ::= bC | bB |cC }	d) G_D = ({c,f,d}, {A,B,C,D,E,F}, A, P_D) P_D = { A ::= cB | cE | f | fC B ::= cB | fD | dE C ::= cA D ::= cD | fD E ::= cE | fF | dF F ::= cF | fF }

5. (Continuation Exercise 2, Unit 3, Part 1) Given the alphabet {a,b}, construct a DFA which recognizes string with length "3" of the universal language. Obtain the G3 corresponding to this automaton.

6. (Continuation Exercise 4, Unit 3, Part 2) [Exam] We have a door with only one lock. To open it, it is necessary to use three different keys (called *a*, *b*, and *c*), in a predefined order, which is following described:

   - Key *a*, then key *b*, then key *c*, or
   - Key *b*, then key *a*, then key *c*.

   If this order is not followed, then the lock is blocked (for instance, if the key *a* is used and following it is introduced again).

   Once the door is open, the introduction of keys in the lock (in every possible order) does not affect the closing device (i.e. the door remains open).

   Consider that the names of the different keys are symbols of an alphabet, over which a language L whose words are the valid sequences for the opening of the door is defined. For instance, *abcbc* is a word included in the language.

   It is required:

   a) Design a finite automata FA which accepts L.
   b) Well-formed Grammar which generates words in L.